Recommender Systems: Two Threads and Their Meeting

Guangneng Hu

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

22 Mar 2019 (Fri),
WeChat, Guangzhou

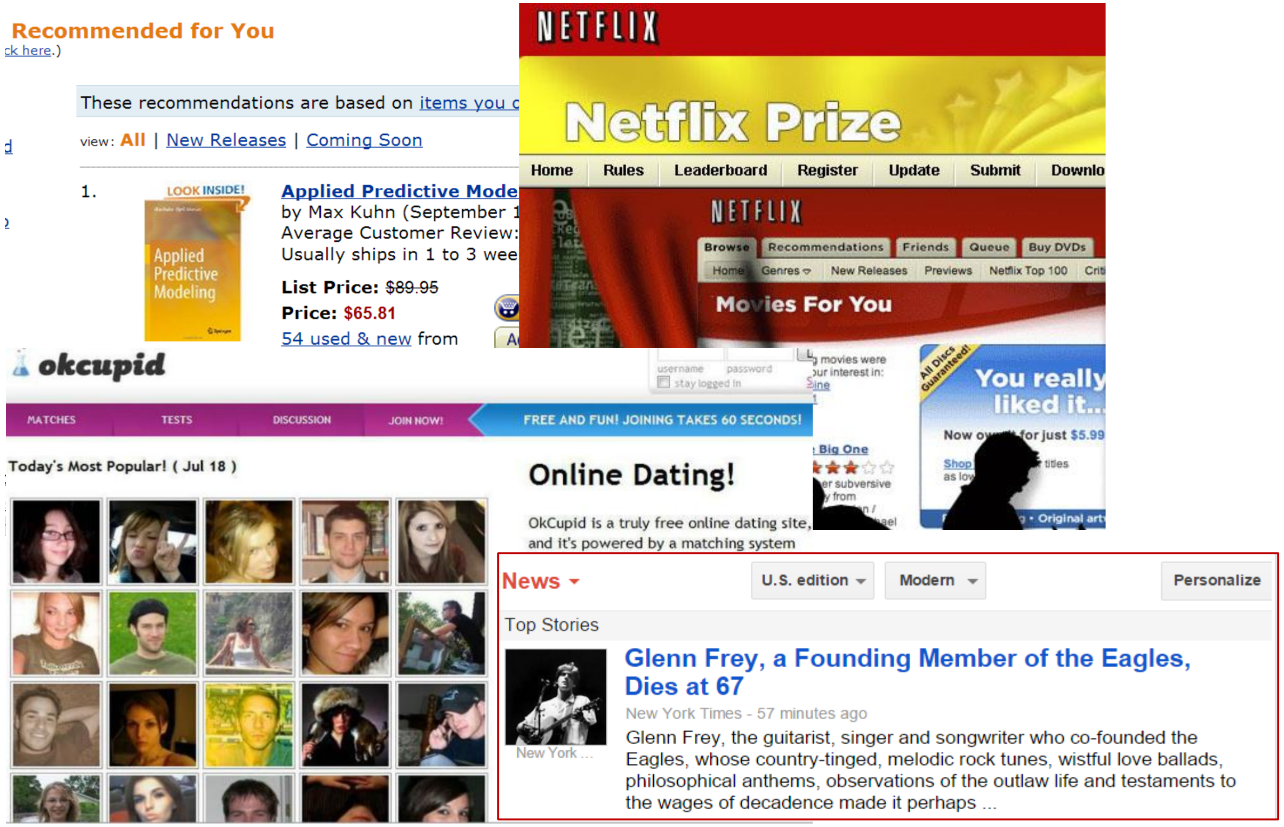
Outline

- Introduction
- Collaborative filtering
  - Matrix factorization, metric learning, neural approaches
- Cross-domain recommendation
  - Collective matrix factorization
  - Deep transfer learning
- Hybrid filtering
  - Personalized word embeddings
- Transfer meets hybrid
- Conclusion

Introduction

Recommendations Are Ubiquitous: Products, Medias, Entertainment…

- Amazon
  - 300 million customers
  - 564 million products
- Netflix
  - 480,189 users
  - 17,770 movies
- WeChat
  - 474,726 groups
  - 245,352,140 users
- Spotify
  - 40 million songs
- OkCupid
  - 10 million members



Evaluating Recommender Systems

- Accuracy of predictions
  - Root Mean Square Error (RMSE)
    - E.g. Netflix grand prize $1M
  - Mean Absolute Error (MAE)

$$RMSE_{\mathcal{T}} = \sqrt{\sum_{(u_i,v_j)\in\mathcal{T}} (R_{i,j} - \hat{R}_{i,j})^2 \Big/ |\mathcal{T}|}$$

- Accuracy of classifications
  - Hit Rate/Ratio (HR)
  - Precision, Recall, F1, ROC curves

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

- Accuracy of ranks
  - Mean Reciprocal Rank (MRR)
  - Normalized Discounted Cumulative Gain (NDCG)

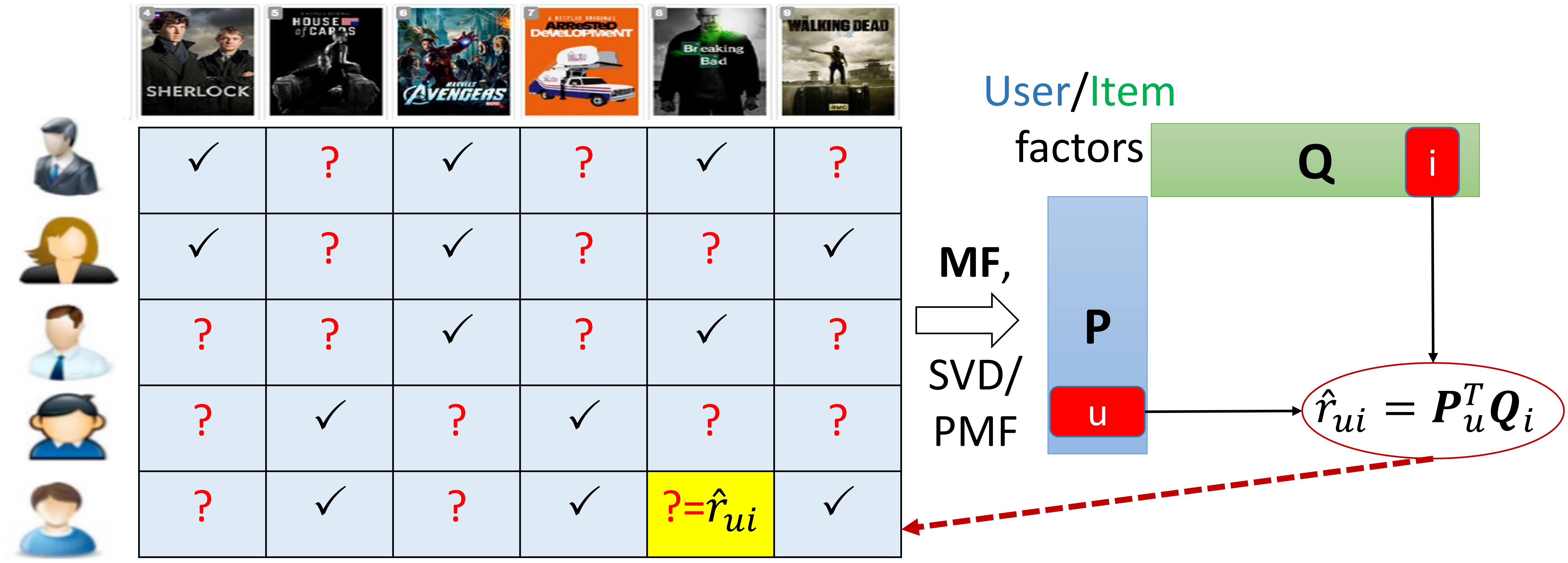
$$NDCG = \frac{1}{|\mathcal{U}|} \sum_{u\in\mathcal{U}} \frac{\log 2}{\log(p_u + 1)},$$

Herlocker et al. Evaluating Collaborative Filtering Recommender Systems. ACM TIS 2014

Collaborative filtering

Typical Methods: Matrix Factorization

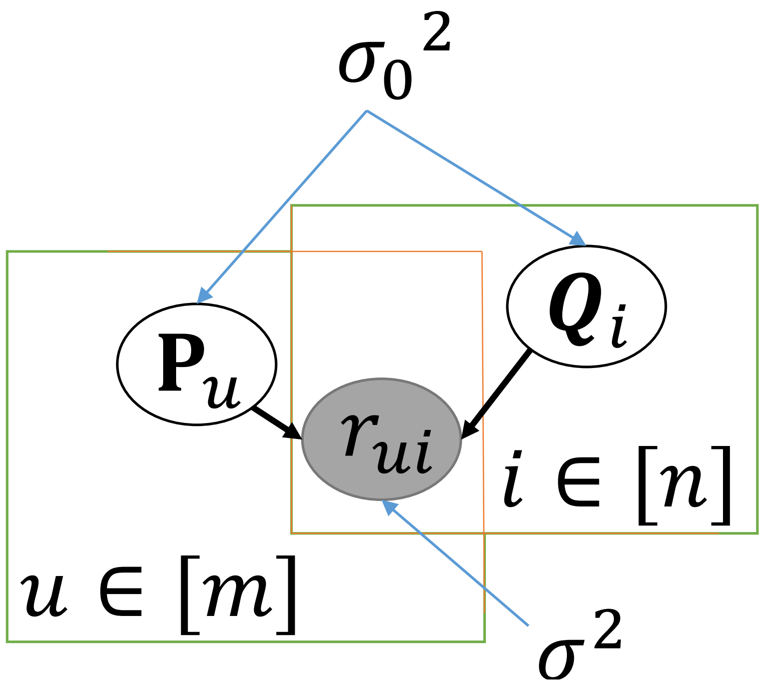
(Koren KDD’08, KDD 2018 TEST OF TIME award)



Probabilistic Interpretations: PMF

- The objective of matrix factorization

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



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

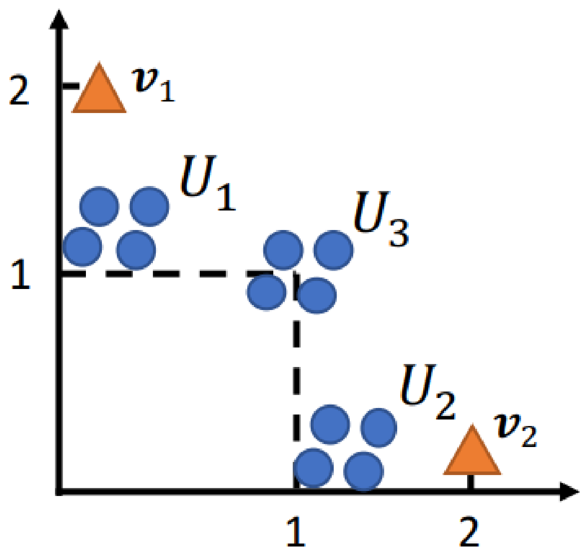
$$\ln p(\Theta|\boldsymbol{R}, \Phi) = -\frac{1}{2\sigma^2}\sum_{u,i}\delta(r_{ui})(r_{ui} - \boldsymbol{P}_u^T\boldsymbol{Q}_i)^2 - \frac{1}{2\sigma_0^2}\left(||\boldsymbol{P}||^2_{Frob} + ||\boldsymbol{Q}||^2_{Frob}\right)$$

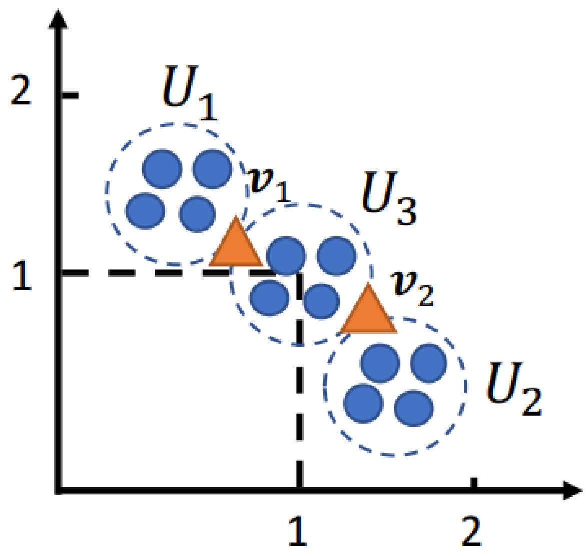
- Maximum a posteriori (MAP) estimation ←→ Minimizing sum-of-squared-errors with quadratic regularization (Loss + Regu)

Mnih & Salakhutdinov. Probabilistic matrix factorization. NIPS'07

Limitations of MF: Transitivity

- 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





Hsieh et al. Collaborative metric learning. WWW'17

Metric Learning: Replace Inner Products in MF with (Euclidean) Distances

- An item users liked will be closer to them than other items they did not like

$$d(i, j) = \|\mathbf{u}_i - \mathbf{v}_j\|,$$

- Hinge loss (margin-based)
  - For items user likes, their gradients move inward. For other items, their gradients move outward until they are pushed out by a safe margin

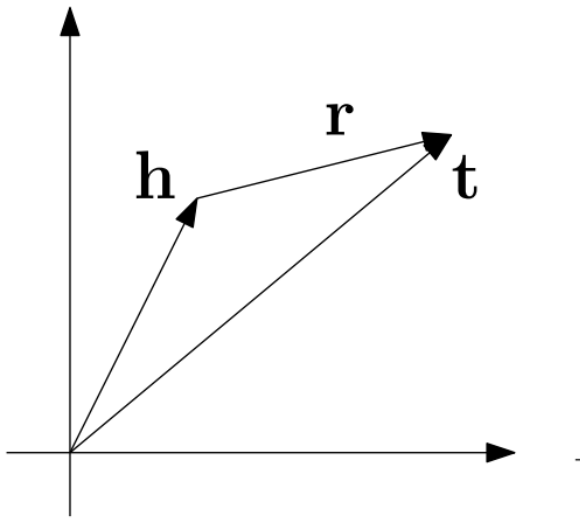
$$\mathcal{L}_m(d) = \sum_{(i,j)\in\mathcal{S}} \sum_{(i,k)\notin\mathcal{S}} w_{ij}[m + d(i,j)^2 - d(i,k)^2]_+,$$

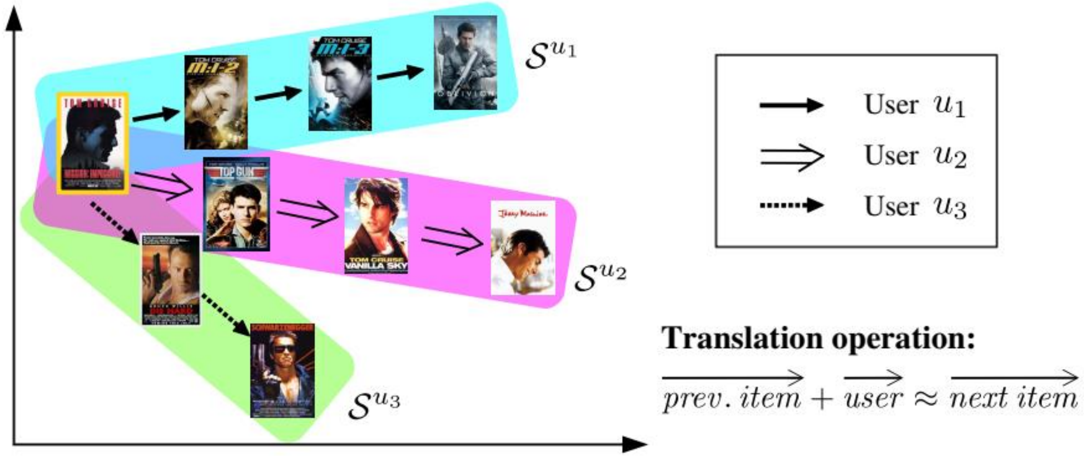
- Rank-based weighting scheme
  - Penalizes a positive item at a lower rank heavily than one at the top

$$w_{ij} = log(rank_d(i,j) + 1).$$

Translation-based Recommendation: Capture Sequential Behavior

- Inspired by advances in knowledge graph completion
  - Entities as points and relations as translation vectors
- Items as "entities", users as "relations" from one item to another





Bordes et al, Translating embeddings for modeling multi-relational data, NIPS'13
He et al, Translation-based Recommendation, IJCAI'18

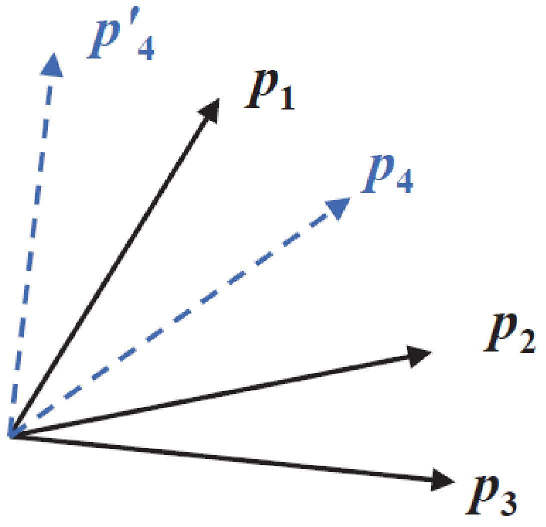
Limited Expressiveness of MF: Nonlinearity

- Similarity of given user u4:
  - Given: <u>Sim(u4,u**1**) > Sim(u4,u**3**) > Sim(u4,u**2**)</u>
  - Q: Where to put the latent factor vector p4?
- MF can not capture highly nonlinear
  - Deep learning, nonlinearity

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	0	1
u_2	0	1	1	0	0
u_3	0	1	1	1	0
u_4	1	0	1	1	1

items

users



Xiangnan He et al. Neural collaborative filtering. 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

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

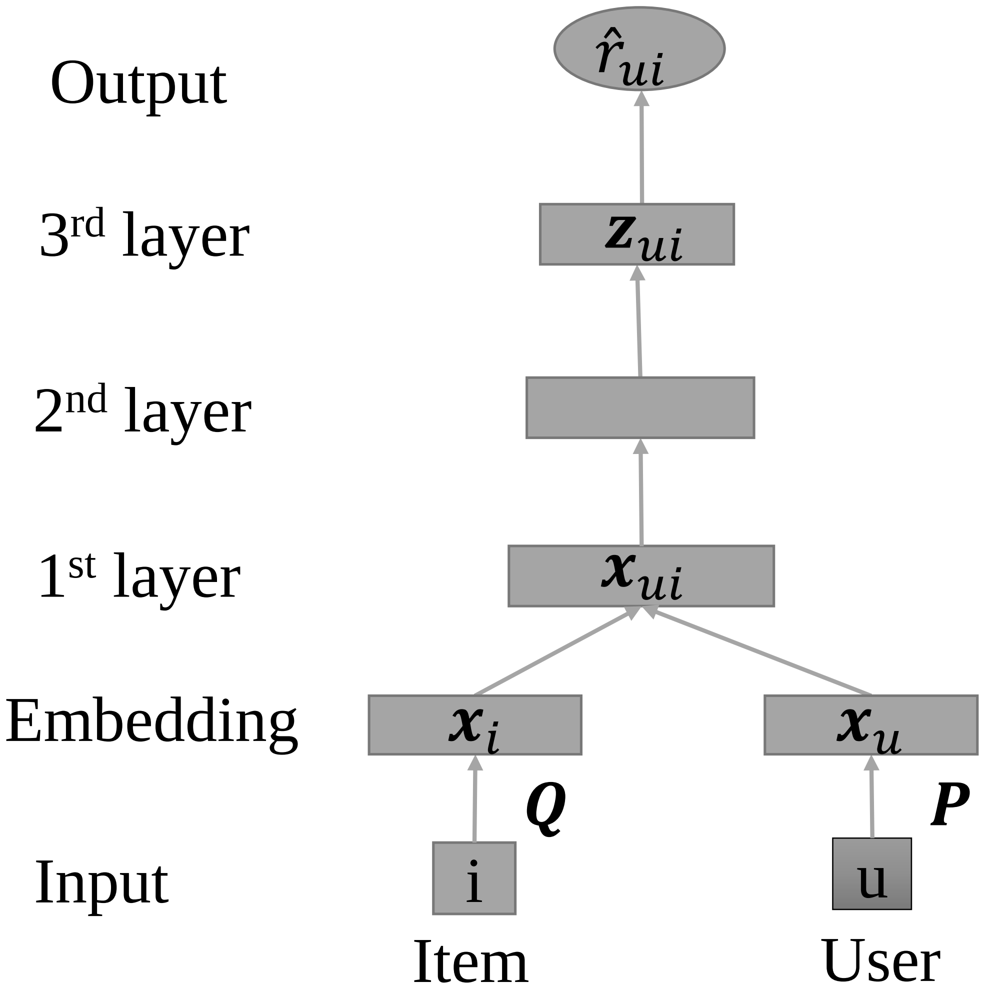
identity activation **all-one vector**

Go Deeper: Neural Collaborative Filtering

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

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

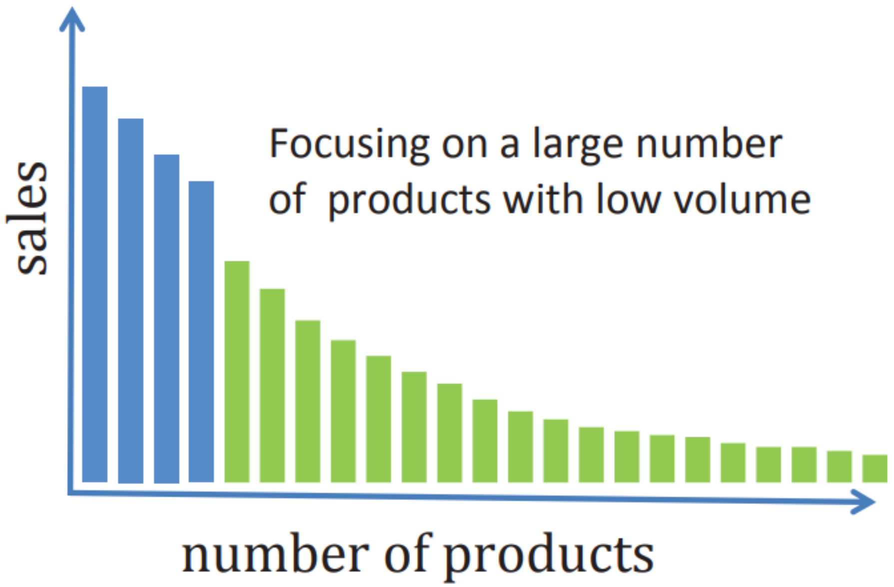
- Capture the complex user-item interaction relationships via the expressiveness of multilayer NNs



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

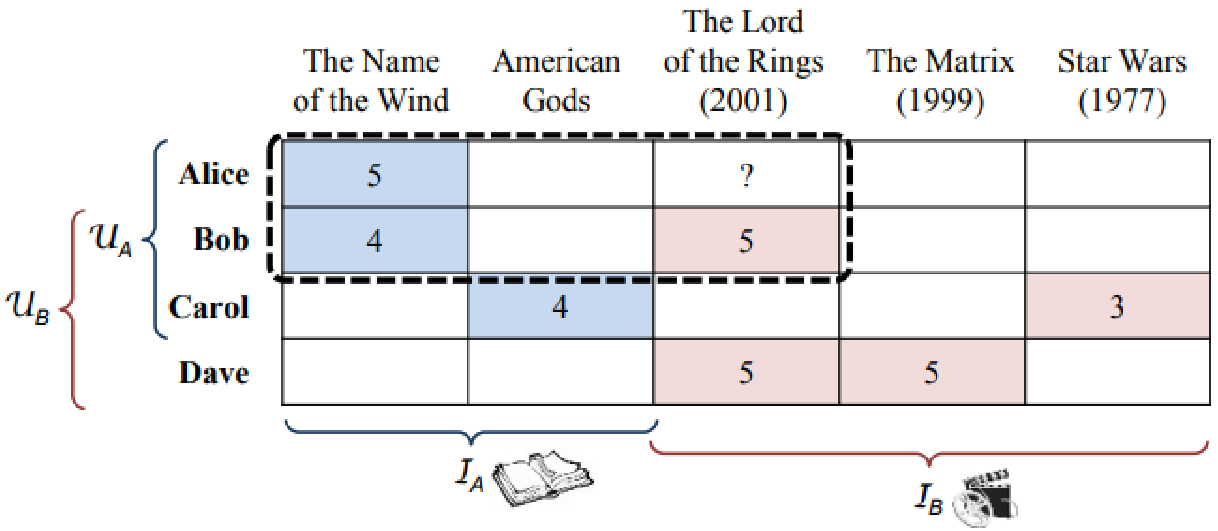
2	?	2	?	5	?
5	?	4	?	?	1
?	?	5	?	2	?
?	1	?	5	?	?
?	5	?	1	?	4



Cross-domain recommendation

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

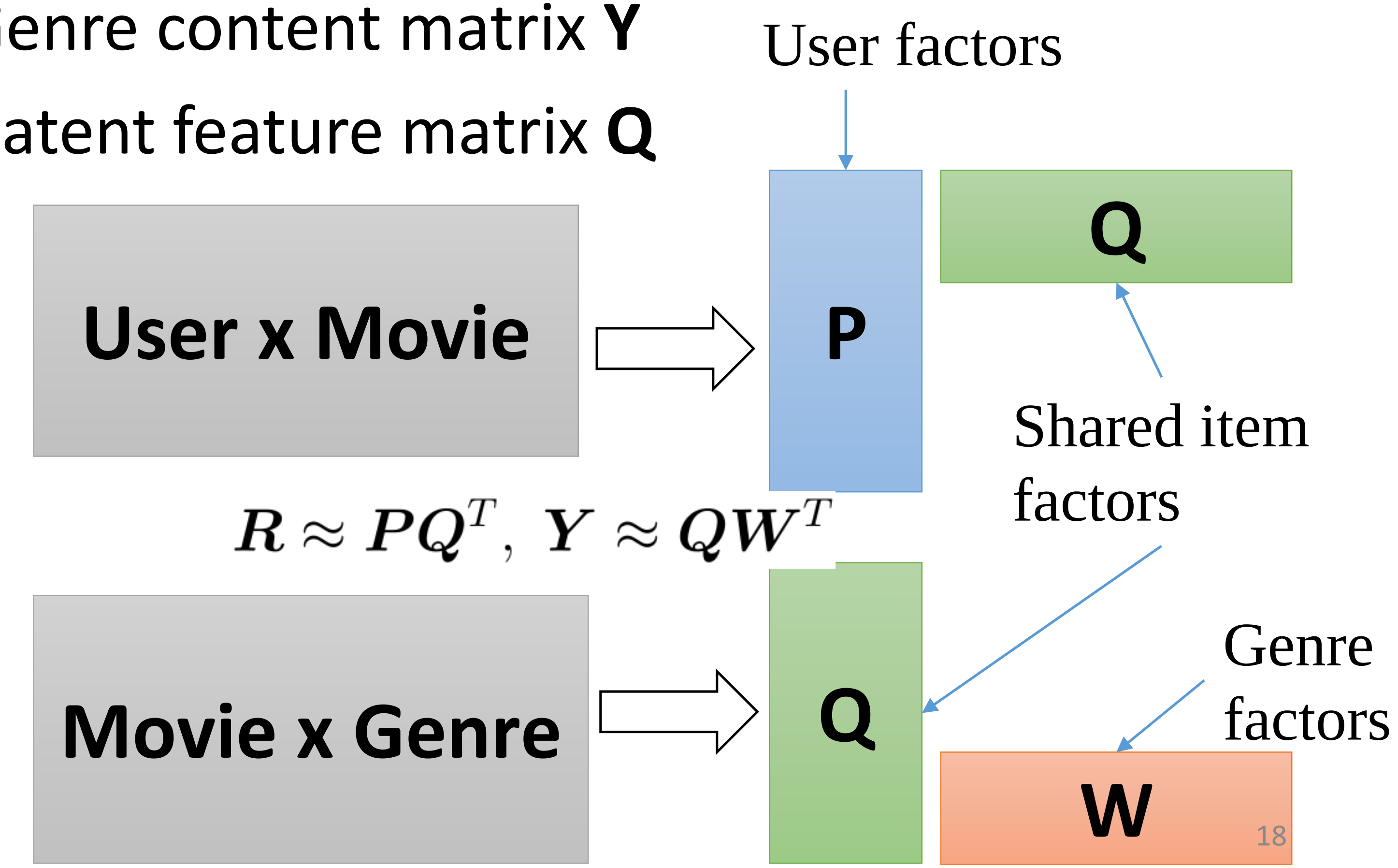


$$\hat{r}_{ui} \triangleq p(r_{ui} = 1 | u, [j]^u)$$

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



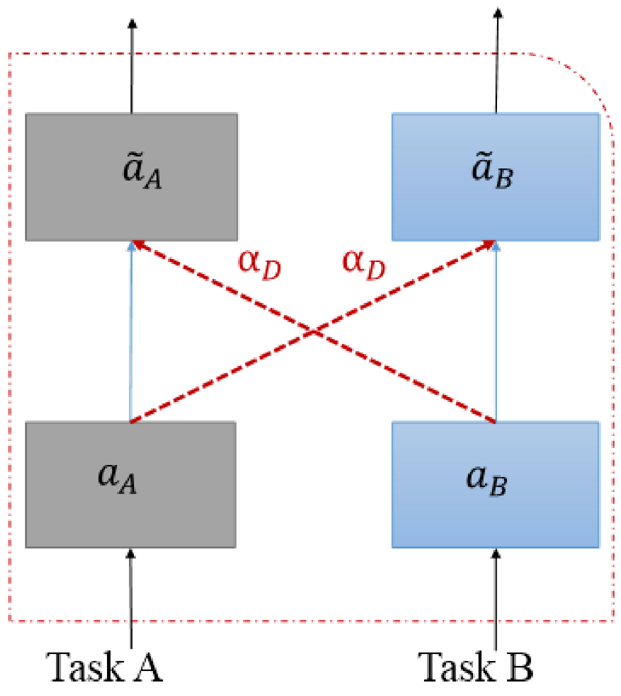
$$R \approx PQ^T, \; Y \approx QW^T$$

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



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

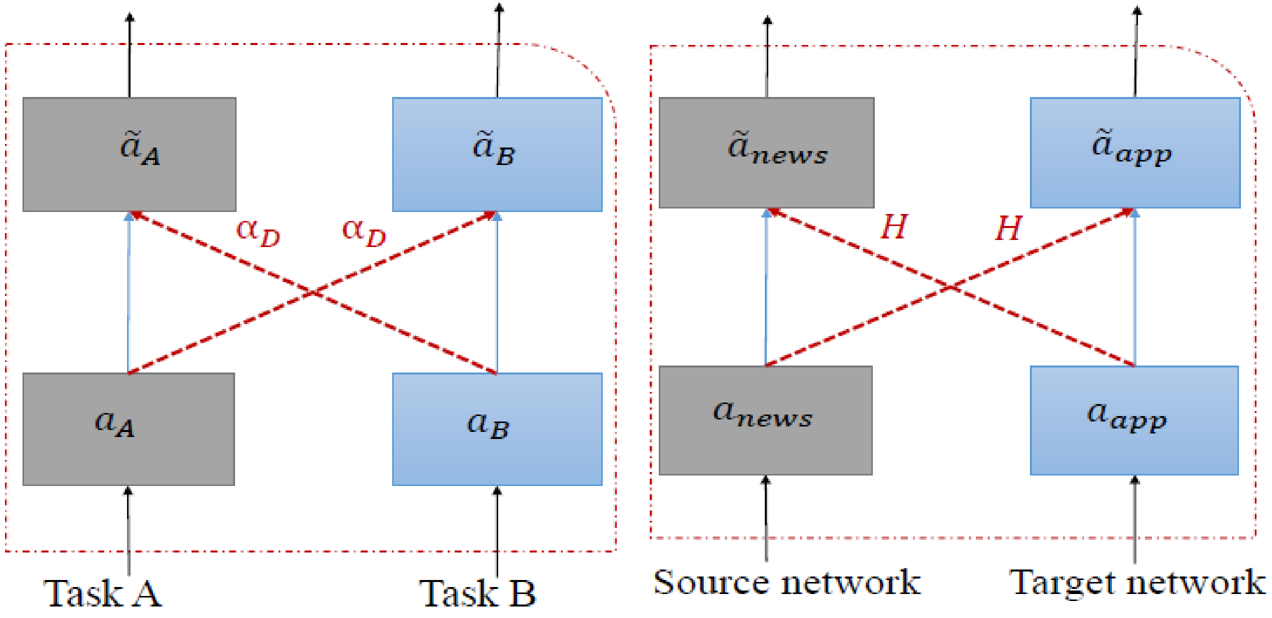
Collaborative Cross Networks (CoNet)

- A novel deep transfer learning method
- Alleviate the data sparsity issue faced by deep collaborative filtering
  - By transferring knowledge from a related source domain
- Relax strong assumptions faced by existing cross-domain recommendation
  - By transferring knowledge via a matrix and …
  - …enforcing sparsity-induced regularization

Hu et al. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. CIKM'18

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

- We can relax the <u>SA 1</u> assumption (equally important)

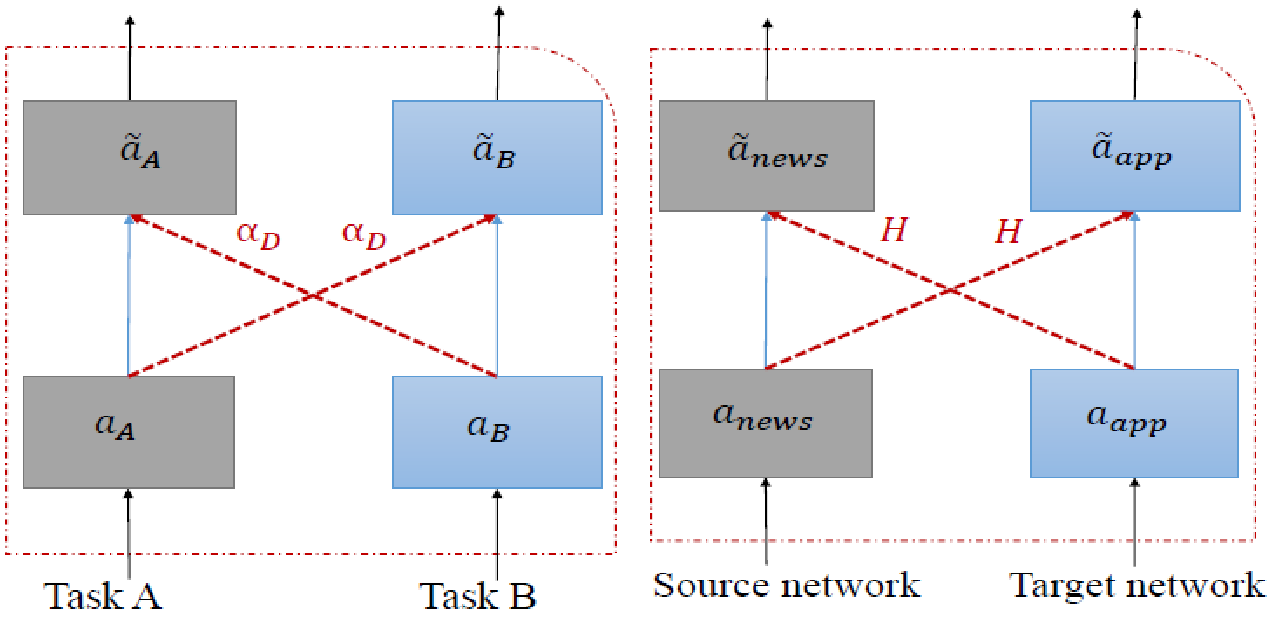


(a) Cross-stitch unit (b) Cross connection unit

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

Idea 2: Selecting representations via sparsity-induced regularization

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

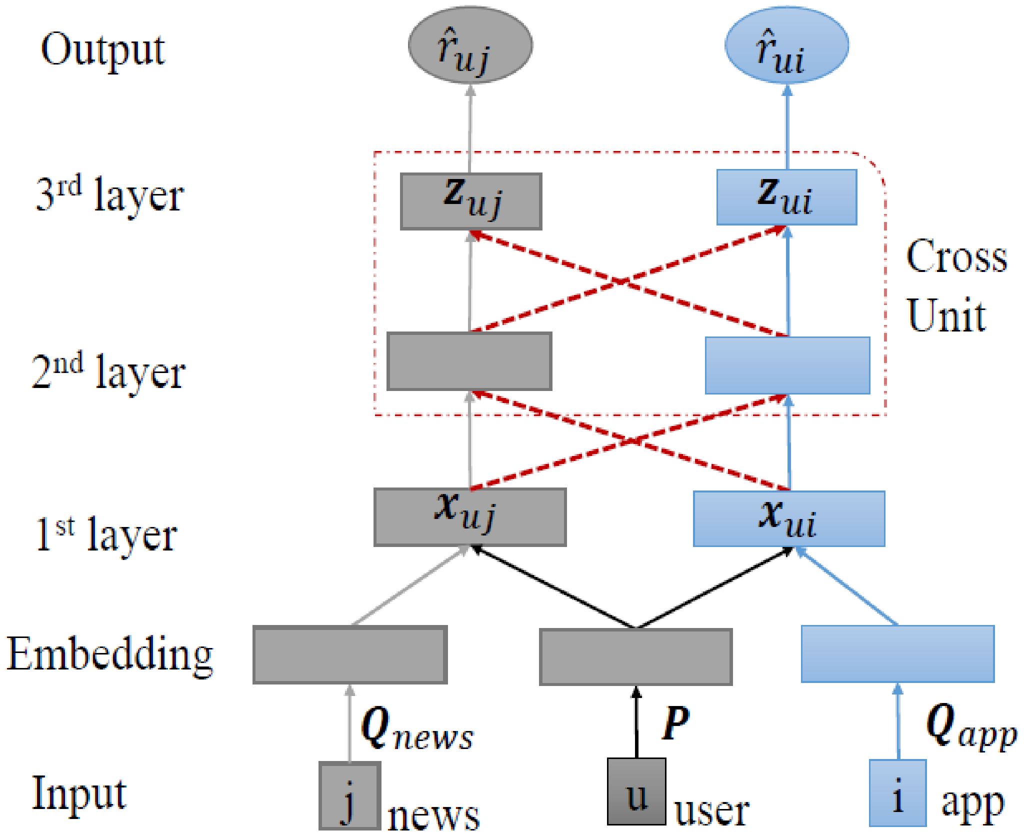


(a) Cross-stitch unit (b) Cross connection unit

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

Architecture of CoNet

- 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

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

- 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 Tensor Flow (https://www.tensorflow.org) provides automatic differentiation which can be computed by chain rule in back-propagation.

Complexity Analysis

- Model analysis

The model parameters Θ 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%

- Cheetah 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]

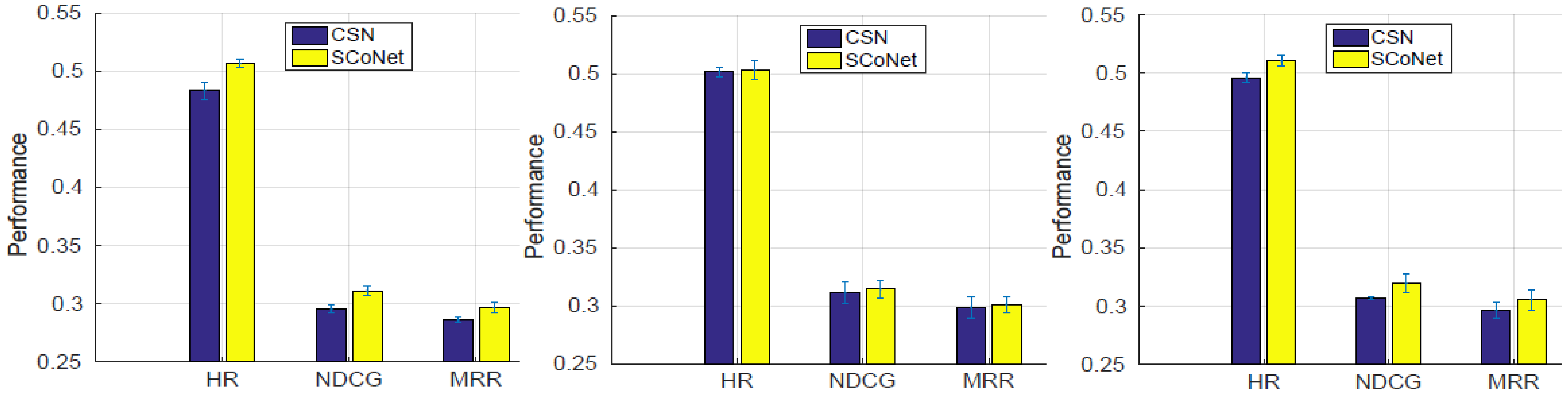
Comparing Different Approaches

- 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



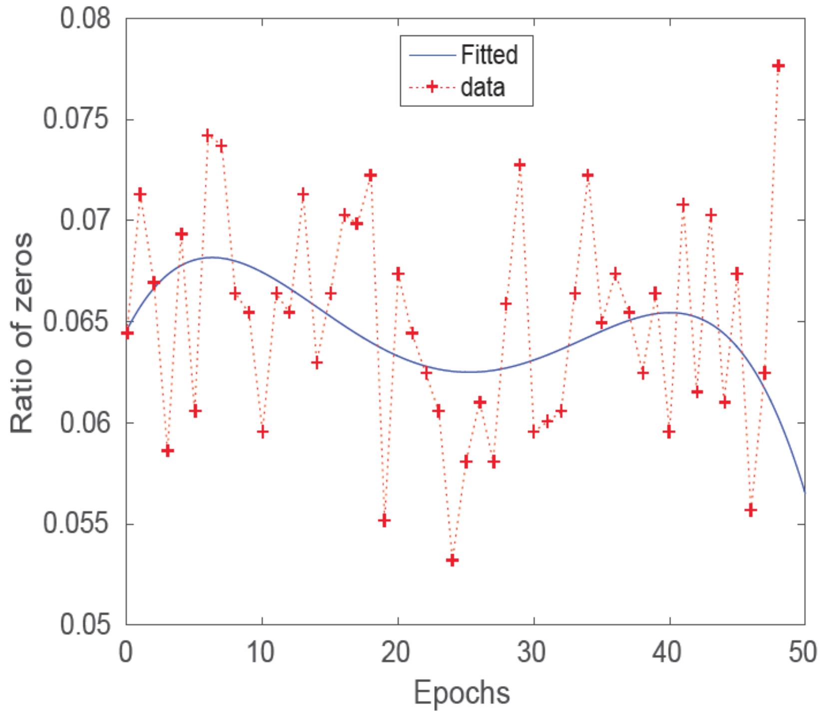
Benefit of Transferring Knowledge

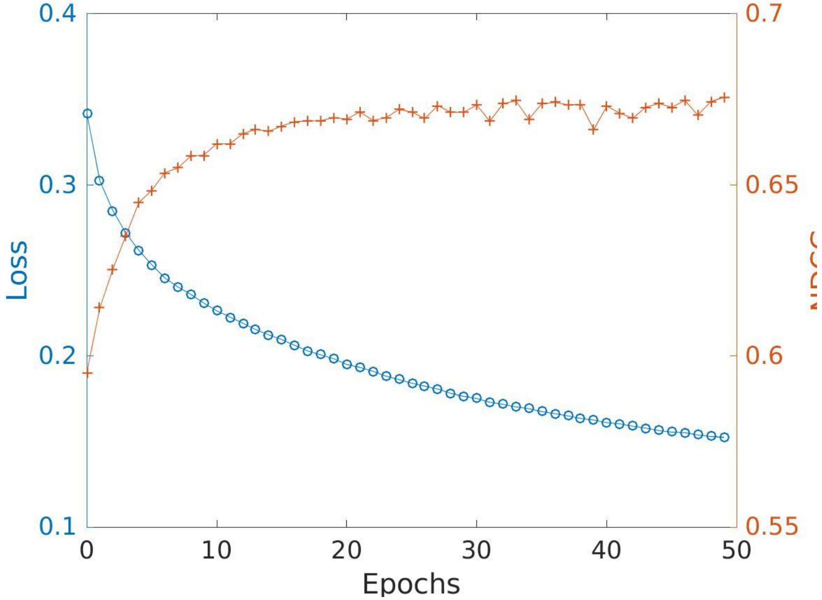
- The more training examples we can reduce, the more benefit we can get from transferring knowledge
- Our model can reduce tens of thousands training examples by comparing with non-transfer methods without performance degradation

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





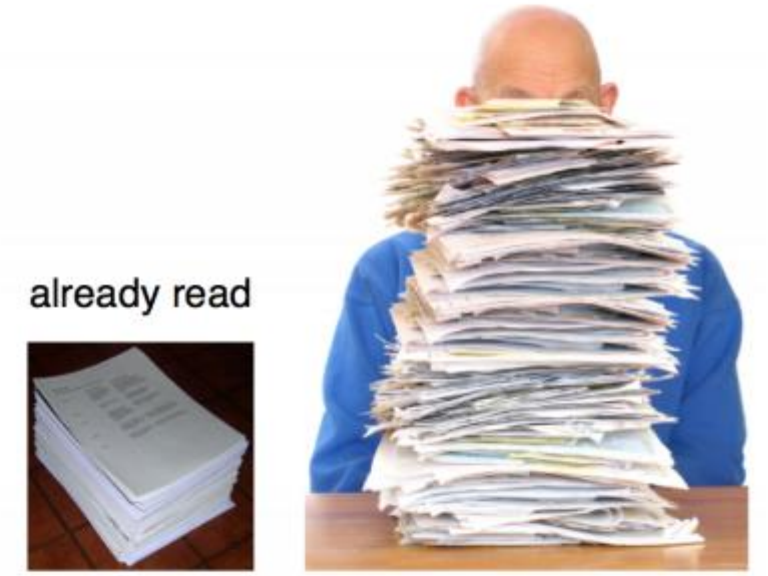
Summary

- 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,

Hybrid filtering

Another Solution: Hybrid Filtering (Collaborative + Content)

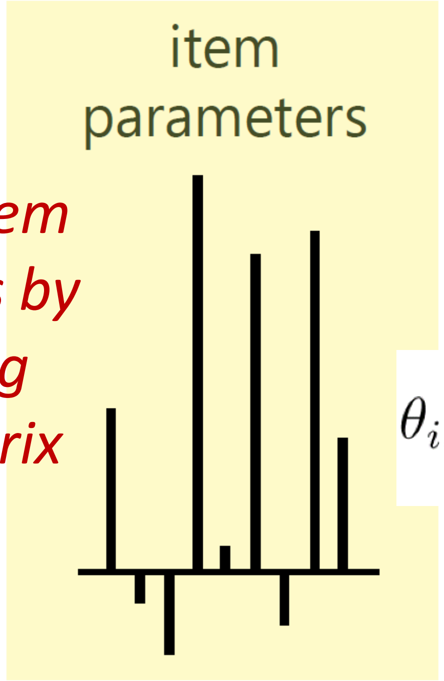
- Item reviews justify ratings
- Item content reveals topic semantics



Topic Modelling: Hidden Factors & Topics

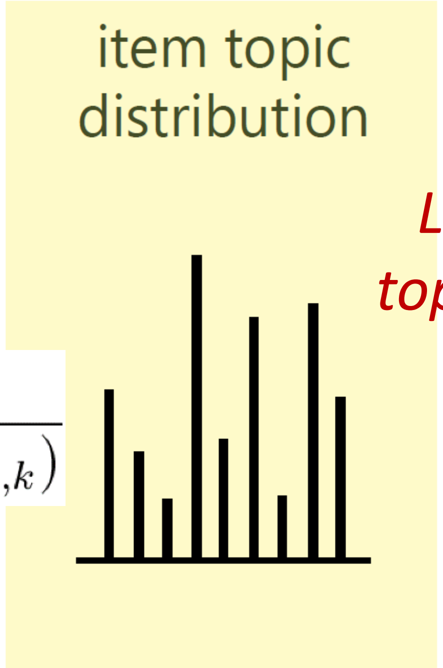
- using a transform that aligns latent rating and review terms, so that both are determined by a single parameter

Learning item parameters by factorizing rating matrix



transform

$$\theta_{i,k} = \frac{\exp(\kappa\gamma_{i,k})}{\sum_{k'} \exp(\kappa\gamma_{i,k})}$$



Learning item topic distribution by topic modeling

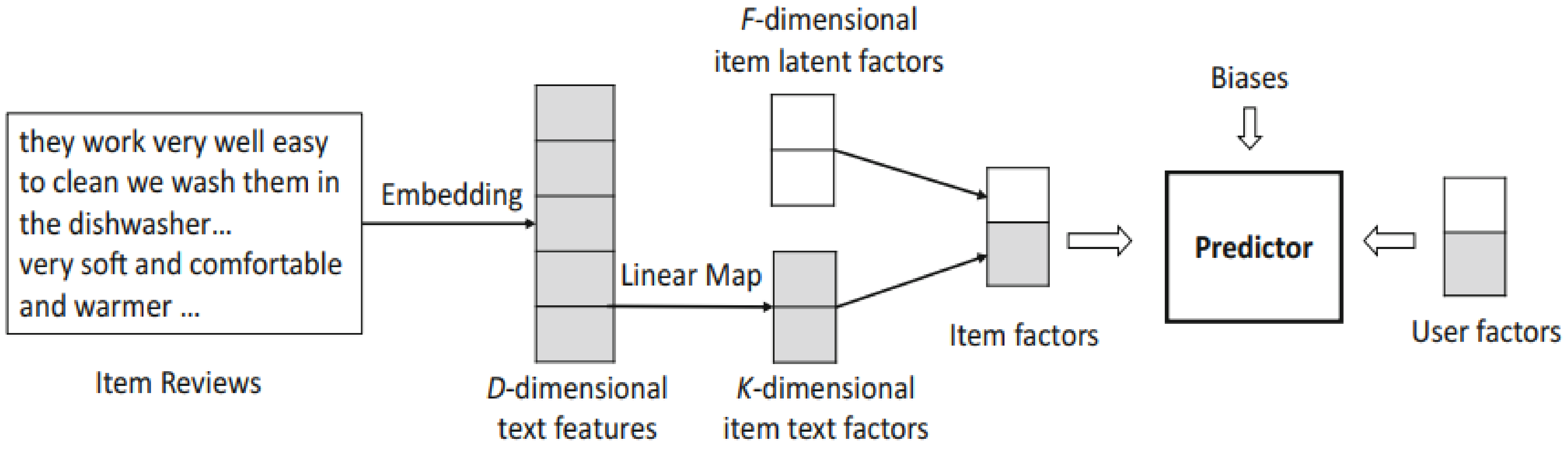
$$\gamma_i \in \mathbb{R}^K$$

$$\theta_i \in \Delta^K \text{ (i.e., } \sum_k \theta_{i,k} = 1)$$

McAuley & Leskovec, Hidden factors and hidden topics, RecSys'13

Pre-extracted Word-embedding Features

- Basic MF factorizes ratings into user/item *latent* factors
- Another MF factorizes reviews into user/item *text* factors



$$f_i \equiv \frac{1}{|d_i|} \sum_{w \in d_i} \mathbf{e}_w \qquad P_u^{\top} Q_i + \theta_u^{\top} (H f_i)$$

Hu & Dai, Integrating Reviews into Personalized Ranking for Cold Start Recommendation, PAKDD'17

Personalized Neural Embeddings (PNE)

- The way of pre-extracted embeddings *separates* the extraction of text features from the learning of user-item interaction
- These two processes cannot benefit from each other and *errors* in the previous step maybe propagate to the successive steps
- PNE learns embeddings of users, items, and words jointly, and predict user preferences on items based on these learned representations
- PNE estimates the probability that a user will like an item by two terms — *behavior* factors and *semantic* factors

Hu, Personalized Neural Embeddings for Collaborative Filtering with Text, NAACL'19

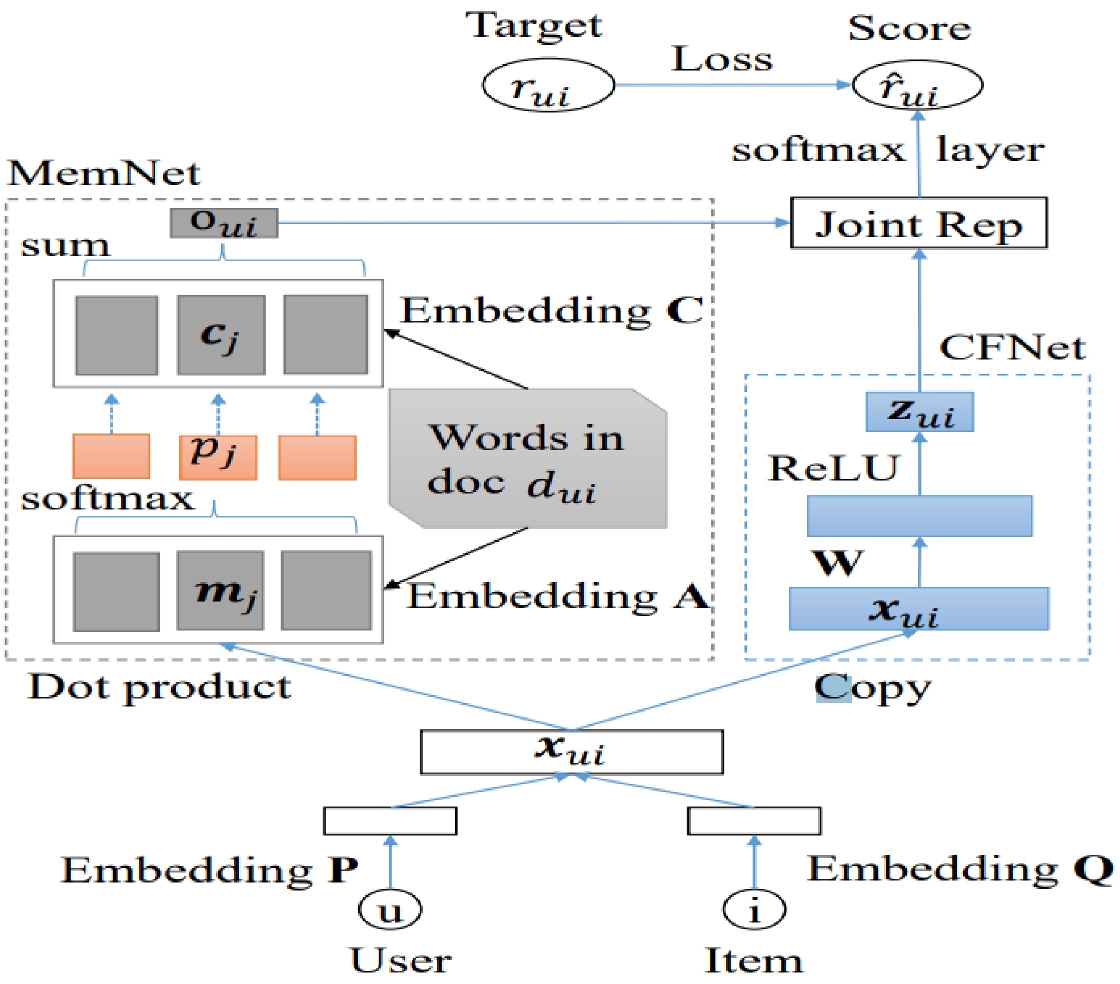
Architecture of PNE

- Behavior factors: same with neural CF

$$z_{ui}^{\text{behavior}} = \text{ReLU}(\boldsymbol{W}\boldsymbol{x}_{ui} + \boldsymbol{b})$$

- Semantic factors: relevance of a user to a word is learned by attention mechanism

$$z_{ui}^{\text{semantic}} = \sum_{j:\, w_j \in d_{ui}} \text{Softmax}(a_j^{u,i})\boldsymbol{c}_j$$



$$a_j^{u,i} = \boldsymbol{x}_{ui}^T \boldsymbol{m}_j^{u,i}$$

Dataset and Baselines

- Datasets
  - Amazon reviews
  - Cheetah news

Dataset	#user	#item	#rating	#word	#density	avg. words
Amazon	8,514	28,262	56,050	1,845,387	0.023%	65.3
Cheetah	15,890	84,802	477,685	612,839	0.035%	7.2

- Baselines

Baselines	Shallow method	Deep method
CF	BPR	MLP
CF w/ text	HFT, TBPR	LCMR, PNE (ours)

Comparing Different Approaches

- PNE vs MLP: Since CFNet of PNE is a neural CF (with one hidden layer), results show the benefit of exploiting unstructured text to alleviate the data sparsity issue faced by CF methods

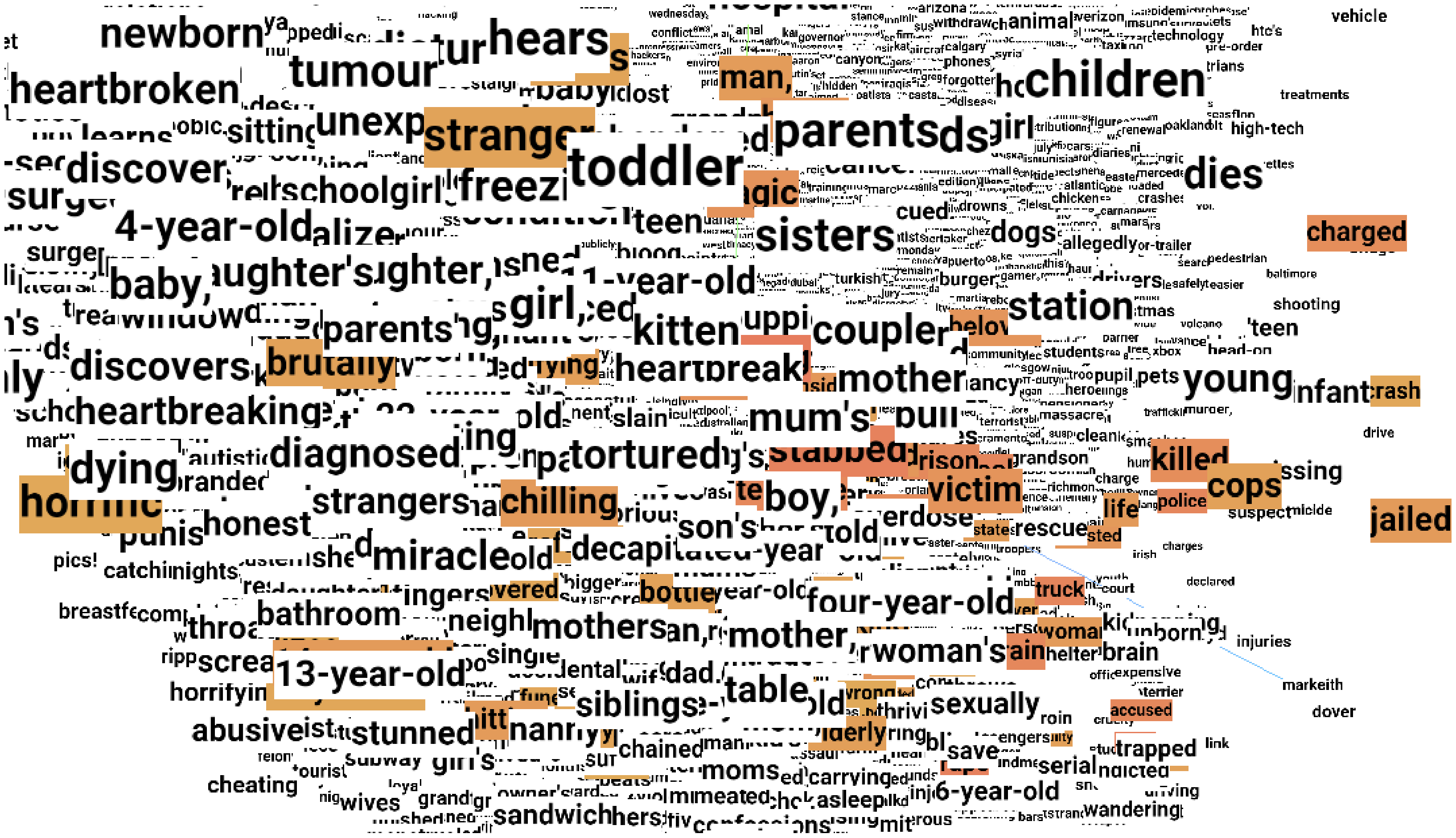
- PNE vs HFT/TBPR: Results show the benefit of integrating content text through MemNet (and exploiting interactions through neural CF)

TopK	Metric	Method					
		BPR	HFT	TBPR	MLP	LCMR	PNE
5	HR	8.10	10.77	15.17	21.00*	20.24	**23.52**
	NDCG	5.83	8.15	12.08	14.86*	14.51	**16.46**
	MRR	5.09	7.29	11.04	12.83*	12.63	**14.13**
10	HR	12.04	13.60	17.77	28.36*	28.36*	**31.86**
	NDCG	7.10	9.07	12.91	16.97*	16.78	**19.15**
	MRR	5.61	7.67	11.38	13.71*	13.56	**15.24**
20	HR	18.21	27.82	22.68	38.20	39.51*	**42.21**
	NDCG	8.64	12.52	14.14	18.99	19.18*	**21.75**
	MRR	6.02	8.54	11.71	14.26*	14.20	**15.95**

- PNE vs LCMR: Since MemNet of PNE is the same with **L**ocal MemNet of LCMR (with one-hop), results show the design of CFNet of PNE is more reasonable than that of **C**entralized MemNet of LCMR

PNE Learns Meaningful Word Embeddings

- Nearest neighbors of <u>*drug*</u>: *shot, shoots, gang, murder, killing, rape, stabbed, truck, school, police, teenage*
- Google word2vec: drugs, heroin, addiction, abuse, fda, alcoholism, cocaine, lsd, alcohol, schedule, substances



Transfer meets hybrid

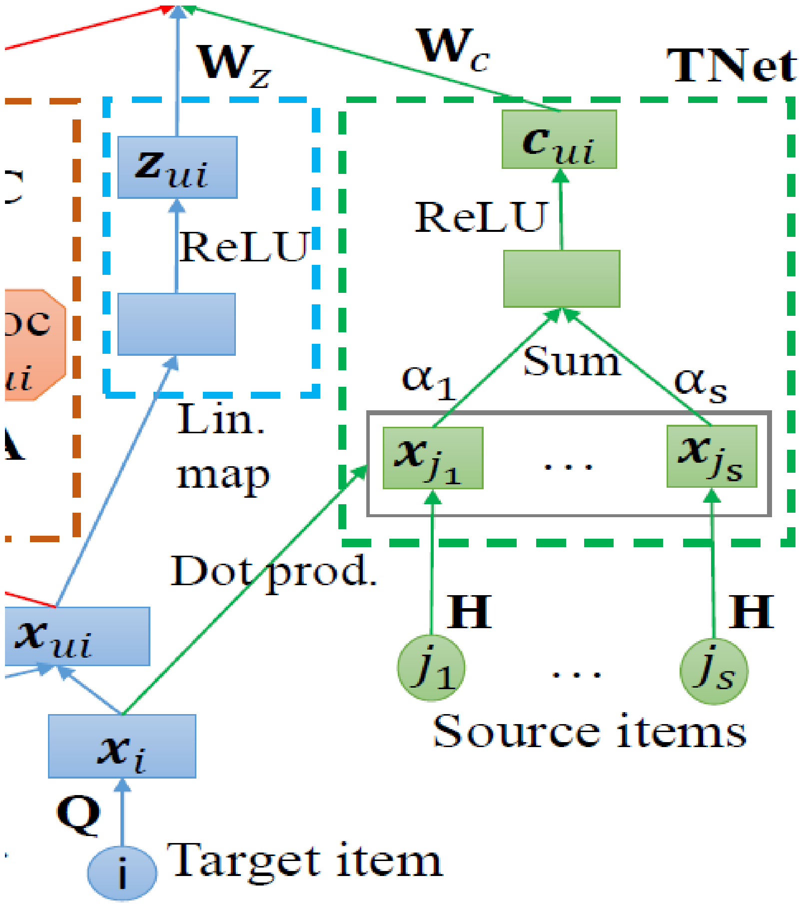
Transfer Meets Hybrid: A Synthetic Approach for Cross-Domain Collaborative Filtering with Text

- Hybrid filtering methods integrate content information, e.g. product reviews and news titles
- Cross-domain methods leverage knowledge from a related domain, e.g. from Apps to News
- TMH attentively extracts useful content from unstructured text via a memory network and …
- … selectively transfers knowledge from a source domain via a transfer network

Hu et al, Transfer meets hybrid, WWW’19 short

Architecture of TMH

- A MemNet: Matching Word Semantics with User Preferences
  - Same with MemNet of PNE and Local MemNet of LCMR
- A TransNet: Selecting Source Items to Transfer by a way of coarse-to-fine
  - *Coarse*: transfer source items such that this user has interacted in source domain
  - *Fine*: similarities between target item and coarse source items by content-based addressing
  - Finally: transfer vector is a weighted sum of the corresponding source item embeddings



$$\boldsymbol{c}_{ui} = \text{ReLU}(\sum\nolimits_j \alpha_j^{(i)} \boldsymbol{x}_j)$$

Datasets

Dataset	Domain	Statistics	Amount
Mobile News	Shared	#Users	15,890
	Target	#News	84,802
		#Reads	477,685
		Density	0.035%
		#Words	612,839
		Avg. Words Per News	7.2
	Source	#Apps	14,340
		#Installations	817,120
		Density	0.359%
Amazon Product	Shared	#Users	8,514
	Target	#Clothes (Men)	28,262
		#Ratings/#Reviews	56,050
		Density	0.023%
		#Words	1,845,387
		Avg. Words Per Review	32.9
	Source	#Products (Sports)	41,317
		#Ratings/#Reviews	81,924
		Density	0.023%

Baselines

Baselines	Shallow method	Deep method
Single-domain	BPRMF [41]	MLP [17]
Cross-domain	CDCF [31], CMF [42]	MLP++, CSN [34]
Hybrid	HFT [33], TextBPR [16, 20]	LCMR [19]
Cross + Hybrid	CDCF++	TMH (ours)

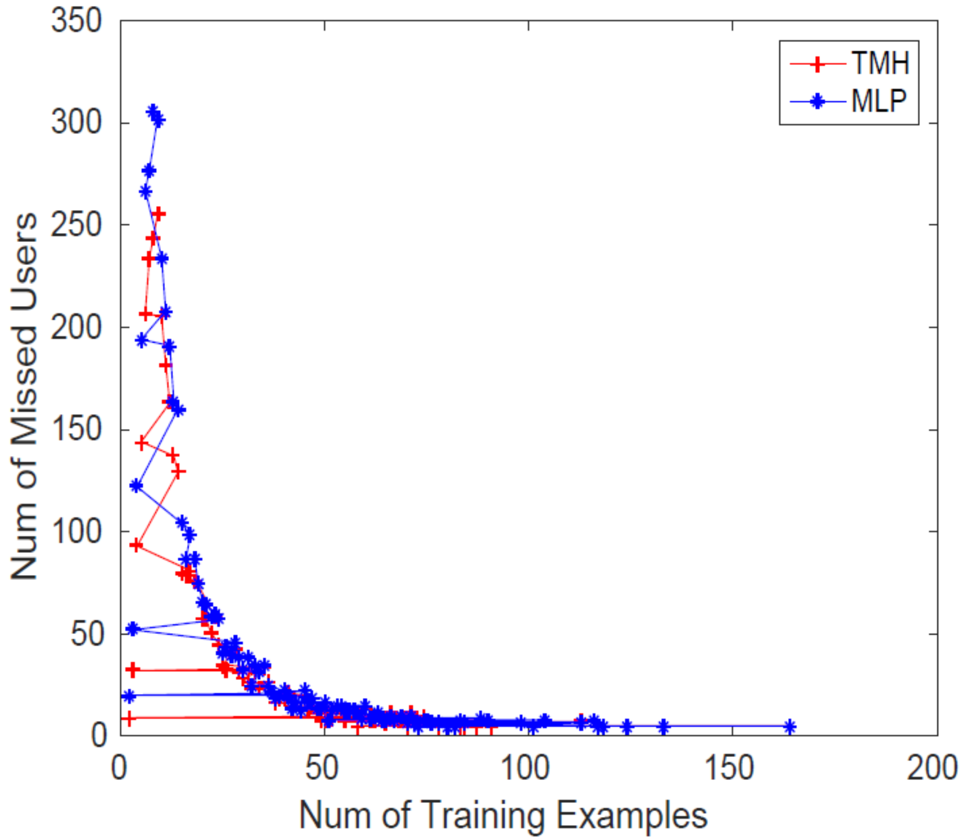
Results on Amazon Dataset

Method	$topK = 5$			$topK = 10$			$topK = 20$		
	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR
BPRMF	.0810	.0583	.0509	.1204	.0710	.0561	.1821	.0864	.0602
CDCF	.1295	.0920	.0797	.2070	.1167	.0897	.3841	.1609	.1015
CMF	.1498	.0950	.0771	.2224	.1182	.0863	.3573	.1521	.0957
HFT	.1077	.0815	.0729	.1360	.0907	.0767	.2782	.1252	.0854
TextBPR	.1517	.1208	.1104	.1777	.1291	.1138	.2268	.1414	.1171
CDCF++	.1314	.0926	.0800	.2102	.1177	.0901	.3822	.1605	.1016
MLP	.2100	.1486	.1283	.2836	.1697	.1371	.3820	.1899	.1426
MLP++	.2263	.1626	.1417	.2992	.1862	.1514	.3810	.2069	.1570
CSN	.2340*	.1680*	.1462*	.3018*	.1898*	.1552*	.3944*	.2091*	.1605*
LCMR	.2024	.1451	.1263	.2836	.1678	.1356	.3951	.1918	.1420
TMH	**.2575**	**.1796**	**.1550**	**.3490**	**.2077**	**.1666**	**.4443**	**.2311**	**.1727**
Our improve	10.04%	6.90%	6.01%	15.63%	9.43%	7.34%	12.65%	10.52%	7.60%

Improvement on Cold Users (and Items)

- Missed Hit Users (MHU) distribution on Cheetah Mobile

- We expect that cold users in MHUs can be reduced by using TMH
- The more amount we can reduce, the more effective that TMH can alleviate the cold-user start issues
- MHUs are most of cold users who have few training examples.
- #cold-users in MHUs of MLP is higher than that of TMH.
- TMH reduces #cold-users from 1,385 to 1,145 on Mobile, achieving relative 20.9% reduction



Future works

- Data privacy
  - Source domain cannot share the raw data, but model parameters

Thanks!

Q & A