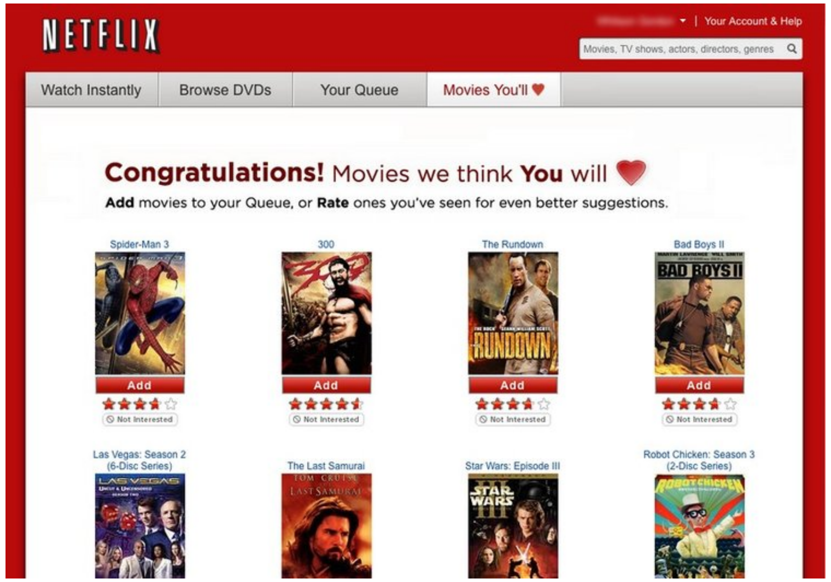
Collaborative Translational Metric Learning
[ICDM 2018]

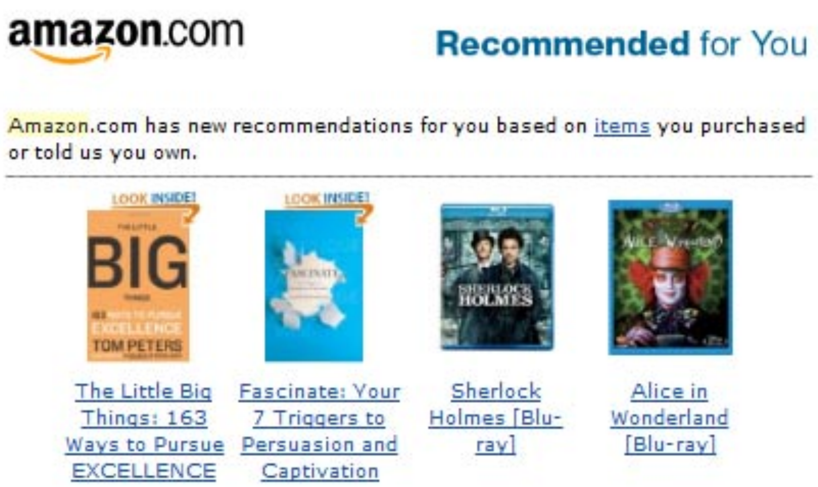
Chanyoung Park[1], Donghyun Kim[2], Xing Xie[3], Hwanjo Yu[1*]
Dept. of Computer Science and Engineering, POSTECH, South Korea[1]
Oath, USA[2]
Microsoft Research Asia, China[3]
{pcy1302, hwanjoyu}@postech.ac.kr, cartopy@gmail.com, xingx@microsoft.com

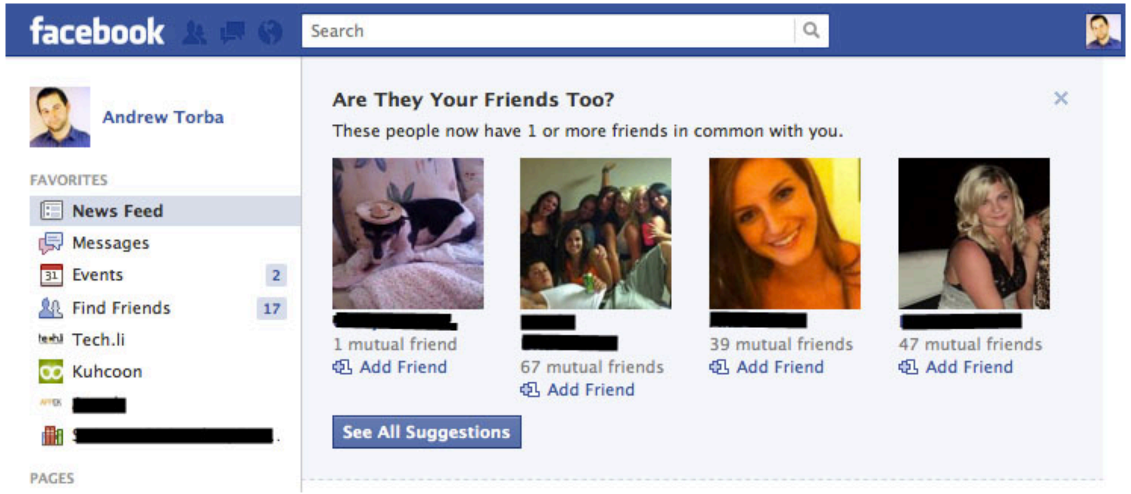
* Corresponding Author

Recommender System

- Movies
- Clothing
- Books
- Friends
- Citation
- Scientific paper
- News article
- TV programs

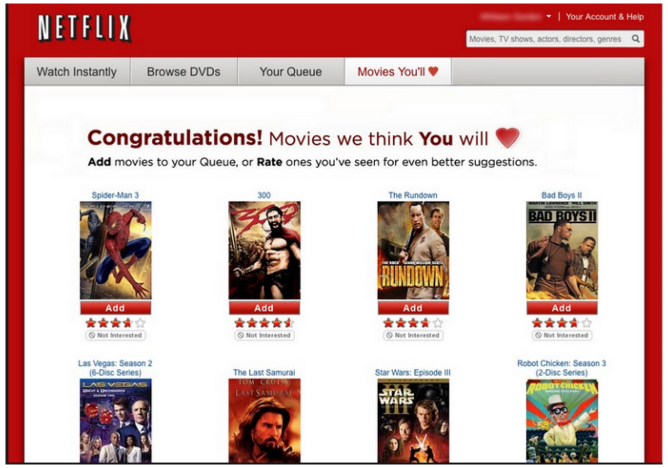






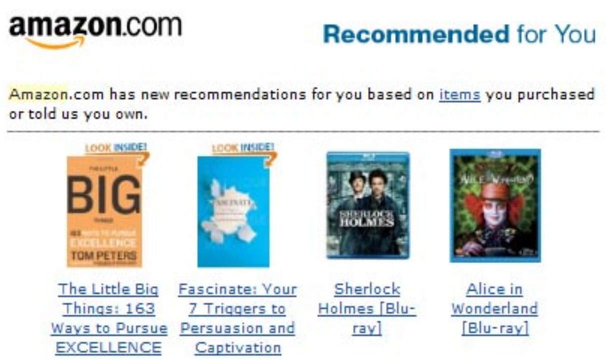
How useful is it?

- Want some evidence?



80% movies watched came from recommendation

[Gomez-Uribe et al, 2016]



30% page views came from recommendation

[Brent, 2017]

38% more click-through are due to recommendation

[Celma & Lamere, ISMIR 2007]

The value of Netflix recommendations is estimated at **more than US$1 billion per year**

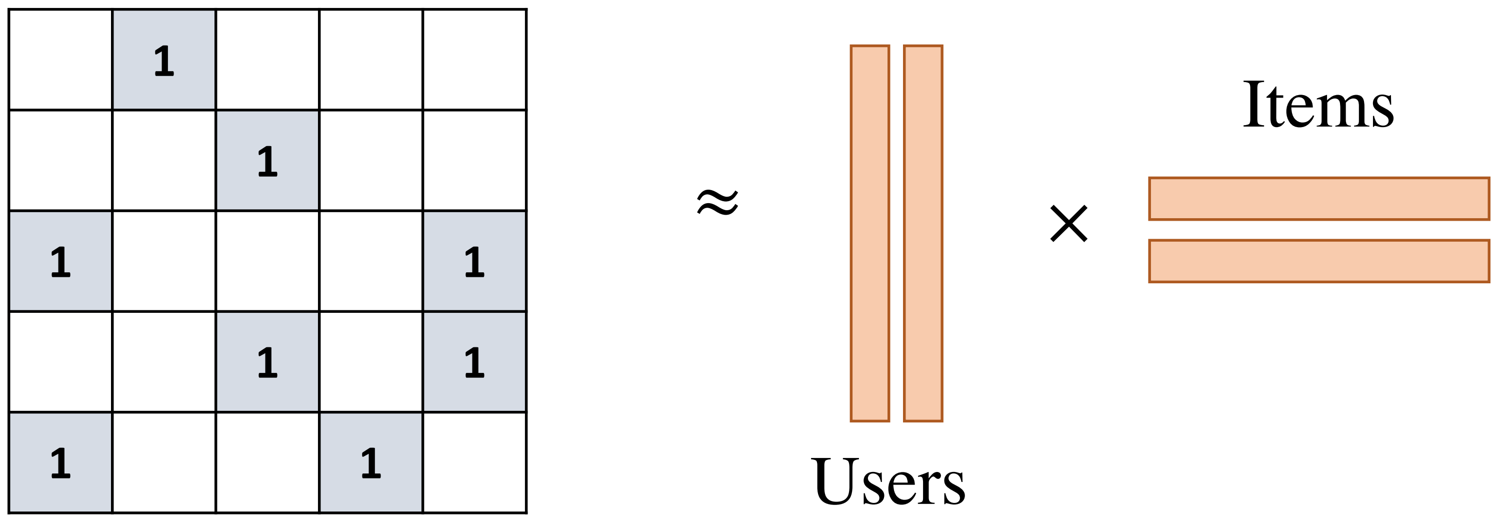
Implicit Feedback

- No explicit ratings
- Any type of interactions between users and items (abundant)

- Only positive feedback is available
- Not about rating prediction,
  - But about **modeling the relationships between different user/item pairs**

Matrix Factorization (MF)

- Matrix factorization-based recommendation methods are popular



MF violates "Triangle Inequality"

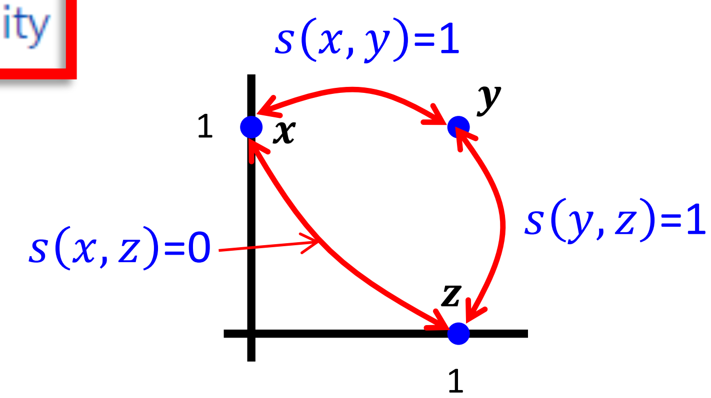
- MF is based on inner product operation, which violates **triangle inequality**
- A metric should satisfy…

1. $d(x, y) \geq 0$ — non-negativity or separation axiom
2. $d(x, y) = 0 \Leftrightarrow x = y$ — identity of indiscernibles
3. $d(x, y) = d(y, x)$ — symmetry
4. $d(x, z) \leq d(x, y) + d(y, z)$ — subadditivity or triangle inequality

$$s(x,z) \geq s(x,y) + s(y,z)$$

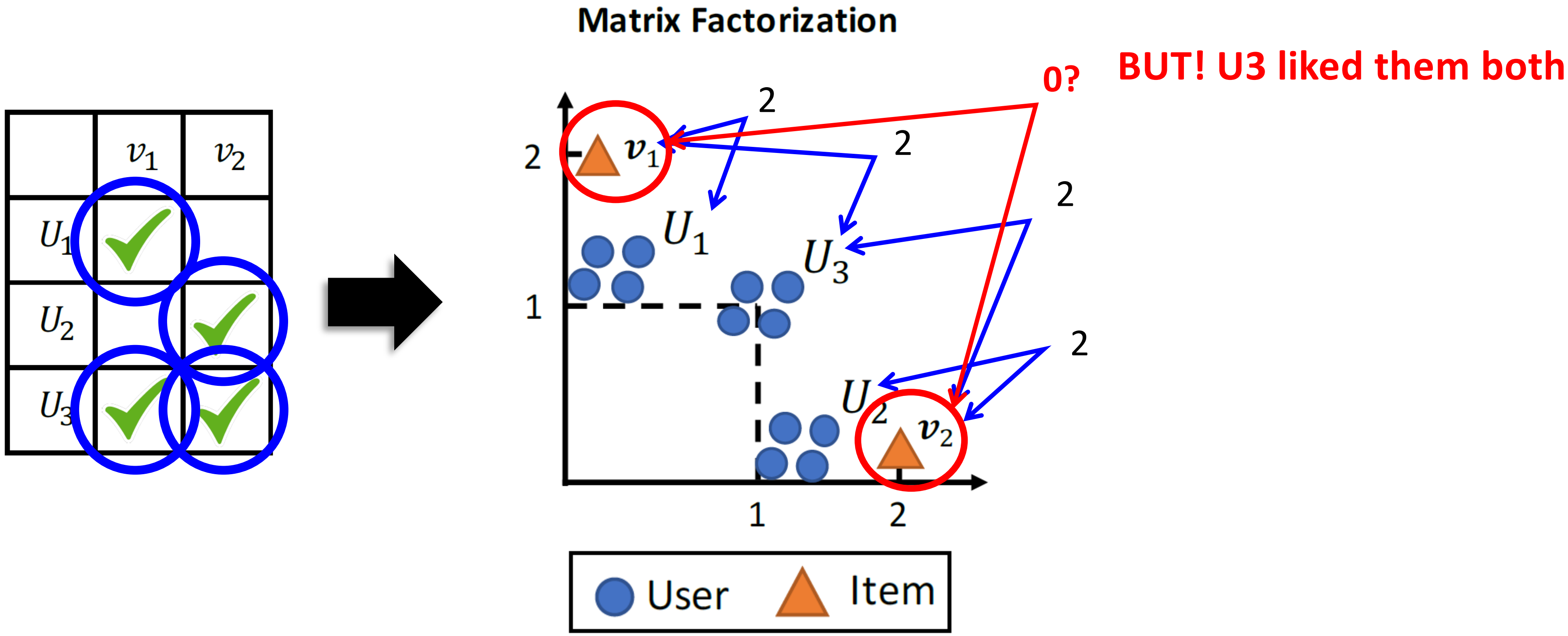
$$d(\cdot) = -s(\cdot)$$

$$s(x,z) \leq s(x,y) + s(y,z)$$



- Counter example
  - $x = [0,1], y = [1,1], z = [1,0]$

MF violates "Triangle Inequality"



Violates triangle inequality, therefore, positive relationships between (U3,v1) and (U3,v2) are not propagated to (v1,v2)

Source: Hsieh, Cheng-Kang, et al. "Collaborative metric learning." Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017.

Metric Learning Approach

- MF Fails to precisely **capture item-item and user-user similarity**

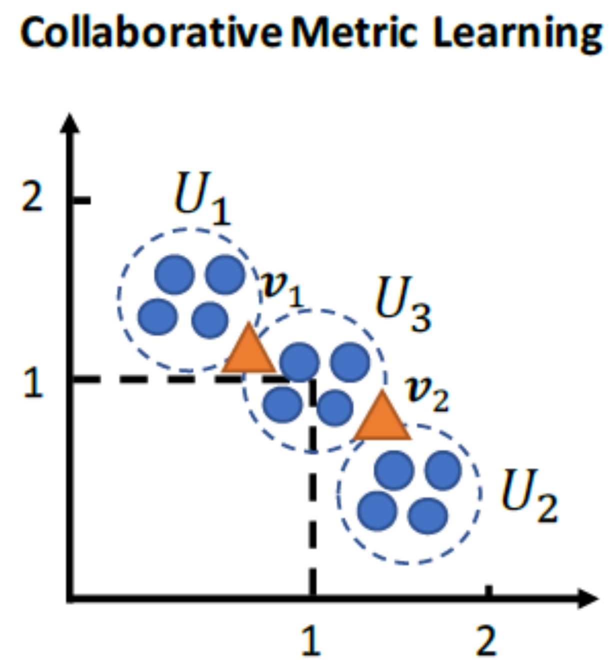
- **Solution: Metric learning approaches**
  - Project users and items into a low-dimensional **metric space**
    - Triangle inequality is satisfied
  - Minimize the distance between each user-item interaction in **Euclidean space**
    - [Recsys10, KDD12, IJCAI15, WWW17]

[WWW17] Collaborative Metric Learning (CML)

- User should be closer to the items the user likes than those the user does not

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

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



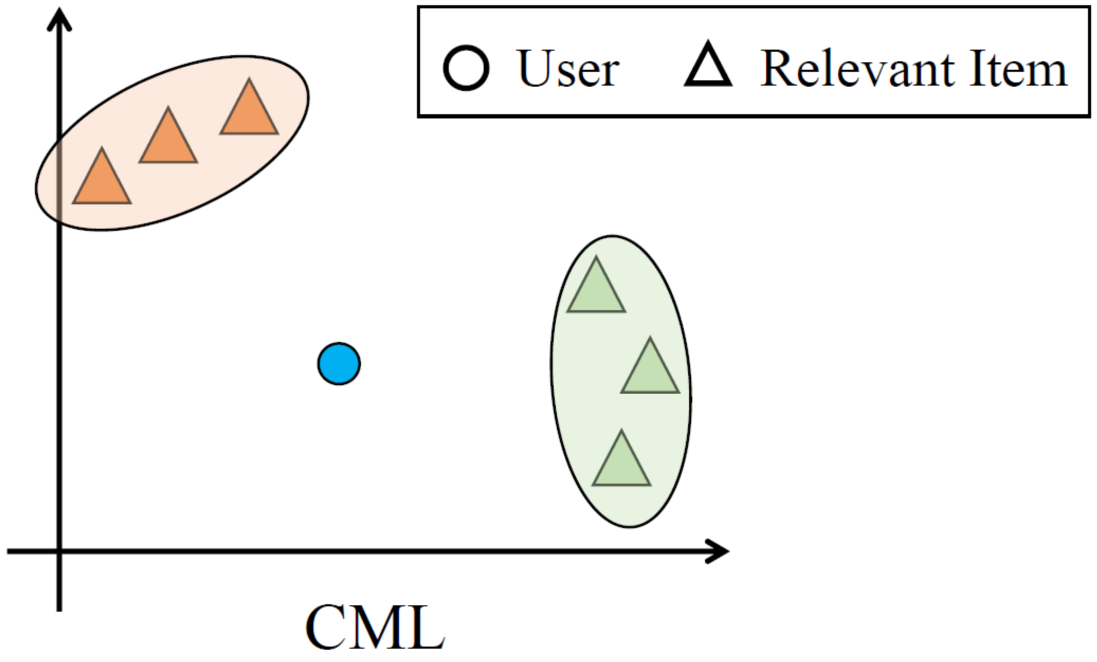
	v_1	v_2
U_1	✓	
U_2		✓
U_3	✓	✓

Expect to capture the similarity among user-user and item-item pairs

Source: Hsieh, Cheng-Kang, et al. "Collaborative metric learning." Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017.

Limitation of CML

- <u>**Each user is projected to a single point**</u> in the metric space



Hard to model the **intensity** and the **heterogeneity** of user–item relationships in implicit feedback

Intensity and Heterogeneity of Implicit Feedback

Intensity

- A user's implicit feedback does not indicate the equal preference
- Some of the items are more relevant to the user than others

Intensity of user-item relationships

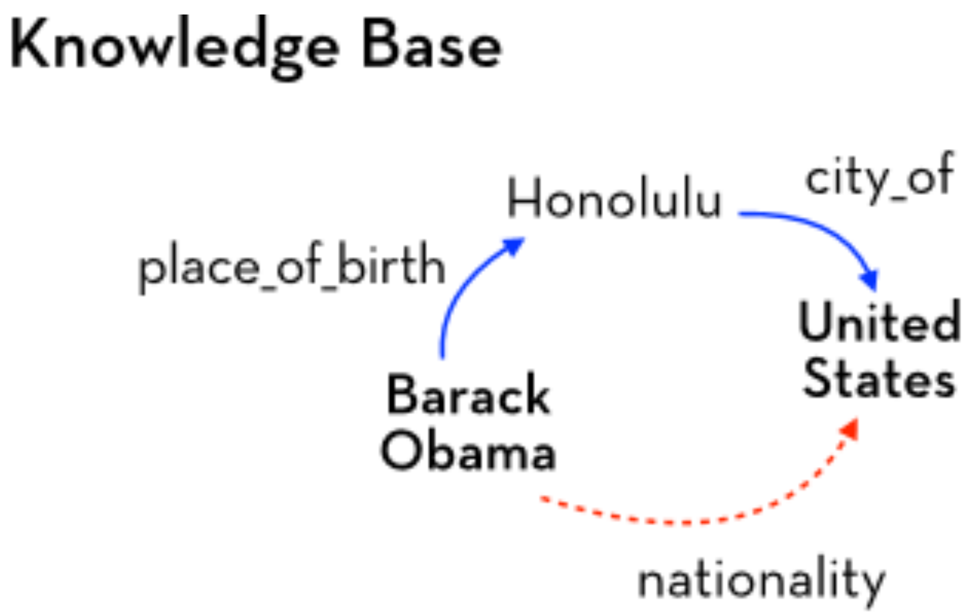
Heterogeneity

- A user may have a wide variety of tastes in different item categories
  - The type of user–item relationship is **heterogeneous** with regard to the user's tastes in various item categories

Preserving a user's **intense** and **heterogeneous** relationships with items is not easy when a user is projected to a single point

Solution: Adopt "translation mechanism"

- Effective for knowledge graph embedding
- Relations between entities are interpreted as **<u>translation operations</u>** between them
  - if a triplet (h, r, t) is true?
    - $[\vec{\boldsymbol{h}} + \vec{\boldsymbol{r}} \approx \vec{\boldsymbol{t}}]$: $\vec{\boldsymbol{t}}$ should be a nearest neighbor of $\vec{\boldsymbol{h}} + \vec{\boldsymbol{r}}$



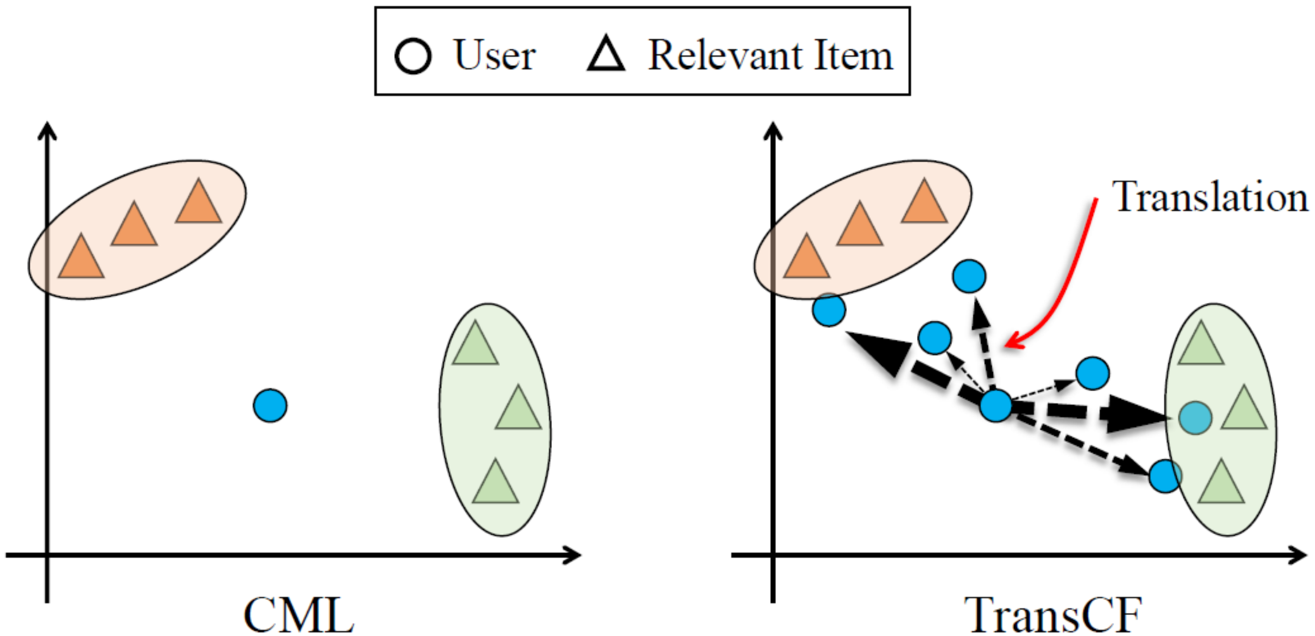
Example

- (Barack_Obama, place_of_birth, Honolulu)

$\overrightarrow{\textbf{Barack\_Obama}} + \overrightarrow{\textbf{place\_of\_birth}} \approx \overrightarrow{\textbf{Honolulu}}$

Translation vector

Translation mechanism



Intensity: Thickness
Heterogeneity: Direction of vectors and angles between them

Technical Challenge

- **Relations are not labeled in implicit feedback**
  - In knowledge base, relations are labeled
    - ex) place_of_birth, city_of, nationality
  - In user-item graph, relations are not labeled (implicit feedback dataset)
    - Every “Observed” is not the same
      - Some items are more preferred by users

Goal: How to model the relationship (r) between user and item

Possible solution: Introducing new parameter for each user-item pair (?)

- Prone to over-fitting (too many parameters)
- The collaborative information is not explicitly modeled

Proposed Method: Neighborhood approach

- Neighborhood information is the core idea of CF
  - A **user** can be represented by the **items that the user consumed**

$$\boldsymbol{\alpha}_u^{nbr} = \frac{1}{|\mathcal{N}_u^{\mathcal{I}}|} \sum_{k \in \mathcal{N}_u^{\mathcal{I}}} \boldsymbol{\beta}_k$$

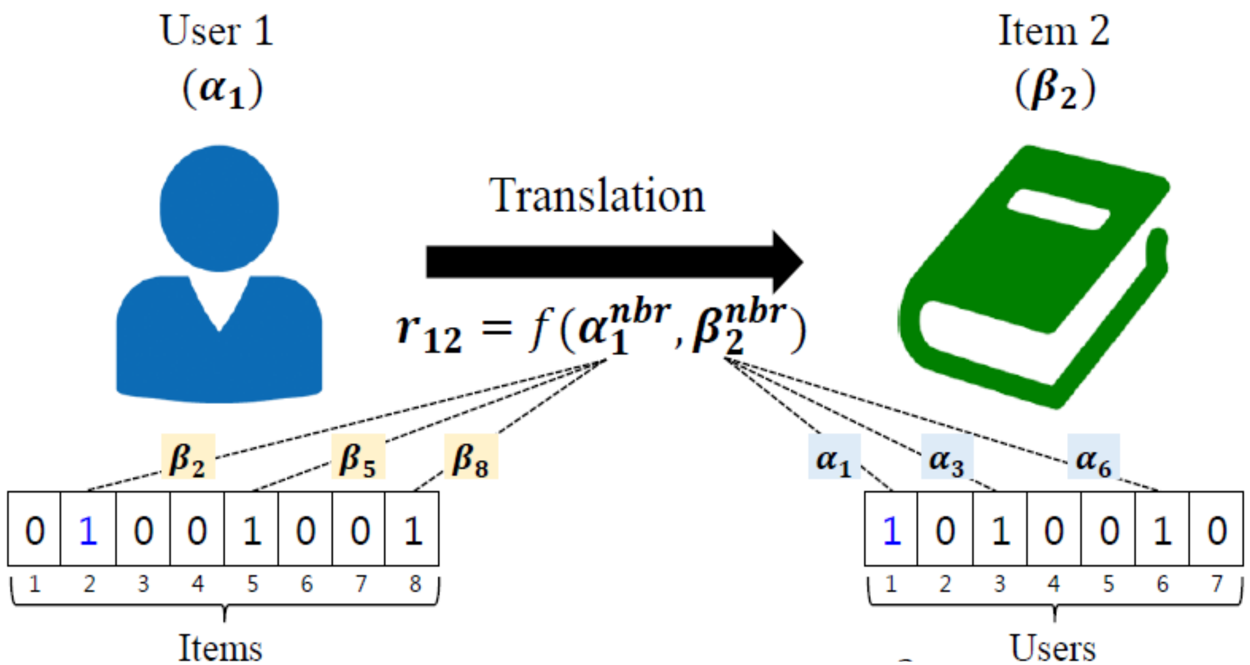
  - An **item** can be represented by the **users that consumed the item**

$$\boldsymbol{\beta}_i^{nbr} = \frac{1}{|\mathcal{N}_i^{\mathcal{U}}|} \sum_{k \in \mathcal{N}_i^{\mathcal{U}}} \boldsymbol{\alpha}_k$$

  - Model the relationship ($\boldsymbol{r}$) between a **user** and an **item** by modeling the interaction between the **[items the user rated]** and **[users that rated the item]**

$$\boldsymbol{r}_{ui} = f(\boldsymbol{\alpha}_u^{\text{nbr}}, \boldsymbol{\beta}_i^{\text{nbr}})$$

Proposed Method: Neighborhood approach



- **Benefit**
  - Explicitly integrate the collaborative information into the model
    - CML does it implicitly by satisfying the triangle inequality
  - <u>**Does not introduce any new parameters**</u>

Proposed Method: Objective Function

- Margin-based pairwise ranking criterion: Hinge loss

$$\mathcal{L}(\Theta) = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{N}_u^{\mathcal{I}}} \sum_{j \notin \mathcal{N}_u^{\mathcal{I}}} [\gamma - s(u,i) + s(u,j)]_+$$

$$s(u,i) = -\|\boldsymbol{\alpha}_u + \boldsymbol{r}_{ui} - \boldsymbol{\beta}_i\|_2^2$$

$$\boldsymbol{r}_{ui} = \boldsymbol{\alpha}_u^{nbr} \odot \boldsymbol{\beta}_i^{nbr}$$

$$\boldsymbol{\alpha}_u^{nbr} = \frac{1}{|\mathcal{N}_u^{\mathcal{I}}|} \sum_{k \in \mathcal{N}_u^{\mathcal{I}}} \boldsymbol{\beta}_k \qquad \boldsymbol{\beta}_i^{nbr} = \frac{1}{|\mathcal{N}_i^{\mathcal{U}}|} \sum_{k \in \mathcal{N}_i^{\mathcal{U}}} \boldsymbol{\alpha}_k$$

- N_u^I: Set of items rated by user u
- N_i^U: Set of users who rated by item i

Regularizer 1 - Neighborhood regularizer

- $reg_{nbr}(\Theta)$: Neighborhood regularizer
  - We implicitly assumed that α_u can be represented by α_u^{nbr}
  - However, if we can explicitly guide α_u to be close to α_u^{nbr}, the neighborhood information will be better reflected into our model

$$reg_{nbr}(\Theta) = \sum_{u \in \mathcal{U}} \left(\boldsymbol{\alpha}_u - \frac{1}{|\mathcal{N}_u^{\mathcal{I}}|} \sum_{k \in \mathcal{N}_u^{\mathcal{I}}} \boldsymbol{\beta}_k \right)^2 + \sum_{i \in \mathcal{I}} \left(\boldsymbol{\beta}_i - \frac{1}{|\mathcal{N}_i^{\mathcal{U}}|} \sum_{k \in \mathcal{N}_i^{\mathcal{U}}} \boldsymbol{\alpha}_k \right)^2$$

Regularizer 2 - Distance regularizer

- $reg_{dist}(\Theta)$: Distance regularizer
  - Currently, item embedding is the nearest neighbor of the translated user embedding
    - Positive item will be pulled to user by pushing the negative item away from the user → **Push loss**
  - However, the relations become more complex as the number of user-item interactions grows
    - Crucial to guarantee that the actual distance between them is small → **Pull loss**

$$reg_{dist}(\Theta) = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{N}_u^{\mathcal{I}}} -s(u,i) = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{N}_u^{\mathcal{I}}} \|\boldsymbol{\alpha}_u + \boldsymbol{r}_{ui} - \boldsymbol{\beta}_i\|_2^2$$

Proposed Method: Optimization

$$\mathcal{J}(\Theta) = (\underbrace{\mathcal{L}(\Theta)}_{\text{Margin-based loss}} + \underbrace{\lambda_{\text{nbr}} \cdot reg_{\text{nbr}}(\Theta) + \lambda_{\text{dist}} \cdot reg_{\text{dist}}(\Theta)}_{\text{Regularizers}})$$

Optimized by stochastic gradient descent (SGD)

Evaluation: Dataset

Dataset	#Users	#Items.	#Inter.	Density	Rat.	#Cat.
Delicious	1,050	1,196	7,698	0.61%	-	-
Tradesy	3,352	5,547	32,710	0.13%	-	-
Ciao	6,760	11,166	146,996	0.19%	1-5	28
Amazon	59,089	17,969	332,236	0.03%	1-5	45
Bookcr	19,571	39,702	605,178	0.08%	1-10	-
Flixster	69,482	25,687	8,000,690	0.45%	0.5-5.0	-
Pinterest	55,187	9,329	1,462,895	0.28%	-	-

To verify the heterogeneity

To verify the intensity
- Considered each observed rating as an implicit feedback record

Baseline Methods

1. **Learning-to-rank baselines**
   - Pointwise methods: eALS [SIGIR 2016], NeuMF [WWW 2017]
   - Pairwise methods: BPR [UAI 2009], AoBPR [WSDM 2014]
2. **Neighborhood-based baselines**
   - FISM [KDD 2013], CDAE [WSDM 2016]
3. **Metric learning-based baselines**
   - CML [WWW 2017]
     - $s(u,i) = -\|\boldsymbol{\alpha}_u - \boldsymbol{\beta}_i\|^2$
   - **Ablation of TransCF**
     - TransCF$^{\text{dot}}$
       - $s(u,i) = (\boldsymbol{\alpha}_u + \boldsymbol{r}_{ui})^T \boldsymbol{\beta}_i$
     - TransCF$^{\text{alt}}$ (without neighborhood information)
       - $s(u,i) = -\|\boldsymbol{\alpha}_u + \boldsymbol{r}_{ui} - \boldsymbol{\beta}_i\|^2$, $\boldsymbol{r}_{ui} = f(\boldsymbol{\alpha}_u, \boldsymbol{\beta}_i)$
     - **TransCF**
       - $s(u,i) = -\|\boldsymbol{\alpha}_u + \boldsymbol{r}_{ui} - \boldsymbol{\beta}_i\|^2$, $\boldsymbol{r}_{ui} = f(\boldsymbol{\alpha}_u^{\boldsymbol{nbr}}, \boldsymbol{\beta}_i^{\boldsymbol{nbr}})$

Performance Comparison

Datasets	Metrics	BPR	FISM	AoBPR	eALS	CDAE	NeuMF	CML	TransCF^{dot}	TransCF^{alt}	TransCF	*Imp.*
Delicious	H@10	0.1981	0.2203	0.2243	0.1992	0.1319	0.1164	0.2470	0.2150	0.2174	**0.2586**	4.70%
	H@20	0.3177	0.3391	0.3602	0.2942	0.2414	0.2171	0.3649	0.3377	0.3084	**0.3786**	3.75%
	N@10	0.1122	0.1124	0.1114	0.1035	0.0674	0.0558	0.1389	0.1101	0.1281	**0.1475**	6.19%
	N@20	0.1418	0.1424	0.1452	0.1271	0.0949	0.0789	0.1678	0.1412	0.1494	**0.1781**	6.14%
Tradesy	H@10	0.2481	0.2676	0.2597	0.2058	0.1652	0.1167	0.3031	0.2846	0.2648	**0.3198**	5.51%
	H@20	0.4174	0.4109	0.4256	0.3314	0.2867	0.2290	0.4413	0.4266	0.3823	**0.4505**	2.08%
	N@10	0.1248	0.1309	0.1300	0.1042	0.0831	0.0538	0.1685	0.1449	0.1466	**0.1767**	4.87%
	N@20	0.1673	0.1670	0.1715	0.1356	0.1136	0.0817	0.2031	0.1806	0.1760	**0.2095**	3.15%
Ciao	H@10	0.1569	0.2100	0.1873	0.1419	0.1770	0.1535	0.2085	0.2011	0.1991	**0.2292**	9.93%
	H@20	0.2811	0.3482	0.3146	0.2570	0.3153	0.2788	0.3337	0.3185	0.3270	**0.3740**	12.08%
	N@10	0.0751	0.1027	0.0891	0.0670	0.0862	0.0741	0.1053	0.1017	0.0989	**0.1167**	10.83%
	N@20	0.1063	0.1374	0.1209	0.0957	0.1208	0.1040	0.1358	0.1311	0.1309	**0.1525**	12.30%
Book-crossing	H@10	0.2425	0.2178	0.2563	0.1655	0.2244	0.2286	0.2885	0.2802	0.2828	**0.3329**	15.39%
	H@20	0.3761	0.3938	0.3916	0.2864	0.3610	0.3747	0.4053	0.3932	0.4069	**0.4744**	17.05%
	N@10	0.1250	0.1002	0.1338	0.0791	0.1164	0.1158	0.1663	0.1618	0.1578	**0.1865**	12.15%
	N@20	0.1585	0.1444	0.1676	0.1093	0.1506	0.1482	0.1956	0.1903	0.1890	**0.2221**	13.55%
Amazon C&A	H@10	0.2489	0.2470	0.2646	0.2161	0.2817	0.1317	0.3011	0.3003	0.3184	**0.3436**	14.11%
	H@20	0.3821	0.3782	0.3946	0.3480	0.4117	0.2390	0.4123	0.4184	0.4509	**0.4658**	12.98%
	N@10	0.1276	0.1247	0.1391	0.1064	0.1613	0.0613	0.1752	0.1648	0.1766	**0.2019**	15.24%
	N@20	0.1610	0.1577	0.1718	0.0739	0.1939	0.0880	0.2031	0.1945	0.2094	**0.2323**	14.38%

- **TransCF** > CML
  - Benefit of the translation vectors that translate each user toward items according to the user's relationships with those items

Performance Comparison

Datasets	Metrics	BPR	FISM	AoBPR	eALS	CDAE	NeuMF	CML	TransCF^{dot}	TransCF^{alt}	TransCF	*Imp.*
Delicious	H@10	0.1981	0.2203	0.2243	0.1992	0.1319	0.1164	0.2470	0.2150	0.2174	**0.2586**	4.70%
	H@20	0.3177	0.3391	0.3602	0.2942	0.2414	0.2171	0.3649	0.3377	0.3084	**0.3786**	3.75%
	N@10	0.1122	0.1124	0.1114	0.1035	0.0674	0.0558	0.1389	0.1101	0.1281	**0.1475**	6.19%
	N@20	0.1418	0.1424	0.1452	0.1271	0.0949	0.0789	0.1678	0.1412	0.1494	**0.1781**	6.14%
Tradesy	H@10	0.2481	0.2676	0.2597	0.2058	0.1652	0.1167	0.3031	0.2846	0.2648	**0.3198**	5.51%
	H@20	0.4174	0.4109	0.4256	0.3314	0.2867	0.2290	0.4413	0.4266	0.3823	**0.4505**	2.08%
	N@10	0.1248	0.1309	0.1300	0.1042	0.0831	0.0538	0.1685	0.1449	0.1466	**0.1767**	4.87%
	N@20	0.1673	0.1670	0.1715	0.1356	0.1136	0.0817	0.2031	0.1806	0.1760	**0.2095**	3.15%
Ciao	H@10	0.1569	0.2100	0.1873	0.1419	0.1770	0.1535	0.2085	0.2011	0.1991	**0.2292**	9.93%
	H@20	0.2811	0.3482	0.3146	0.2570	0.3153	0.2788	0.3337	0.3185	0.3270	**0.3740**	12.08%
	N@10	0.0751	0.1027	0.0891	0.0670	0.0862	0.0741	0.1053	0.1017	0.0989	**0.1167**	10.83%
	N@20	0.1063	0.1374	0.1209	0.0957	0.1208	0.1040	0.1358	0.1311	0.1309	**0.1525**	12.30%
Book-crossing	H@10	0.2425	0.2178	0.2563	0.1655	0.2244	0.2286	0.2885	0.2802	0.2828	**0.3329**	15.39%
	H@20	0.3761	0.3938	0.3916	0.2864	0.3610	0.3747	0.4053	0.3932	0.4069	**0.4744**	17.05%
	N@10	0.1250	0.1002	0.1338	0.0791	0.1164	0.1158	0.1663	0.1618	0.1578	**0.1865**	12.15%
	N@20	0.1585	0.1444	0.1676	0.1093	0.1506	0.1482	0.1956	0.1903	0.1890	**0.2221**	13.55%
Amazon C&A	H@10	0.2489	0.2470	0.2646	0.2161	0.2817	0.1317	0.3011	0.3003	0.3184	**0.3436**	14.11%
	H@20	0.3821	0.3782	0.3946	0.3480	0.4117	0.2390	0.4123	0.4184	0.4509	**0.4658**	12.98%
	N@10	0.1276	0.1247	0.1391	0.1064	0.1613	0.0613	0.1752	0.1648	0.1766	**0.2019**	15.24%
	N@20	0.1610	0.1577	0.1718	0.0739	0.1939	0.0880	0.2031	0.1945	0.2094	**0.2323**	14.38%

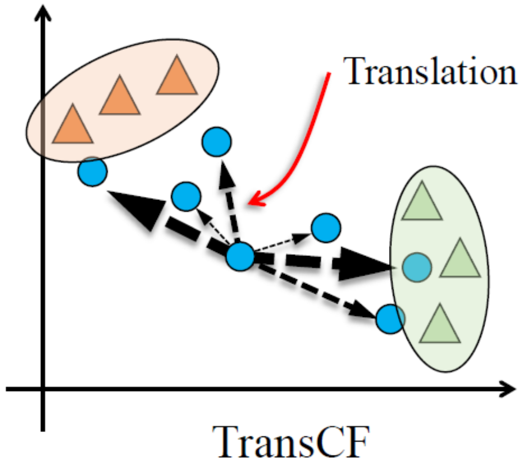
- **CML** > TransCF^{alt}
  - Translation vectors should be carefully designed, otherwise the performance will rather deteriorate

Performance Comparison

Datasets	Metrics	BPR	FISM	AoBPR	eALS	CDAE	NeuMF	CML	TransCF^{do}	TransCF^{alt}	TransCF	*Imp.*
Delicious	H@10	0.1981	0.2203	0.2243	0.1992	0.1319	0.1164	0.2470	0.2150	0.2174	**0.2586**	4.70%
	H@20	0.3177	0.3391	0.3602	0.2942	0.2414	0.2171	0.3649	0.3377	0.3084	**0.3786**	3.75%
	N@10	0.1122	0.1124	0.1114	0.1035	0.0674	0.0558	0.1389	0.1101	0.1281	**0.1475**	6.19%
	N@20	0.1418	0.1424	0.1452	0.1271	0.0949	0.0789	0.1678	0.1412	0.1494	**0.1781**	6.14%
Tradesy	H@10	0.2481	0.2676	0.2597	0.2058	0.1652	0.1167	0.3031	0.2846	0.2648	**0.3198**	5.51%
	H@20	0.4174	0.4109	0.4256	0.3314	0.2867	0.2290	0.4413	0.4266	0.3823	**0.4505**	2.08%
	N@10	0.1248	0.1309	0.1300	0.1042	0.0831	0.0538	0.1685	0.1449	0.1466	**0.1767**	4.87%
	N@20	0.1673	0.1670	0.1715	0.1356	0.1136	0.0817	0.2031	0.1806	0.1760	**0.2095**	3.15%
Ciao	H@10	0.1569	0.2100	0.1873	0.1419	0.1770	0.1535	0.2085	0.2011	0.1991	**0.2292**	9.93%
	H@20	0.2811	0.3482	0.3146	0.2570	0.3153	0.2788	0.3337	0.3185	0.3270	**0.3740**	12.08%
	N@10	0.0751	0.1027	0.0891	0.0670	0.0862	0.0741	0.1053	0.1017	0.0989	**0.1167**	10.83%
	N@20	0.1063	0.1374	0.1209	0.0957	0.1208	0.1040	0.1358	0.1311	0.1309	**0.1525**	12.30%
Book-crossing	H@10	0.2425	0.2178	0.2563	0.1655	0.2244	0.2286	0.2885	0.2802	0.2828	**0.3329**	15.39%
	H@20	0.3761	0.3938	0.3916	0.2864	0.3610	0.3747	0.4053	0.3932	0.4069	**0.4744**	17.05%
	N@10	0.1250	0.1002	0.1338	0.0791	0.1164	0.1158	0.1663	0.1618	0.1578	**0.1865**	12.15%
	N@20	0.1585	0.1444	0.1676	0.1093	0.1506	0.1482	0.1956	0.1903	0.1890	**0.2221**	13.55%
Amazon C&A	H@10	0.2489	0.2470	0.2646	0.2161	0.2817	0.1317	0.3011	0.3003	0.3184	**0.3436**	14.11%
	H@20	0.3821	0.3782	0.3946	0.3480	0.4117	0.2390	0.4123	0.4184	0.4509	**0.4658**	12.98%
	N@10	0.1276	0.1247	0.1391	0.1064	0.1613	0.0613	0.1752	0.1648	0.1766	**0.2019**	15.24%
	N@20	0.1610	0.1577	0.1718	0.0739	0.1939	0.0880	0.2031	0.1945	0.2094	**0.2323**	14.38%

- **TransCF** > TransCF^{alt}
  - Incorporating the neighborhood information is crucial in collaborative filtering

Translation in action



We want to show…

$$\|\boldsymbol{\alpha}_u - \boldsymbol{\beta}_i\|_2^2 > \|\boldsymbol{\alpha}_u + \boldsymbol{r}_{ui} - \boldsymbol{\beta}_i\|_2^2$$

Dataset	*Obs.*	*Unobs.*	Dataset	*Obs.*	*Unobs.*
Delicious	64.63%	43.75%	Amazon	75.57%	31.96%
Tradesy	56.02%	43.01%	Pinterest	36.25%	33.08%
Ciao	54.63%	38.42%	Flixster	22.24%	2.88%
Bookcr.	55.42%	35.57%			

Each translated user is placed closer to the observed (positive) items than to the unobserved (negative) items.

Intensity is encoded in Translation vectors

- **Assumption**: Rating information is a proxy for the intensity of user–item relationships
- Task: Rating prediction with translation vector

$$\boldsymbol{r}_{ui}^{CML} = (\underbrace{\boldsymbol{\alpha}_u - \boldsymbol{\beta}_i}_{\textbf{Learned by CML}})$$

$$\boldsymbol{r}_{ui}^{TransCF^{emb}} = (\underbrace{\boldsymbol{\alpha}_u - \boldsymbol{\beta}_i}_{\textbf{Learned by TransCF}})$$

Acc.(%)	Ciao		Amazon		BookCr.[3]		Flixster	
	Rand	RF	*Rand*	RF	*Rand*	RF	*Rand*	RF
CML	|	50.3	|	50.1	|	39.1	|	20.5
TransCF^{emb}	19.9	50.3	20.1	50.3	13.8	40.1	10.0	20.5
TransCF	|	**53.0**	|	**50.8**	|	**43.7**	|	**23.4**
vs. CML	-	5.3%	-	1.5%	-	11.7%	-	14.2%

Rating prediction accuracy: **TransCF** > CML, TransCF^{emb}

Intensity of user–item relationships is best encoded in the translation vectors learned by TransCF

Intensity is encoded in Translation vectors

- High rating → High intensity → users are translated closer
- Expectation: more observed interactions to satisfy $\|\boldsymbol{\alpha}_u - \boldsymbol{\beta}_i\|_2^2 > \|\boldsymbol{\alpha}_u + \boldsymbol{r}_{ui} - \boldsymbol{\beta}_i\|_2^2$ in higher rating groups.

	Rating						
BookCr.	1-4	5	6	7	8	9	10
Acc.	55.3%	52.7%	55.2%	56.1%	57.2%	58.4%	58.8%
Portion	3.8%	10.3%	7.9%	17.0%	24.5%	17.3%	19.2%
Flixster	0.5-2.5	3.0	3.5	4.0	4.5	5.0	
Acc.	19.6%	19.9%	19.9%	22.2%	25.7%	27.2%	
Portion	17.3%	17.0%	16.8%	19.6%	10.1%	19.2%	
Ciao	1	2	3	4	5		
Acc.	61.5%	51.4%	55.4%	52.2%	55.4%		
Portion	4.8%	5.1%	11.4%	29.0%	49.7%		
Amazon	1	2	3	4	5		
Acc.	76.7%	76.3%	75.7%	75.2%	75.4%		
Portion	7.0%	5.7%	10.7%	20.1%	56.5%		

High rating → More interactions satisfy $\|\boldsymbol{\alpha}_u - \boldsymbol{\beta}_i\|_2^2 > \|\boldsymbol{\alpha}_u + \boldsymbol{r}_{ui} - \boldsymbol{\beta}_i\|_2^2$

Does not agree with our expectation

1) Range of ratings is small
2) Majority belongs to 4,5

→ **Hard to infer users' fine-grained preferences**

Heterogeneity is encoded in Translation vectors

- **Assumption**: Item category = Users' taste
- Task: Item category classification using $\boldsymbol{r}_{ui}$ and $\boldsymbol{\beta}_i$

Dataset	Method	Rand.	Random Forest
Ciao	CML	10.01%	67.86±0.47%
	TransCF$^{\mathrm{emb}}$		67.27±0.28%
	TransCF		**80.97**±0.73%
Amazon C&A	CML	10.40%	54.26±0.74%
	TransCF$^{\mathrm{emb}}$		54.85±0.51%
	TransCF		**81.24**±0.46%

(a) Classification on translation vectors ($\boldsymbol{r}_{ui}$).

Dataset	Method	Rand.	Random Forest
Ciao	CML	10.92%	80.41±1.59%
	TransCF		81.61±1.54%
Amazon C&A	CML	9.40%	47.94±3.34%
	TransCF		47.90±2.54%

(b) Classification on item embeddings ($\boldsymbol{\beta}_i$).

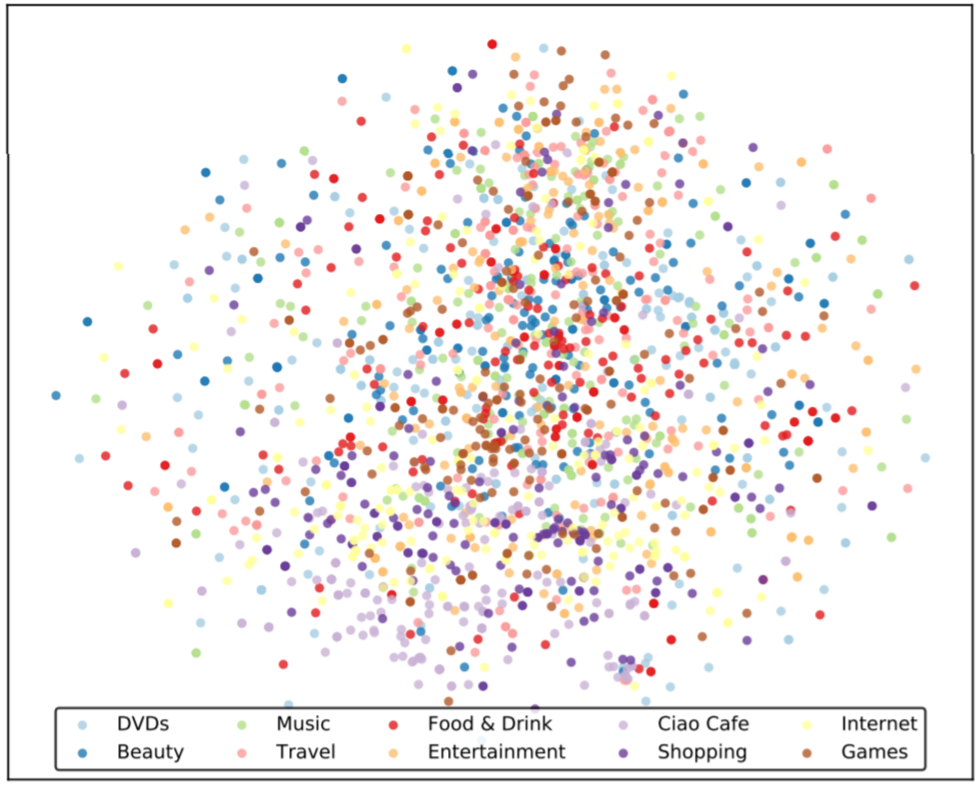
TransCF > CML

- Translation vectors ($\boldsymbol{r}_{ui}$) encode the category information → Heterogeneity of the user–item relationships

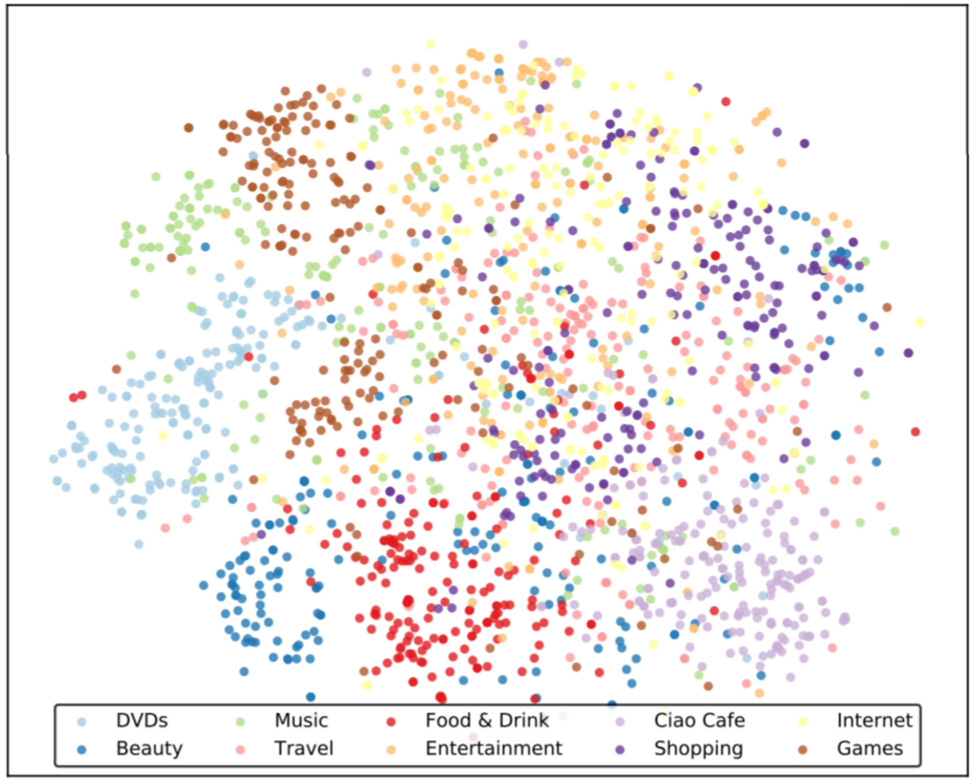
TransCF ≈ CML

- Superior performance of TransCF is not derived from the high-quality embedding vectors

Heterogeneity is encoded in Translation vectors



(a) Visualization of $\boldsymbol{r}_{ui}^{CML}$



(b) Visualization of $\boldsymbol{r}_{ui}^{\text{TransCF}}$

Translation vectors **capture item category information**
(without given any category information)