

Объяснение рекомендаций: обзор подходов

Объяснение рекомендаций: обзор подходов

- А зачем что-то кому-то объяснять?..
-

Причины:

1. Тестирование (debug). Прозрачность и устойчивость результатов
2. Повышение доверия и понимания пользователя
3. Юридическая
4. Этическая

Типы подходов объяснения рекомендаций:

“This product you are looking at is similar to these other products you liked before” (Schafer et al., 1999) - > item-based CF.

“Explainable recommendations” (Zhang et al., 2014)

1. Источник информации или дисплей для объяснения пользователю (HCI)
2. Модель для объяснения рекомендаций (ML)



1. Model-intrinsic (обучение объяснения)
2. Model-agnostic (post-hoc объяснения)

+ **global** vs. **local** уровни интерпретируемости рекомендаций

Zhang Y, Chen X. (2020) **Explainable recommendation: A survey and new perspectives.**
in Information Retrieval: Vol. 14, No. 1, pp 1–101.

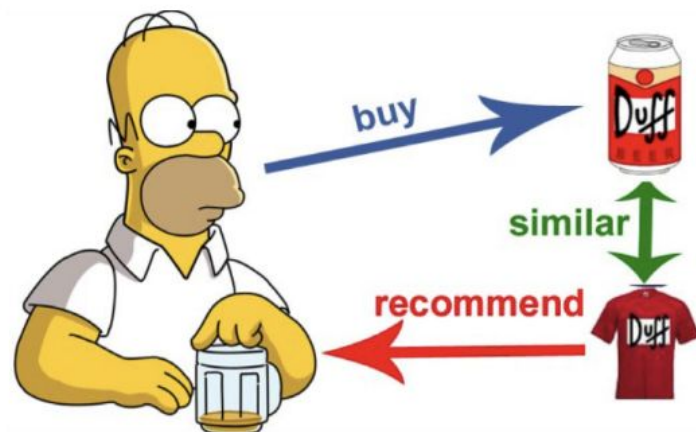
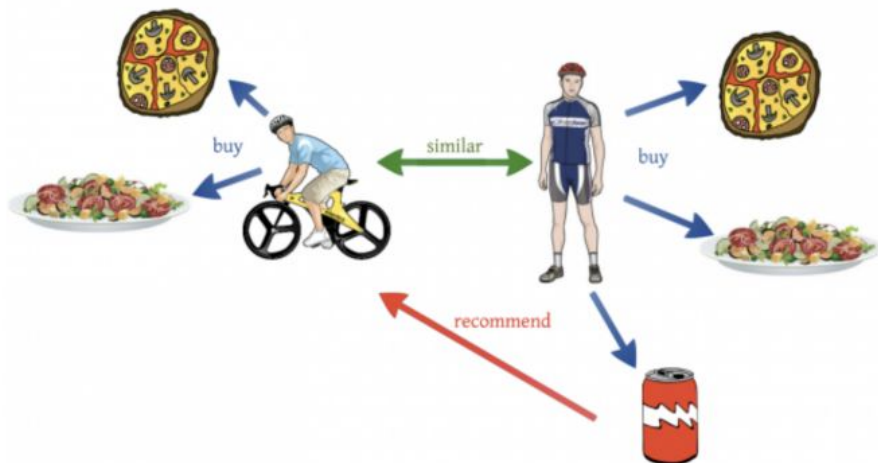
Information/ style of the explanations	Methods for Explainable Recommendation							
	Neighbor- based	Matrix factorization	Topic modeling	Graph- based	Deep learning	Knowledge- based	Rule mining	Post- hoc
Relevant user or item	Herlocker <i>et al.</i> , 2000	Abdollahi and Nasraoui, 2017		Heckel <i>et al.</i> , 2017	Chen <i>et al.</i> , 2018c	Catherine <i>et al.</i> , 2017	Peake and Wang 2018	Cheng <i>et al.</i> , 2019a
User or item features	Vig <i>et al.</i> , 2009	Zhang <i>et al.</i> , 2014a	McAuley and Leskovec, 2013	He <i>et al.</i> , 2015	Seo <i>et al.</i> , 2017	Huang <i>et al.</i> , 2018	Davidson <i>et al.</i> , 2010	McInerney <i>et al.</i> , 2018
Textual sentence explanation		Zhang <i>et al.</i> , 2014a			Li <i>et al.</i> , 2017	Ai <i>et al.</i> , 2018	Balog <i>et al.</i> , 2019	Wang <i>et al.</i> , 2018d
Visual explanation					Chen <i>et al.</i> , 2019b			
Social explanation	Sharma and Cosley, 2013		Ren <i>et al.</i> , 2017	Park <i>et al.</i> , 2018				
Word cluster		Zhang, 2015	Wu and Ester 2015					

Подход	Пример объяснения	Типы объяснения
MF, CF, user/item similarity, rule based	<p>Покупателю советуют товар, потому что:</p> <ol style="list-style-type: none"> 1. он купил такие-то товары (похожие, комплиментарные, встречаемые). Item-based. 2. похожие по истории покупок на него пользователи покупают эти товары / X% пользователей со схожими интересами оценили рекомендуемый товар. User-based. 3. конкретный признак товара важен для пользователя (EFM). 	<p>Есть список топ-k товаров, повлиявших на рекомендацию. Список соседей и их оценки. etc.</p>
Content-based models, Knowledge-based graphs	<p>Пользователю рекомендуют Титаник, потому что он посмотрел Forrest Gump, т.е. фильм с таким же свойством - например, жанром драма.</p>	<p>Объяснения через общие признаки по объектам, агрегации на графе по k близким соседям</p>

Подход	Пример объяснения	Типы объяснения
NN	<p>Помимо примеров из предыдущих подходов:</p> <ol style="list-style-type: none"> 1.Пользователю предлагается посетить ресторан, потому что в нем хорошая еда, вкусные стейки (пары (opinion-aspect) из отзывов ресторанов). 2. На основе сходства по общим предпочтениям похожих пользователей, их признаков и поведения (эмбединги) 3. Купите чехол для телефона, так как недавно вы купили Iphone 13. 	<p>Через контексты и паттерны, которые нельзя было уловить другими моделями (повторяемость, последовательность, частота, время).</p>

Коллаборативная фильтрация

- Похожие на вас пользователи выбирают (покупают/смотрят...) это
- Похожие товары на те, которые вы выбирали (покупали/смотрели) до этого



Collaborative Filtering for Implicit Feedback Datasets

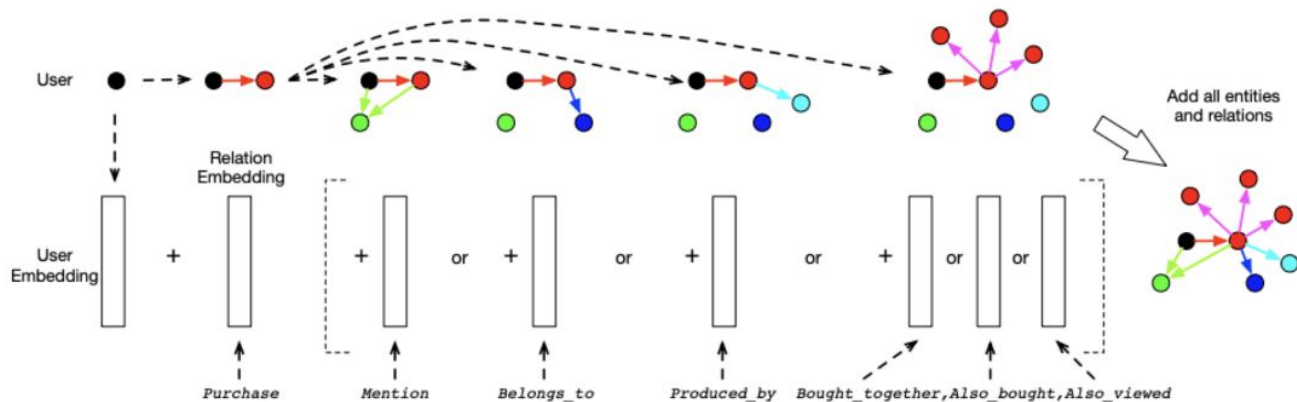
Hu Y., Koren Y., Volinsky C. 2008

- **Вид объяснения:** Покупателю советуют товар, потому что он купил такие-то товары (доступен список TOP-K товаров, повлиявших на рекомендацию).
- **Подход:** Матричные разложения

Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

Qingyao A. ,Vahid A., Xu C. et al. 2018

- **Вид объяснения:** Покупателю советуют iPad, потому что он уже покупал товар такого бренда (Apple)
- **Подход:** knowledge graph



Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

Сущности (entities):

- Пользователи (users)
- Объекты (items)
- Слова (описания, отзывы)
- Бренды
- Категории

Бинарные отношения:

- **Покупка** (пользователь– объект)
- **Упоминание** (пользователь/объект– слово)
- **Принадлежность** (объект – категория)
- **Производство** (объект – бренд)
- **Совстречаемость в одной заказе** (объект – объект)
- **Совстречаемость у одного пользователя** (объект – объект)
- **Последовательно просмотренные** (объект – объект)

Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

$$S = \{(e_h, e_t, r)\}$$

$$\mathbf{e}_t = \textit{trans}(e_h, r) = \mathbf{e}_h + \mathbf{r}$$

Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

$$S = \{(e_h, e_t, r)\}$$

$$\mathbf{e}_t = \text{trans}(e_h, r) = \mathbf{e}_h + \mathbf{r}$$

$$\text{trans}(e_h, r) \approx \mathbf{e}_t \quad (e_h, e_t, r) \in S$$

$$\text{trans}(e_h, r) \neq \mathbf{e}'_t \quad (e_h, e'_t, r) \notin S$$

Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

$$\begin{aligned} S &= \{(e_h, e_t, r)\} & trans(e_h, r) &\approx \mathbf{e}_t & (e_h, e_t, r) &\in S \\ \mathbf{e}_t &= trans(e_h, r) = \mathbf{e}_h + \mathbf{r} & trans(e_h, r) &\neq \mathbf{e}'_t & (e_h, e'_t, r) &\notin S \end{aligned}$$

$$P(e_t | trans(e_h, r)) = \frac{\exp(\mathbf{e}_t \cdot trans(e_h, r))}{\sum_{e'_t \in E_t} \exp(\mathbf{e}'_t \cdot trans(e_h, r))}$$

Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

$$S = \{(e_h, e_t, r)\} \quad \begin{array}{ll} trans(e_h, r) \approx \mathbf{e}_t & (e_h, e_t, r) \in S \\ \mathbf{e}_t = trans(e_h, r) = \mathbf{e}_h + \mathbf{r} & trans(e_h, r) \neq \mathbf{e}'_t \quad (e_h, e'_t, r) \notin S \end{array}$$

$$P(e_t | trans(e_h, r)) = \frac{\exp(\mathbf{e}_t \cdot trans(e_h, r))}{\sum_{e'_t \in E_t} \exp(\mathbf{e}'_t \cdot trans(e_h, r))}$$

$$\mathcal{L}(S) = \log \prod_{(e_h, e_t, r) \in S} P(e_t | trans(e_h, r))$$

$$\mathcal{L}(S) = \sum_{(e_h, e_t, r) \in S} \log \sigma(\mathbf{e}_t \cdot trans(e_h, r)) + k \cdot \mathbb{E}_{e'_t \sim P_t} [\log \sigma(-\mathbf{e}'_t \cdot trans(e_h, r))]$$

Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

Algorithm 1: Recommendation Explanation Extraction

Input: $S = \{(E_h, E_t, r)\}, e_u, e_i$, maximum depth z

Output: e_x, R_α, R_β

Procedure Main()

1 $V_u, P_u, \mathcal{R}_u = \text{BFS}(S, e_u, z).$

2 $V_i, P_i, \mathcal{R}_i = \text{BFS}(S, e_i, z).$

3 $P \leftarrow \{\}.$

4 **for** $e \in V_u \cap V_i$ **do**

5 $P[e] = P_u(e) \cdot P_i(e).$

end

6 Pick up $e_x \in V_u \cap V_i$ with the largest $P[e].$

7 $R_\alpha = \mathcal{R}_u[e_x], R_\beta = \mathcal{R}_i[e_x].$

8 **return** e_x, R_α, R_β

Function BFS(S, e, z)

9 $V_e \leftarrow$ all entities in the entity set E_t within z hops from $e.$

10 $P_e \leftarrow$ the probability of each entity in V_e computed by Eq (10).

11 $\mathcal{R}_e \leftarrow$ the paths from e to the space of each entity in $V_e.$

12 **return** $V_e, P_e, \mathcal{R}_e;$

$$e_u + \sum_{\alpha=1}^m r_\alpha = e_i + \sum_{\beta=1}^n r_\beta$$

Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

Algorithm 1: Recommendation Explanation Extraction

Input: $S = \{(E_h, E_t, r)\}, e_u, e_i$, maximum depth z

Output: e_x, R_α, R_β

Procedure Main()

```

1   $V_u, P_u, \mathcal{R}_u = \text{BFS}(S, e_u, z).$ 
2   $V_i, P_i, \mathcal{R}_i = \text{BFS}(S, e_i, z).$ 
3   $P \leftarrow \{\}$ .
4  for  $e \in V_u \cap V_i$  do
5     $P[e] = P_u(e) \cdot P_i(e).$ 
6  end
7  Pick up  $e_x \in V_u \cap V_i$  with the largest  $P[e]$ .
8   $R_\alpha = \mathcal{R}_u[e_x], R_\beta = \mathcal{R}_i[e_x].$ 
9  return  $e_x, R_\alpha, R_\beta$ 

```

Function BFS(S, e, z)

```

9   $V_e \leftarrow$  all entities in the entity set  $E_t$  within  $z$  hops from  $e$ .
10  $P_e \leftarrow$  the probability of each entity in  $V_e$  computed by Eq (10).
11  $\mathcal{R}_e \leftarrow$  the paths from  $e$  to the space of each entity in  $V_e$ .
12 return  $V_e, P_e, \mathcal{R}_e$ ;

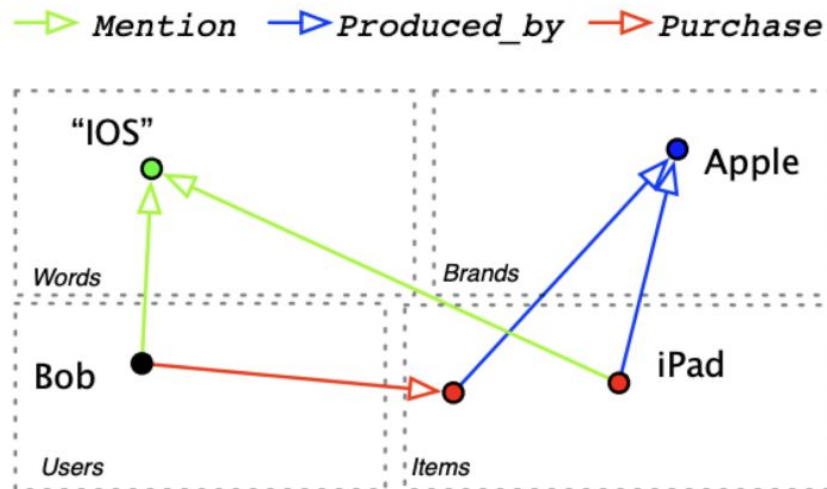
```

$$P(e_x | \text{trans}(e_u, R_\alpha)) = \frac{\exp(e_x \cdot \text{trans}(e_u, R_\alpha))}{\sum_{e' \in E_t^{rm}} \exp(e' \cdot \text{trans}(e_u, R_\alpha))}$$

$$P(e_x | e_u, R_\alpha, e_i, R_\beta) = P(e_x | \text{trans}(e_u, R_\alpha)) P(e_x | \text{trans}(e_i, R_\beta))$$

Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

Relations	CDs and Vinyl				Clothing			
Measures(%)	NDCG	Recall	HT	Prec	NDCG	Recall	HT	Prec
<i>Purchase only</i>	1.725	2.319	7.052	0.818	0.974	1.665	2.651	0.254
<i>+Also_view</i>	1.722	2.356	6.967	0.817	1.800	3.130	4.672	0.448
<i>+Also_bought</i>	3.641	5.285	12.332	1.458	1.352	2.419	3.580	0.343
<i>+Bought_together</i>	1.962	2.712	7.473	0.861	0.694	1.284	2.026	0.189
<i>+Produced_by</i>	1.719	2.318	6.842	0.792	0.579	1.044	1.630	0.155
<i>+Belongs_to</i>	2.799	4.028	10.297	1.200	1.453	2.570	3.961	0.376
<i>+Mention</i>	3.822	5.185	12.828	1.628	1.019	1.754	2.780	0.265
<i>+all (our model)</i>	5.563	7.949	17.556	2.192	3.091	5.466	7.972	0.763



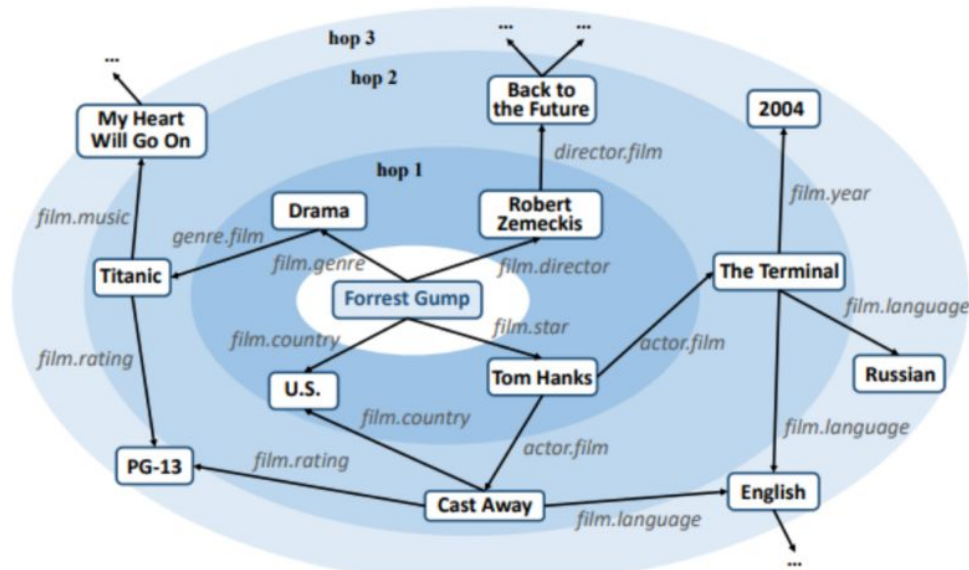
Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation

Dataset	CDs and Vinyl				Clothing				Cell Phones				Beauty			
Measures (%)	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.
BPR	2.009	2.679	8.554	1.085	0.601	1.046	1.767	0.185	1.998	3.258	5.273	0.595	2.753	4.241	8.241	1.143
BPR-HFT	2.661	3.570	9.926	1.268	1.067	1.819	2.872	0.297	3.151	5.307	8.125	0.860	2.934	4.459	8.268	1.132
VBPR	0.631	0.845	2.930	0.328	0.560	0.968	1.557	0.166	1.797	3.489	5.002	0.507	1.901	2.786	5.961	0.902
TransRec	3.372	5.283	11.956	1.837	1.245	2.078	3.116	0.312	3.361	6.279	8.725	0.962	3.218	4.853	9.867	1.285
DeepCoNN	4.218	6.001	13.857	1.681	1.310	2.332	3.286	0.229	3.636	6.353	9.913	0.999	3.359	5.429	9.807	1.200
CKE	4.620	6.483	14.541	1.779	1.502	2.509	4.275	0.388	3.995	7.005	10.809	1.070	3.717	5.938	11.043	1.371
JRL	5.378 *	7.545 *	16.774 *	2.085 *	1.735 *	2.989 *	4.634 *	0.442 *	4.364 *	7.510 *	10.940 *	1.096 *	4.396 *	6.949 *	12.776 *	1.546 *
Our model	5.563	7.949	17.556	2.192	3.091	5.466	7.972	0.763	5.370	9.498	13.455	1.325	6.399	10.411	17.498	1.986
Improvement	3.44	5.35	4.66	5.13	78.16	82.87	72.03	72.62	23.05	26.47	22.99	20.89	45.56	49.82	36.96	28.46

RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems

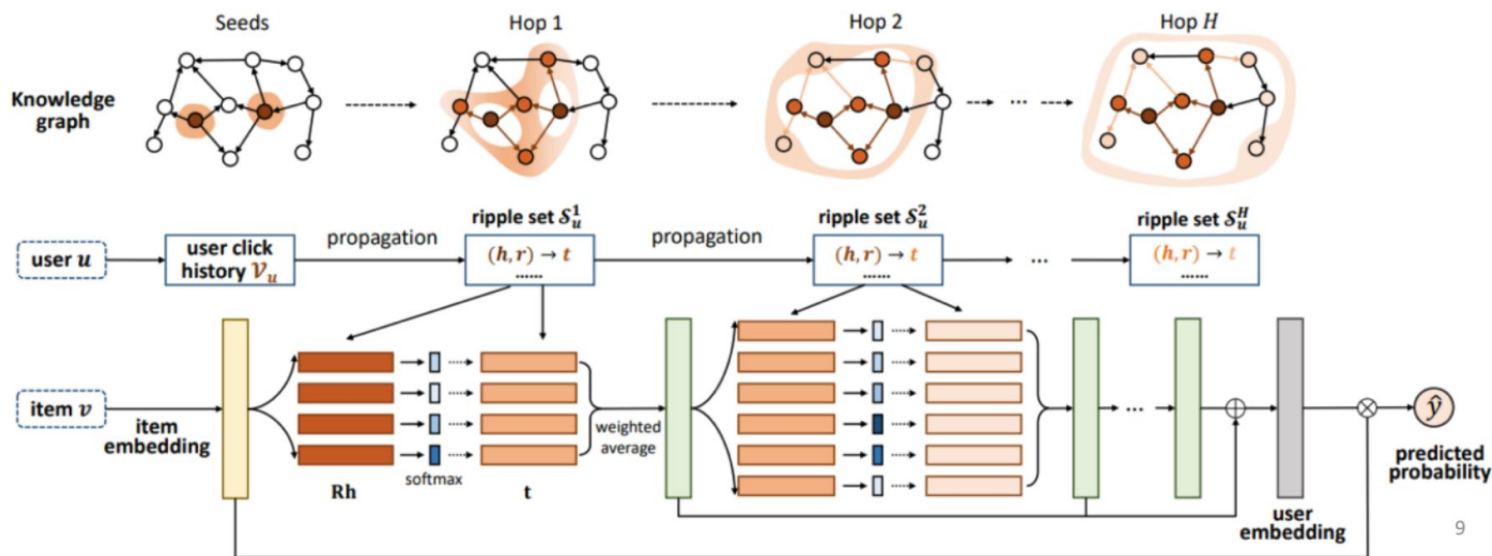
Hongwei W. , Fuzheng Z. , Jialin W. et al. Microsoft Research. 2018

- **Вид объяснения:** Пользователь посмотрел драму Forrest Gump. Модель советует посмотреть фильм того же жанра - Титаник.
- **Подход:** knowledge graph



RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems

$$p_i = \text{softmax} \left(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i \right) = \frac{\exp \left(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i \right)}{\sum_{(h,r,t) \in \mathcal{S}_u^1} \exp \left(\mathbf{v}^T \mathbf{R} \mathbf{h} \right)}$$



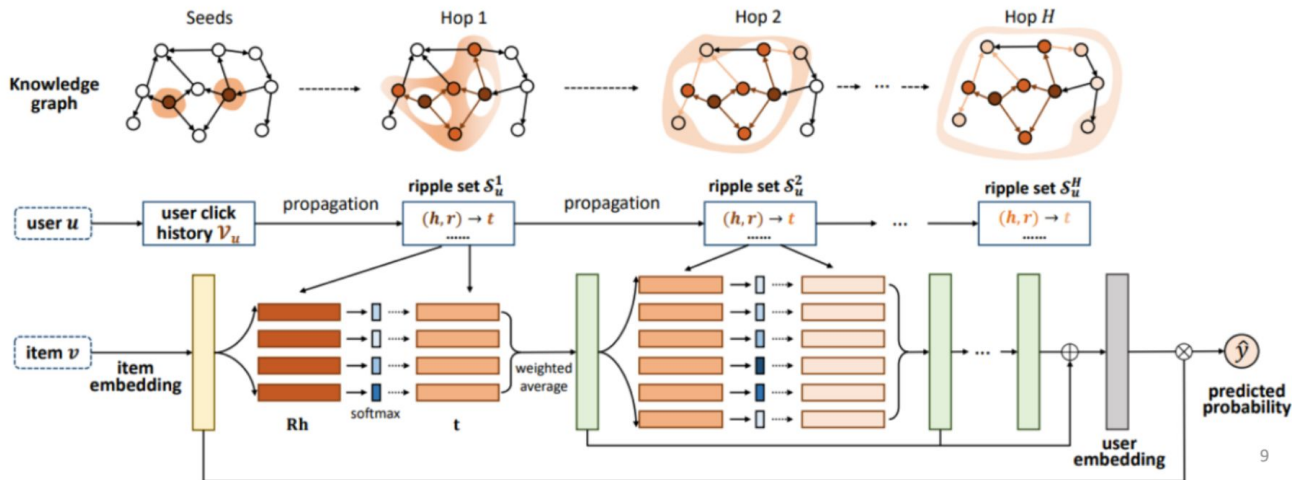
RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems

$$p_i = \text{softmax} \left(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i \right) = \frac{\exp \left(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i \right)}{\sum_{(h, r, t) \in \mathcal{S}_u^1} \exp \left(\mathbf{v}^T \mathbf{R} \mathbf{h} \right)}$$

$$\mathbf{o}_u^1 = \sum_{(h_i, r_i, t_i) \in \mathcal{S}_u^1} p_i \mathbf{t}_i$$

$$\mathbf{u} = \mathbf{o}_u^1 + \mathbf{o}_u^2 + \dots + \mathbf{o}_u^H$$

$$\hat{y}_{uv} = \sigma(\mathbf{u}^T \mathbf{v})$$



RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems

Algorithm 1 Learning algorithm for RippleNet

Input: Interaction matrix \mathbf{Y} , knowledge graph \mathcal{G}

Output: Prediction function $\mathcal{F}(u, v|\Theta)$

- 1: Initialize all parameters
- 2: Calculate ripple sets $\{\mathcal{S}_u^k\}_{k=1}^H$ for each user u ;
- 3: **for** number of training iteration **do**
- 4: Sample minibatch of positive and negative interactions from \mathbf{Y} ;
- 5: Sample minibatch of true and false triples from \mathcal{G} ;
- 6: Calculate gradients $\partial\mathcal{L}/\partial\mathbf{V}$, $\partial\mathcal{L}/\partial\mathbf{E}$, $\{\partial\mathcal{L}/\partial\mathbf{R}\}_{r\in\mathcal{R}}$, and $\{\partial\mathcal{L}/\partial\alpha_i\}_{i=1}^H$ on the minibatch by back-propagation according to Eq. (4)-(13);
- 7: Update \mathbf{V} , \mathbf{E} , $\{\mathbf{R}\}_{r\in\mathcal{R}}$, and $\{\alpha_i\}_{i=1}^H$ by gradient descent with learning rate η ;
- 8: **end for**
- 9: **return** $\mathcal{F}(u, v|\Theta)$

RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems

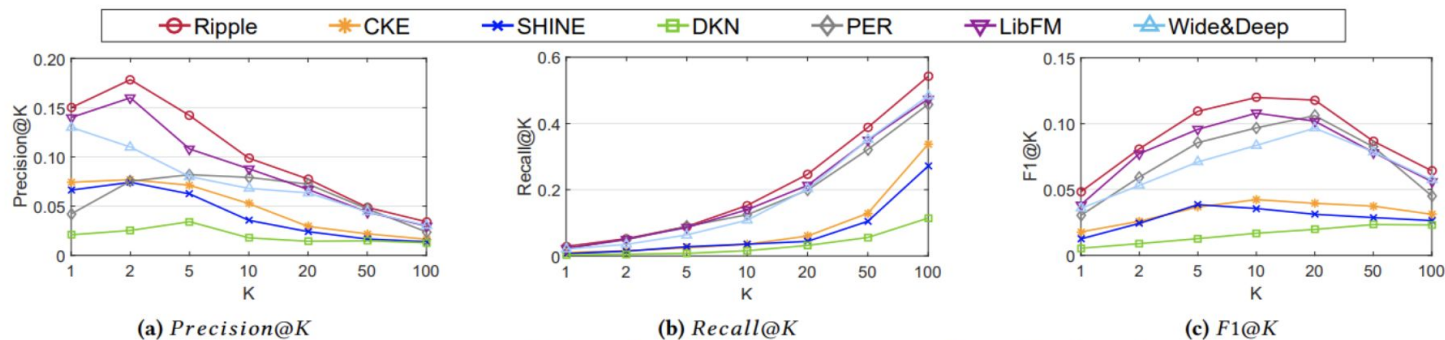


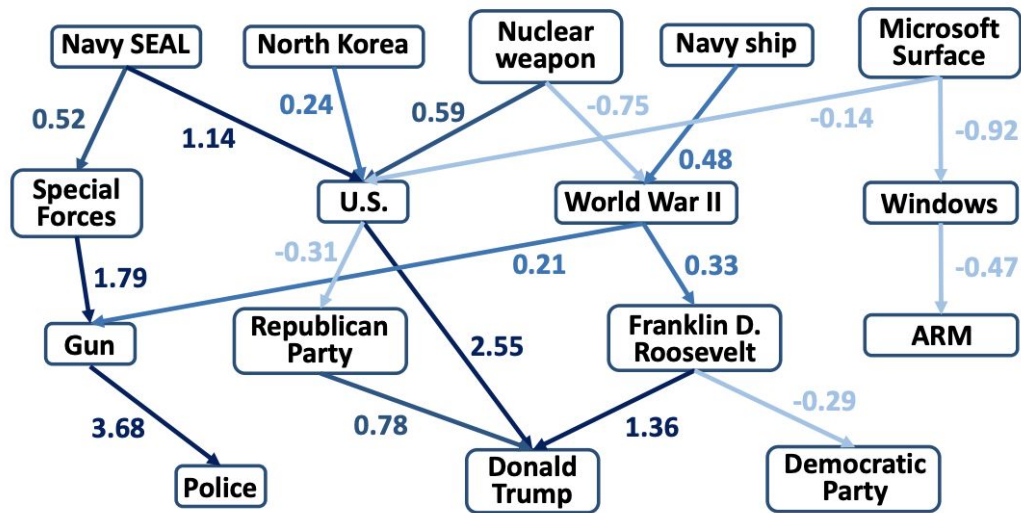
Figure 5: Precision@K, Recall@K, and F1@K in top-K recommendation for MovieLens-1M.

Table 3: The results of AUC and Accuracy in CTR prediction.

Model	MovieLens-1M		Book-Crossing		Bing-News	
	AUC	ACC	AUC	ACC	AUC	ACC
RippleNet*	0.921	0.844	0.729	0.662	0.678	0.632
CKE	0.796	0.739	0.674	0.635	0.560	0.517
SHINE	0.778	0.732	0.668	0.631	0.554	0.537
DKN	0.655	0.589	0.621	0.598	0.661	0.604
PER	0.712	0.667	0.623	0.588	-	-
LibFM	0.892	0.812	0.685	0.639	0.644	0.588
Wide&Deep	0.903	0.822	0.711	0.623	0.654	0.595

* Statistically significant improvements by unpaired two-sample t -test with $p = 0.1$.

RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems



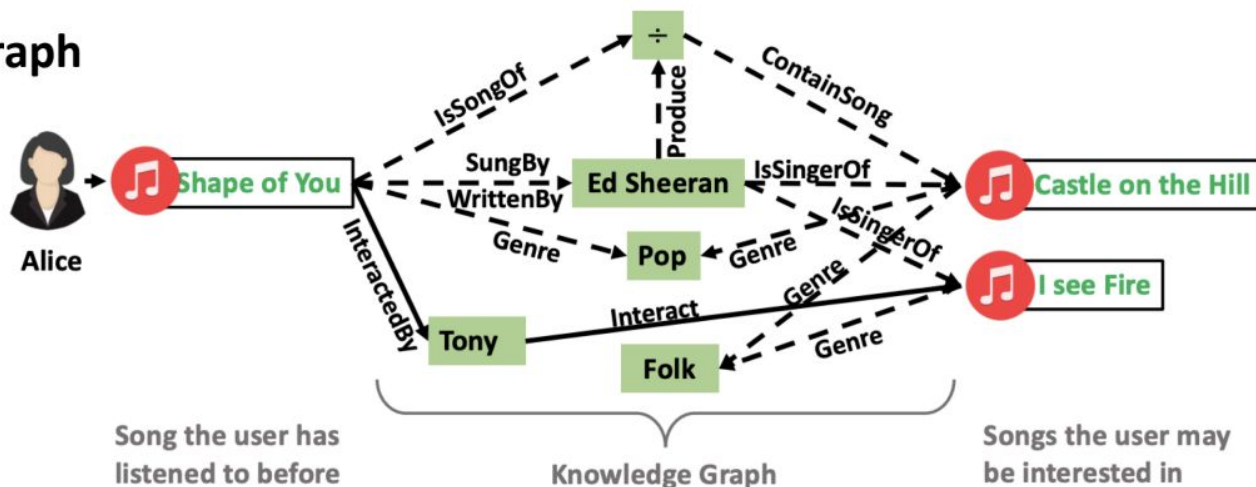
Candidate news: **Trump** Announces Gunman Dead, Credits 'Heroic Actions' of **Police**

Figure 8: Visualization of relevance probabilities for a randomly sampled user w.r.t. a piece of candidate news with label 1. Links with value lower than -1.0 are omitted.

Explainable Reasoning over Knowledge Graphs for Recommendation

Xiang W., Dingxian W. , Canran X. al. eBay. 2018

- **Вид объяснения:** пользователю рекомендуют прослушать "Castle on the Hill", потому что он слушал другую песню ("Shape of you") этого исполнителя (Ed Sheeran).
- **Подход:** knowledge graph



Explainable Reasoning over Knowledge Graphs for Recommendation

$$\hat{y}_{ui} = \sigma\left(\frac{1}{K} \sum_{k=1}^K s_k\right)$$

$$s(\tau|p_k) = \mathbf{W}_2' \text{ReLU}(\mathbf{W}_1^\top \mathbf{p}_k)$$

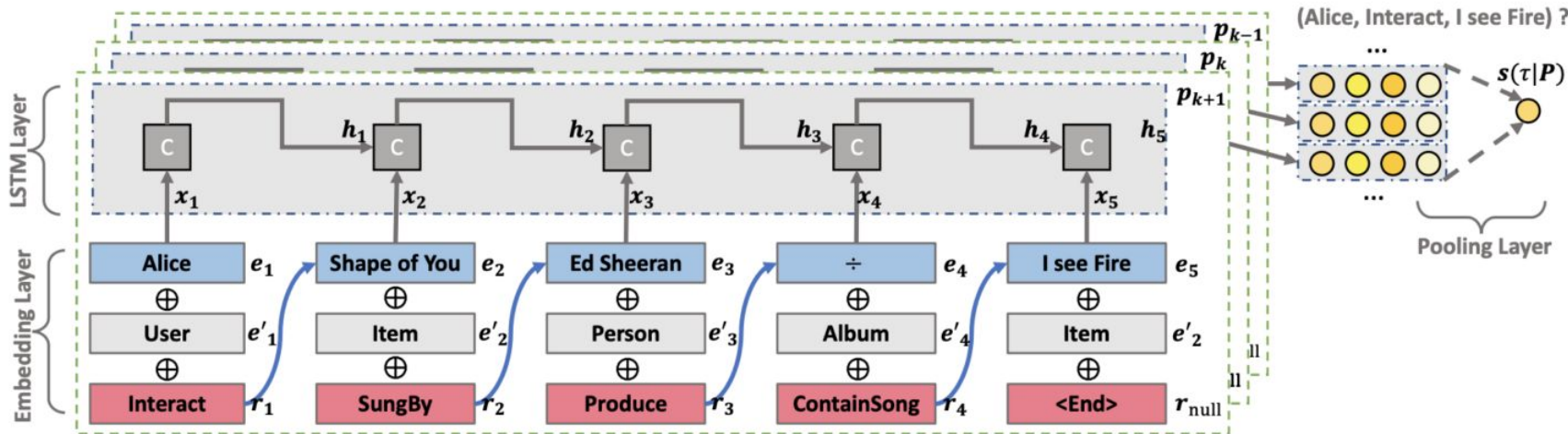
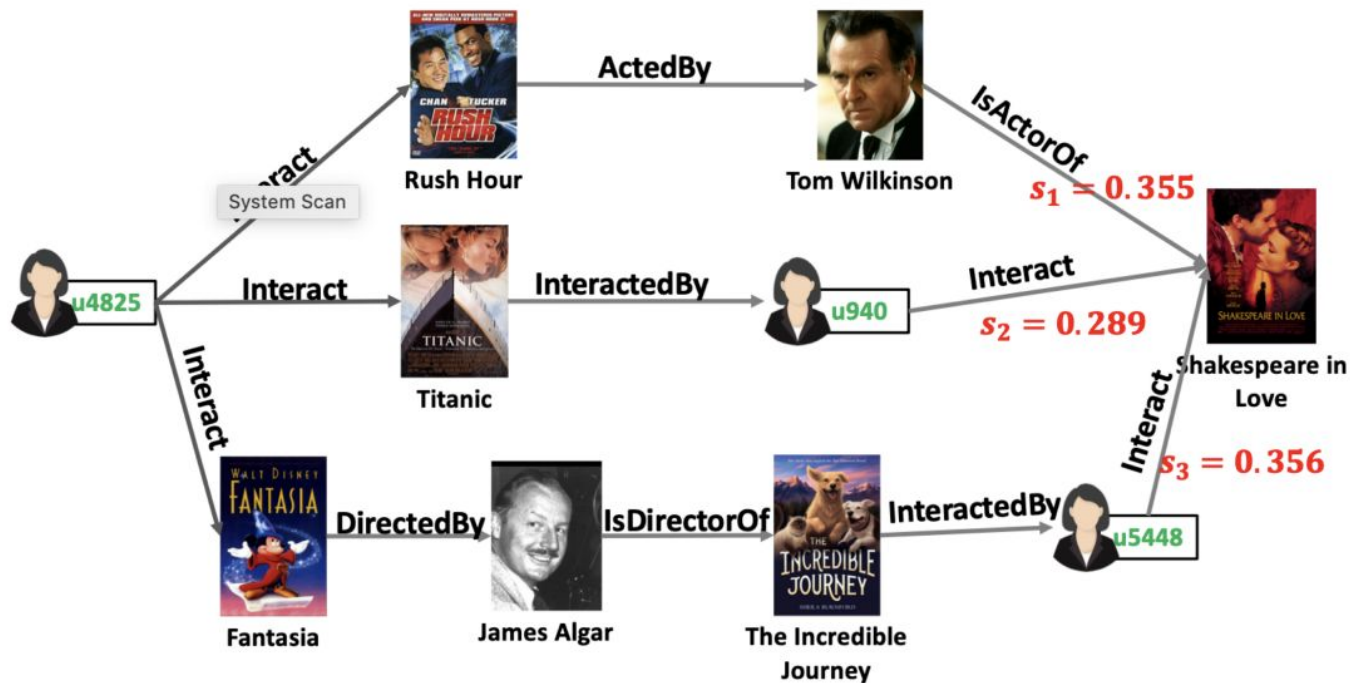


Figure 2: Schematic overview of our model architecture. The embedding layer contains 3 individual layers for entity, entity type, and relation type, respectively. The concatenation of the 3 embedding vectors is the input of LSTM for each path.

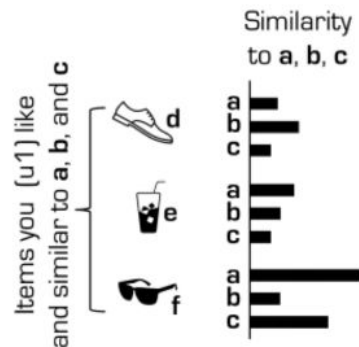
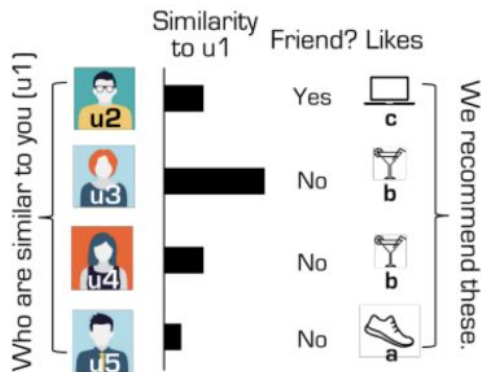
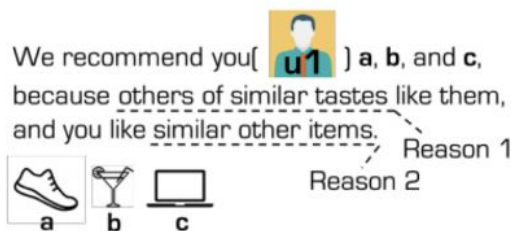
Explainable Reasoning over Knowledge Graphs for Recommendation



Выбираем третий путь,
с максимальной
вероятностью

UniWalk: Explainable and Accurate Recommendation for Rating and Network Data

- Haekyu P., Hyunsik J., Junghwan K. 2017
- **Вид объяснения:** Рекомендуем вам X,Y,Z, потому что близкие к вам пользователи предпочли эти товары (один из них-ваш друг), и эти товары близки к тем, которые вы уже покупали: M,K,P.
- **Подход:** knowledge graph

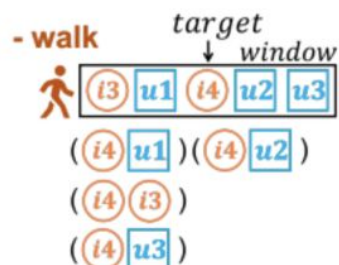
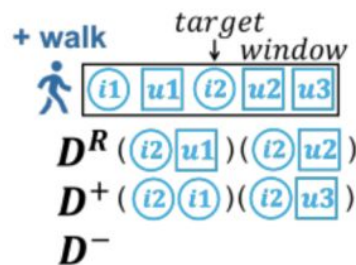
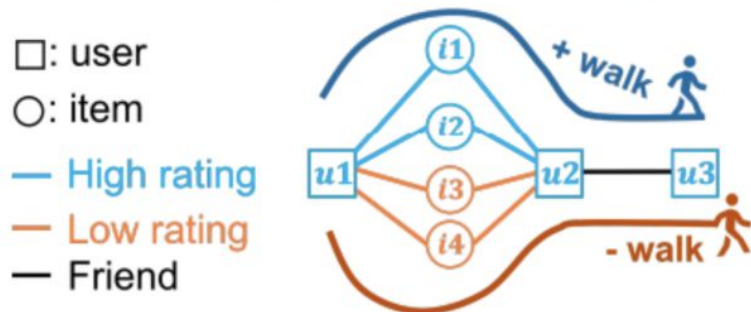


UniWalk: Explainable and Accurate Recommendation for Rating and Network Data

$$(3.2) \quad L = \underbrace{\sum_{(u,i) \in \mathcal{D}^R} \frac{1}{2} (r_{ui} - \hat{r}_{ui})^2}_{\text{Supervised term}} + \underbrace{\frac{\lambda_b}{2} \|\mathbf{b}\|^2 + \frac{\lambda_z}{2} \|\mathbf{Z}\|_F^2}_{\text{Regularization term}}$$

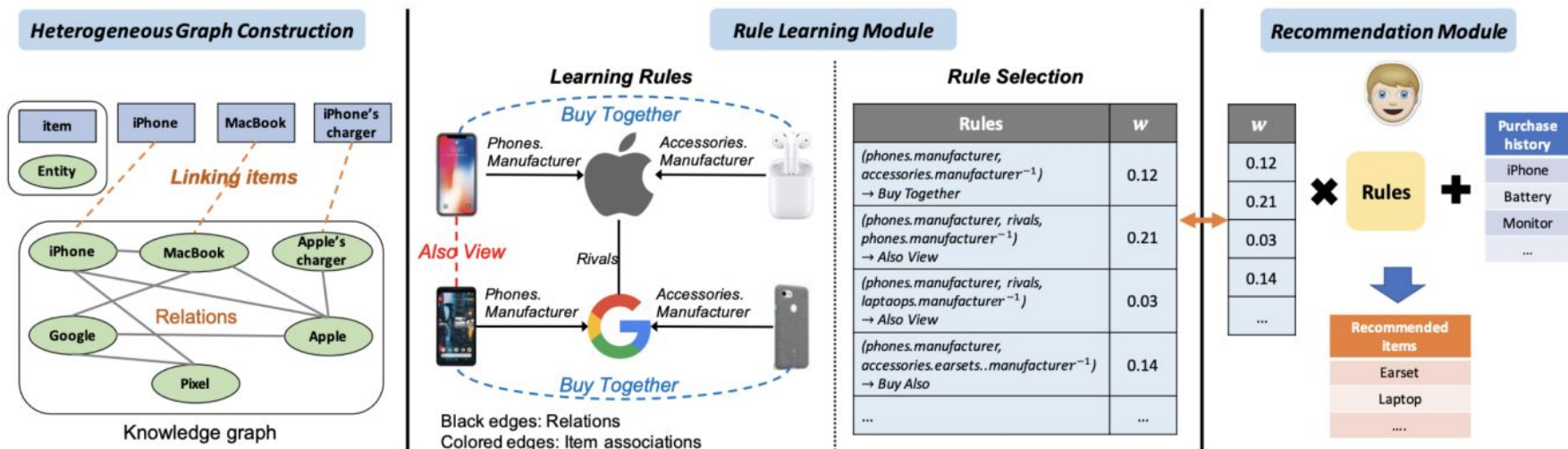
$$(3.3) \quad + \alpha \cdot \underbrace{\sum_{(v,w) \in \mathcal{D}^+} -\mathbf{z}_v^T \mathbf{z}_w}_{\text{Positive unsupervised term}} + \beta \cdot \underbrace{\sum_{(v,w) \in \mathcal{D}^-} \mathbf{z}_v^T \mathbf{z}_w}_{\text{Negative unsupervised term}}$$

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{z}_u^T \mathbf{z}_i$$

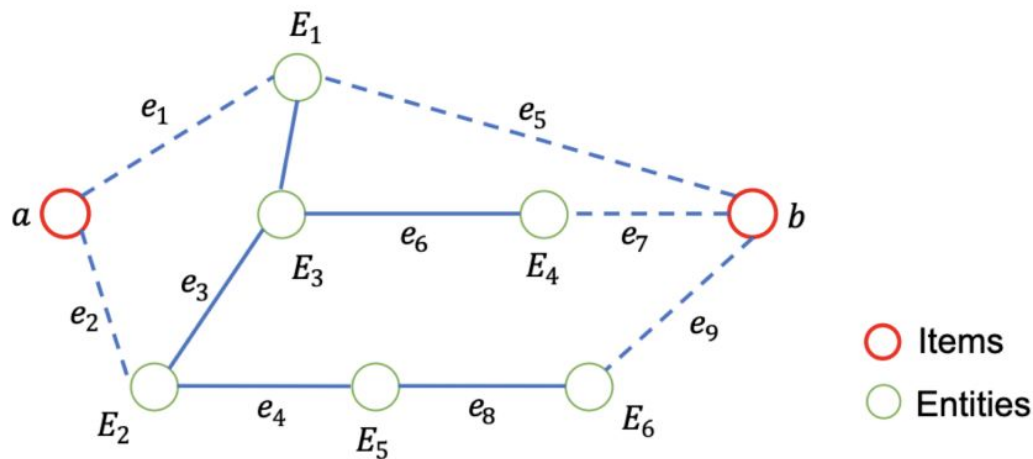


Jointly Learning Explainable Rules for Recommendation with Knowledge Graph

- **Вид объяснения:** Windows Phone, потому что вы купили “Surface Pro” (планшет от Windows)
- **Подход:** knowledge graph



Jointly Learning Explainable Rules for Recommendation with Knowledge Graph



Edge type:

$e_1, e_2, e_5, e_7, e_9 \in r_1$

$e_3, e_4 \in r_2$

$e_6, e_8 \in r_3$

Paths between a and b :

$P_1 = e_1 E_1 e_5$

$P_2 = e_2 E_2 e_3 E_3 e_6 E_4 e_7$

$P_3 = e_2 E_2 e_4 E_5 e_8 E_6 e_9$

Related reasoning rules:

Rules for P_1 : $R_1 = r_1 r_1$

Rules for P_2, P_3 : $R_2 = r_1 r_2 r_3 r_1$

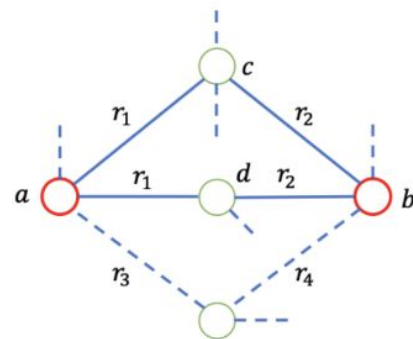
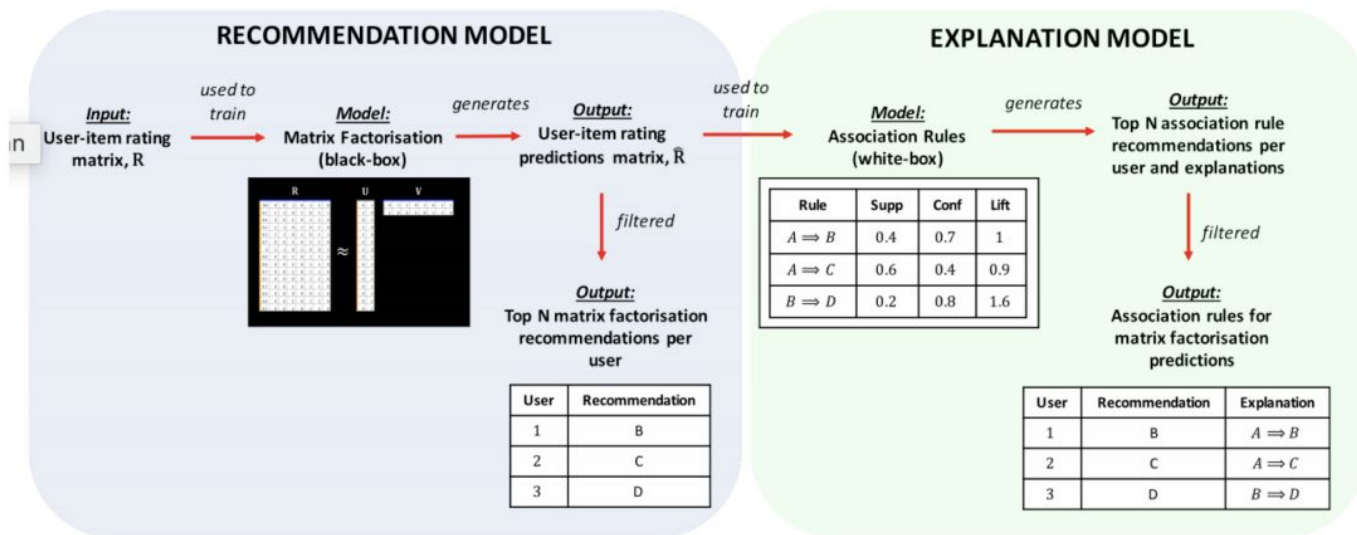


Figure 4: An example of a graph between items a and b . r represents a edge type or a relation type.

Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems

Peake G., Wang J. 2018

- **Вид объяснения:** Если пользователь покупает кофе, он, вероятно, купит сливки.
- **Подход:** Матричные разложения



Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems

Algorithm 1 Approximate matrix factorisation using global association rules

Input: Matrix factorisation predictions \hat{R} ; training data

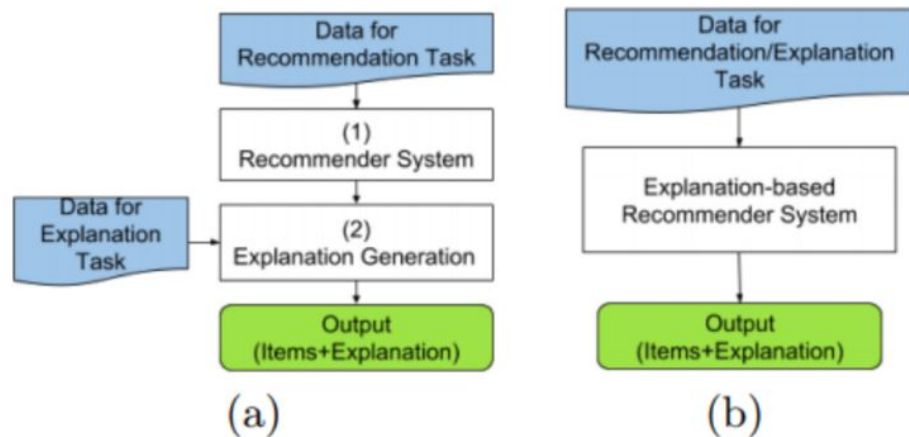
- 1: For each user i , generate a transaction list T_i of the index of top D matrix factorisation predictions, \hat{R}_i
- 2: Generate the set \mathcal{Z}_i of rules $(X \Rightarrow Y)$ that satisfy *min_supp*, *min_conf*, *min_lift* criteria from all transactions T .
Rules generated by the apriori algorithm using the *apyori* [27] Python package
- 3: **for all** users, $i = 1 \dots N$ **do**
- 4: Compute the list $\{\text{unseen}\}$ of items Y where $X \Rightarrow Y$ if $X \in \{\text{train}\}$ and $Y \notin \{\text{train}\}$.
- 5: Order $\{\text{unseen}\}$ by *supp/conf/lift*. Compute $\{\text{recommended}\} = \{\text{unseen}\}[: \text{top_n}]$
- 6: Return list of recommended items, $\{\text{recommended}\}$ and corresponding rules $X \Rightarrow Y$ as explanations.
- 7: **end for**

Output: $\{\text{unseen}\}$

Explainable Matrix Factorization for Collaborative Filtering

Abdollahi B., Nasraoui O. 2017

- **Вид объяснения:** доля пользователей $X\%$ с похожими предпочтениями оценили рекомендуемый объект.
- Оптимизируем объяснения совместно с предсказаниями
- Если товар j объясним, то должен быть близок к вектору пользователя
- **Подход:** Матричные разложения



TEM: Tree-enhanced Embedding Model for Explainable Recommendation

- **Вид объяснения:** Про пользователя известно, что он стремится к тишине. Ему советуют достопримечательность N, потому у нее Item_tag = "Walk_around".
- **Подход:** Деревья решений

$$\begin{cases} e_{avg}(u, i, \mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{v_l \in \mathcal{V}} w_{uil} v_l, \\ e_{max}(u, i, \mathcal{V}) = \max_{v_l \in \mathcal{V}} (w_{uil} v_l), \end{cases}$$

$$\begin{cases} w'_{uil} = \mathbf{h}^\top \text{ReLU}(\mathbf{W} ([p_u \odot q_i, v_l]) + \mathbf{b}) \\ w_{uil} = \frac{\exp(w'_{uil})}{\sum_{(u, i, x) \in \mathcal{O}} \exp(w'_{uil})} \end{cases}$$

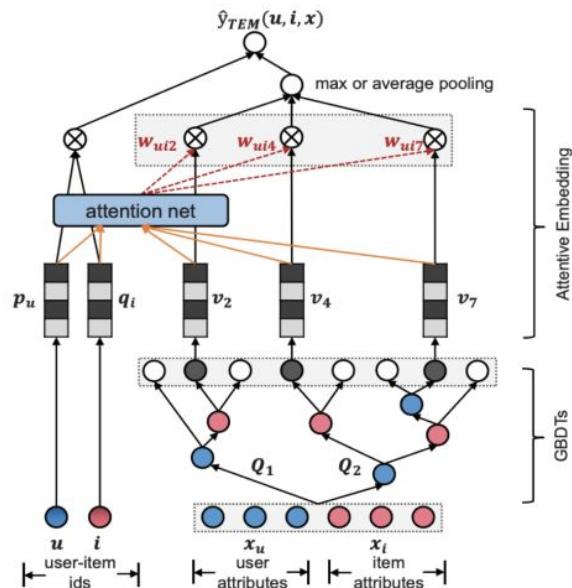


Figure 2: Illustrative architecture of our TEM framework.