When Recommender Systems Meet Network Representation Learning

王鸿伟

June 3, 2018

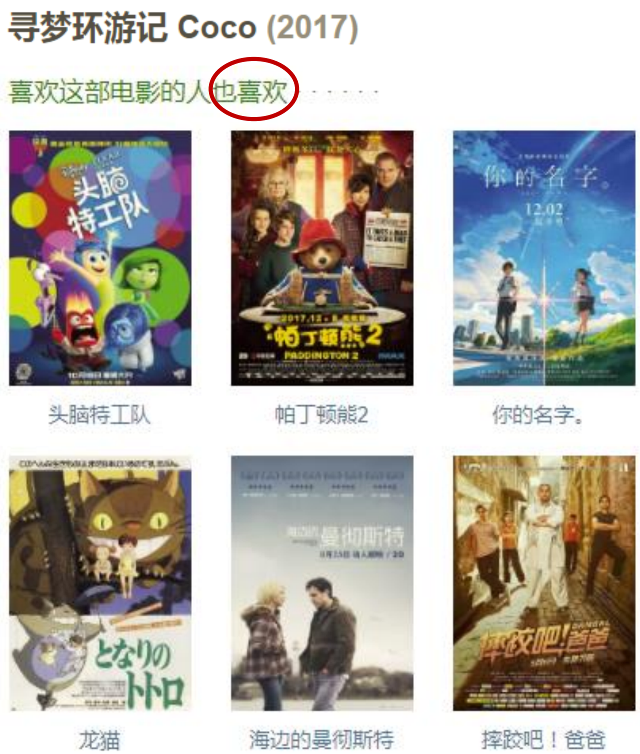
Outline

- Recommender systems (RS)

- Network representation learning (NRL)
  - GraphGAN [AAAI 18]
  - Knowledge graph embedding

- RS + NRL
  - One-by-one learning [WWW 18]
  - Alternate learning [NIPS 18 in submission]
  - Joint learning [CIKM 17][WSDM 18][CIKM 18 in submission]
  - Pros and cons

Recommender Systems



Movie

Book

Recommender Systems

Music

Restaurant

Recommender Systems

QA

Video

News

Recommender Systems

Recommender systems (RS) intend to address the information explosion by finding a small set of items for users to meet their personalized interests and demands

RS Task

- Recommendation Task 1: Rating / Click-through rate (CTR) prediction

2	?	3	?
?	?	4	?
?	5	?	2
3	1	4	?

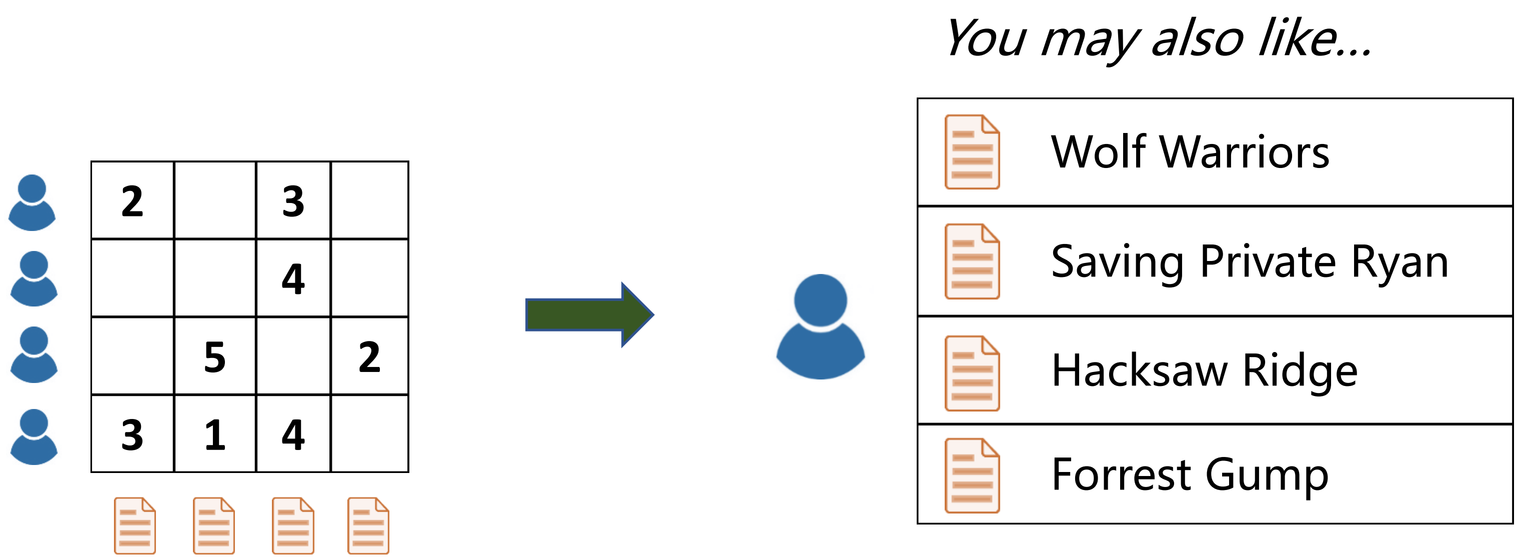
explicit feedback

1	?	0	?
?	?	1	?
?	0	?	1
0	1	0	?

implicit feedback

RS Task

- Recommendation Task 2: Top-K recommendation



Collaborative Filtering

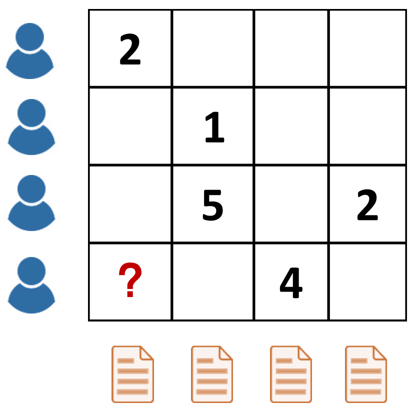
- **Collaborative filtering (CF)** considers users' historical interactions and makes recommendations based on their potential common preferences
  - Matrix factorization (MF)

$$R_{pq} = \mathbf{p}^\top \mathbf{q}$$

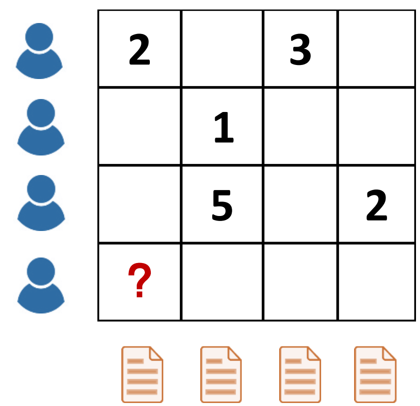
$$L = \| \boldsymbol{R} - \boldsymbol{P}^{\mathrm{T}} \boldsymbol{Q} \|_2^2 + \| \boldsymbol{P} \|_2^2 + \| \boldsymbol{Q} \|_2^2$$

CF fails to address…

- Sparsity of user-item interactions
- Cold start problem

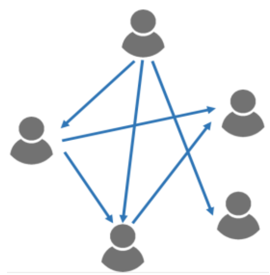


sparsity



cold start

CF + Side Information



Social network

Alice
Female
California
…

iPhone X
2017
5.8 inch
$999
…

User/item attributes

Multimedia

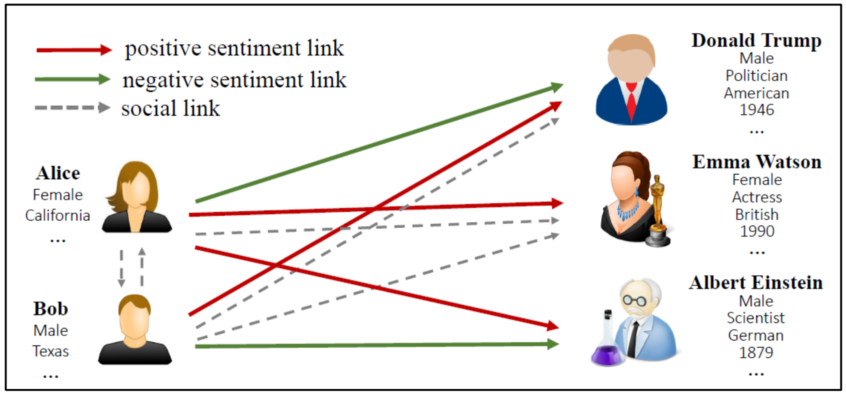
purchase

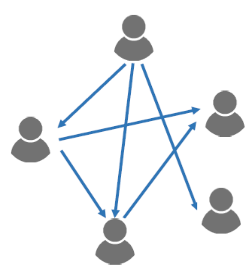
time: 20:10
location： Beijing
What else in carts： …

Contexts

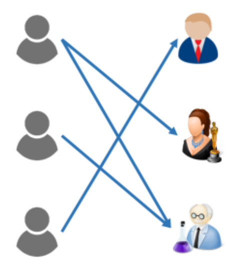
Side Information with Network Structure

Celebrity recommendation in Weibo

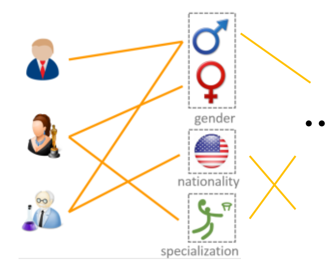




user-end
social network



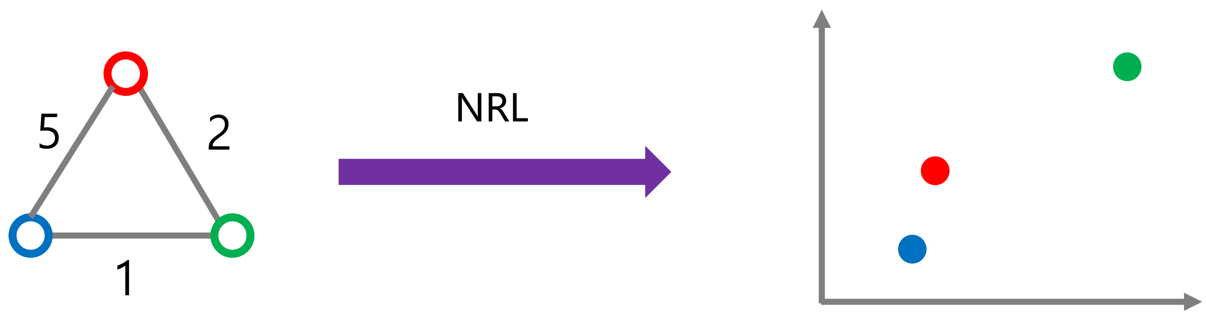
user-item
interaction



item-end
knowledge graph

Network Representation Learning

- **Network representation learning (NRL)** tries to embed each node of a network into a low-dimensional vector space, which preserves the structural similarities or distances among the nodes in the network
- $G(V, E) \rightarrow \mathbf{E} \in \mathbb{R}^{|V| \times d}$
- NRL can be viewed as a **dimension reduction** technology
- a.k.a network embedding / graph representation learning / graph embedding
- **Applications**: node classification, link prediction, clustering, anomaly detection, social network analysis, etc.

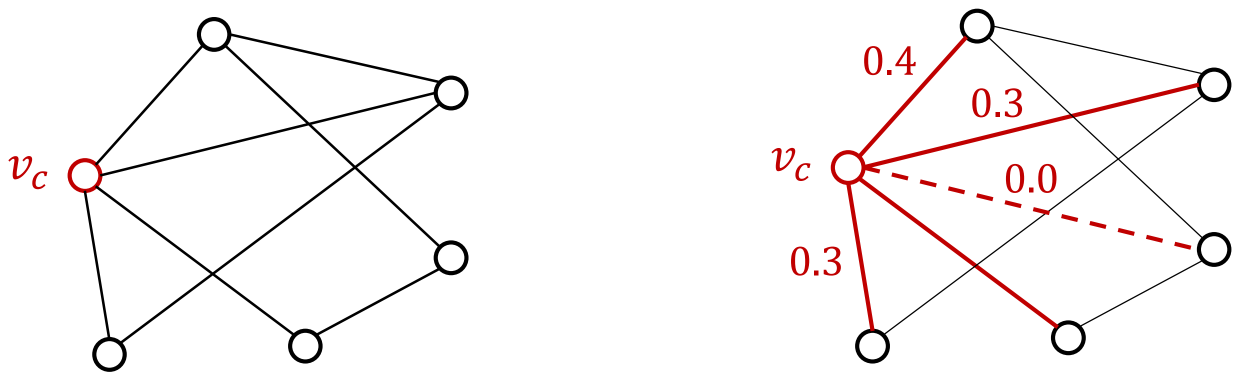


Network Representation Learning

- **Traditional dimension reduction methods**
  - **PCA** (principle component analysis)
  - **LDA** (linear discriminant analysis)
  - **MDS** (multiple dimensional scaling)
- **Manifold Learning methods**
  - **Isomap** (isometric mapping) [Science 2000]
  - **LLE** (locally linear embedding) [Science 2000]
  - **LE** (Laplacian eigenmaps) [NIPS 2001]
- **Random-walk-based methods**
  - **DeepWalk** [KDD 2014]
  - **Node2vec** [KDD 2016]
- **Deep-learning-based methods**
  - **SDNE** (structural deep network embedding) [KDD 2016]
  - **HNE** (heterogeneous network embedding) [KDD 2015]
- **Others**
  - **LINE** (large-scale information network embedding) [WWW 2015]
  - **GraphSAGE** (sample and aggregate) [NIPS 2017]

Generative Models

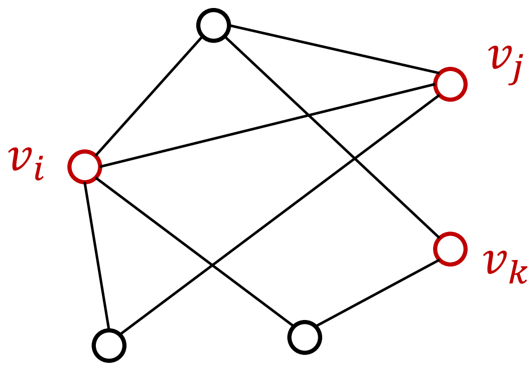
- **Generative** graph representation learning model assumes an underlying true connectivity distribution $p\_true(v|v\_c)$ for each vertex v_c
  - The edges can be viewed as observed samples generated by $p\_true(v|v\_c)$
  - Vertex embeddings are learned by maximizing the likelihood of edges
  - E.g., DeepWalk and node2vec



Original graph $p_{true}(v|v_c)$

Discriminative Models

- **Discriminative** graph representation learning model aims to learn a classifier for predicting edges directly
  - Consider two vertices v_i and v_j jointly as features and predict the probability of an edge existing between them, i.e., $p(edge|v\_i,v\_j)$
  - E.g., SDNE and PPNE



$$p(edge|v_i, v_j) = 0.8$$
$$p(edge|v_i, v_k) = 0.3$$
$$\ldots\ldots$$

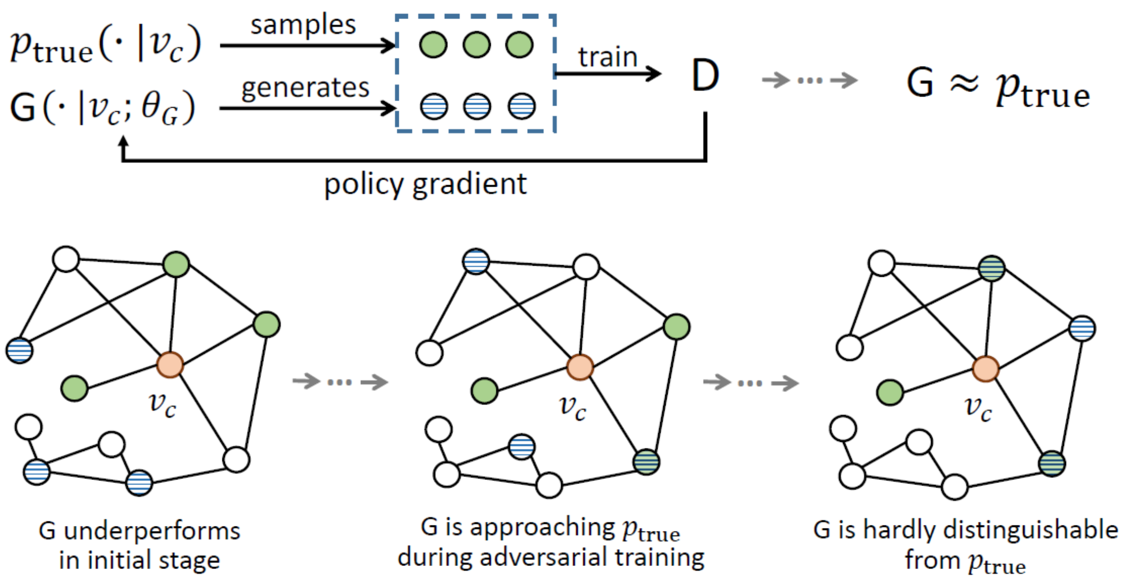
G + D ?

- **GraphGAN**: a framework unifying generative and discriminative models for graph representation learning
- Objective:
  - $G(v \mid v\_c;\ \theta\_G)$: trying to approximate $p\_true(v\_c)$
  - $D(v,\ v\_c;\ \theta\_D)$: aiming to discriminate the connectivity for the vertex pair $(v,\ v\_c)$
- The two-player minimax game:

$$\min_{\theta_G} \max_{\theta_D} V(G, D) = \sum_{c=1}^{V} \Big(\mathbb{E}_{v \sim p_{\text{true}}(\cdot|v_c)} \big[\log D(v, v_c; \theta_D) \big] + \mathbb{E}_{v \sim G(\cdot|v_c;\theta_G)} \big[\log \left(1 - D(v, v_c; \theta_D)\right) \big] \Big)$$

GraphGAN [AAAI 18]

$$\min_{\theta_G}\max_{\theta_D} V(G,D) = \sum_{c=1}^{V}\Big(\mathbb{E}_{v\sim p_{\mathrm{true}}(\cdot|v_c)}\big[\log D(v,v_c;\theta_D)\big]+ \mathbb{E}_{v\sim G(\cdot|v_c;\theta_G)}\big[\log\big(1-D(v,v_c;\theta_D)\big)\big]\Big)$$



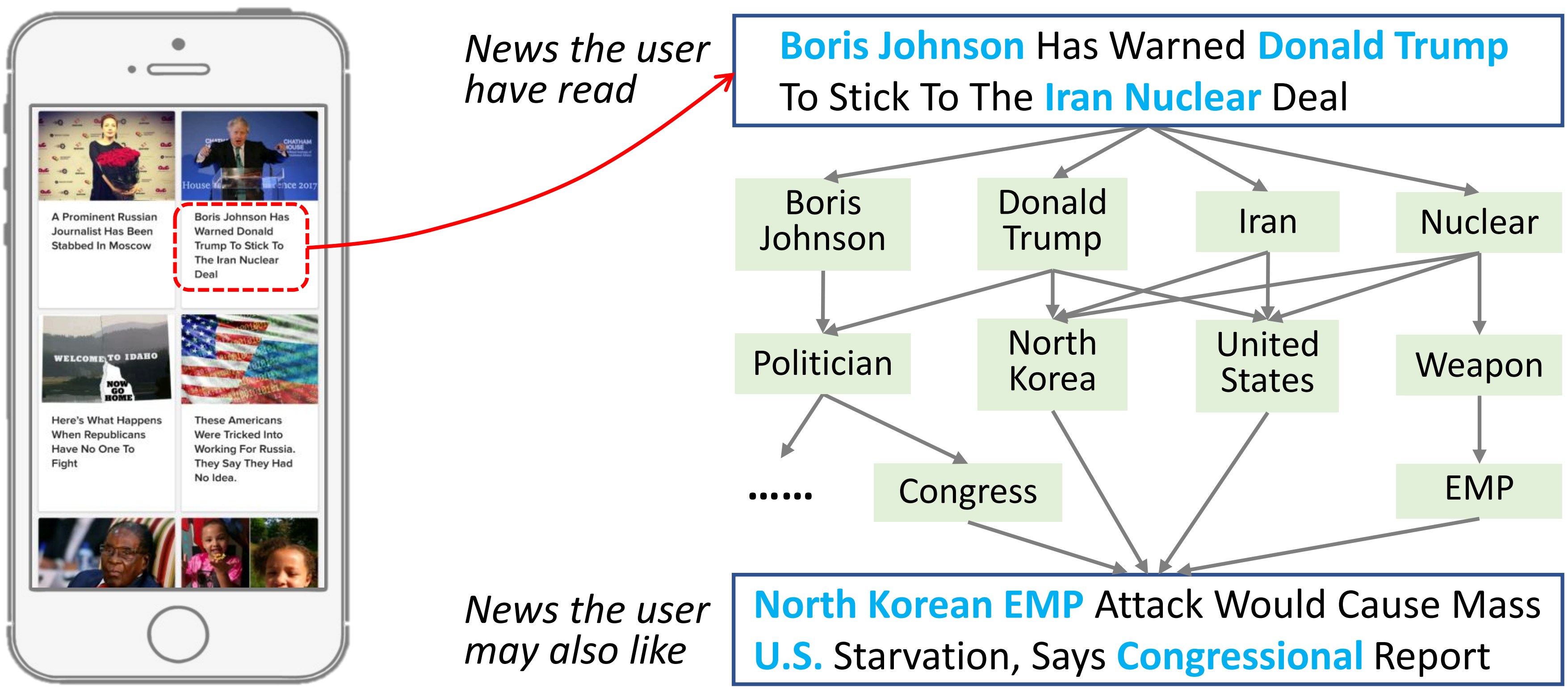
Knowledge Graph

- **Knowledge graph (KG)** is a semantic network in which **nodes** correspond to **entities** and **edges** correspond to **relations**
- A KG usually consists of massive **triples** (head, relation, tail)
- E.g.,

Forrest Gump $\xrightarrow{\textit{film.film.director}}$ Tom Hanks

Tom Hanks $\xrightarrow{\textit{people.people.date\_of\_brith}}$ July 7, 1956

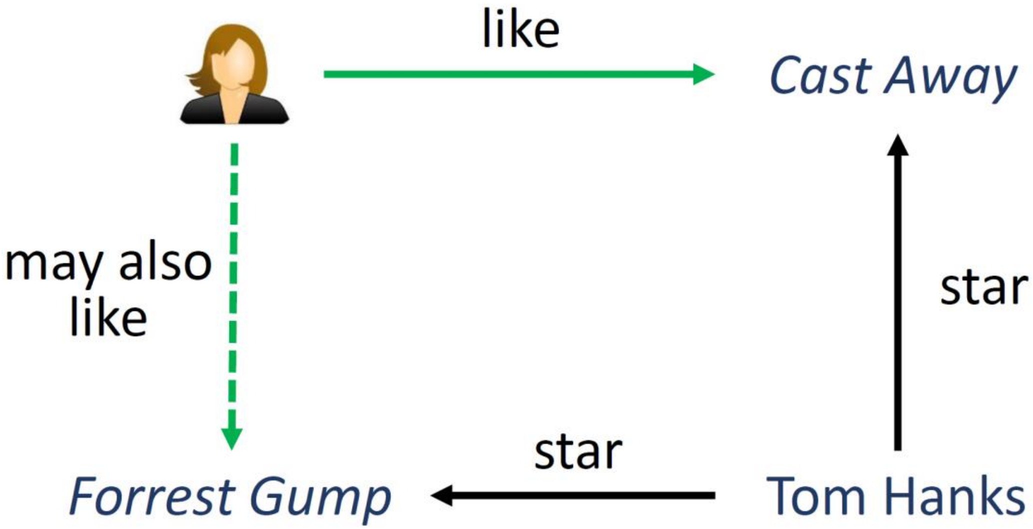
Why Using KG in RS ?



There is more…

Advantages of using KG in RS

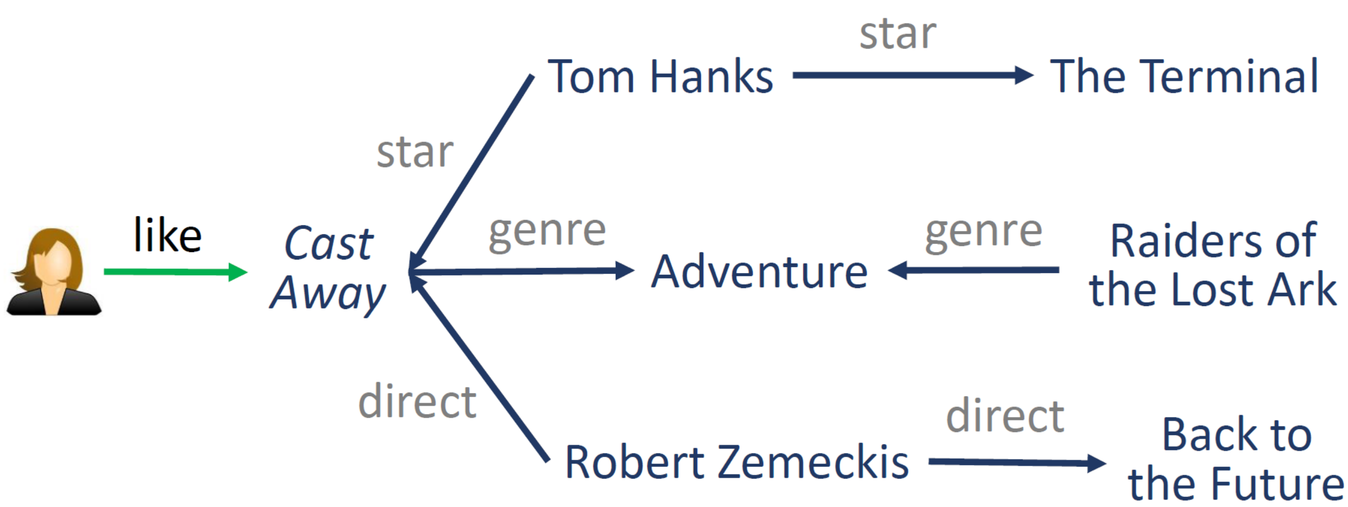
- **Precision**



There is more…

Advantages of using KG in RS

- **Diversity**



There is more...

Advantages of using KG in RS

- **Explainability**

like → *Cast Away*

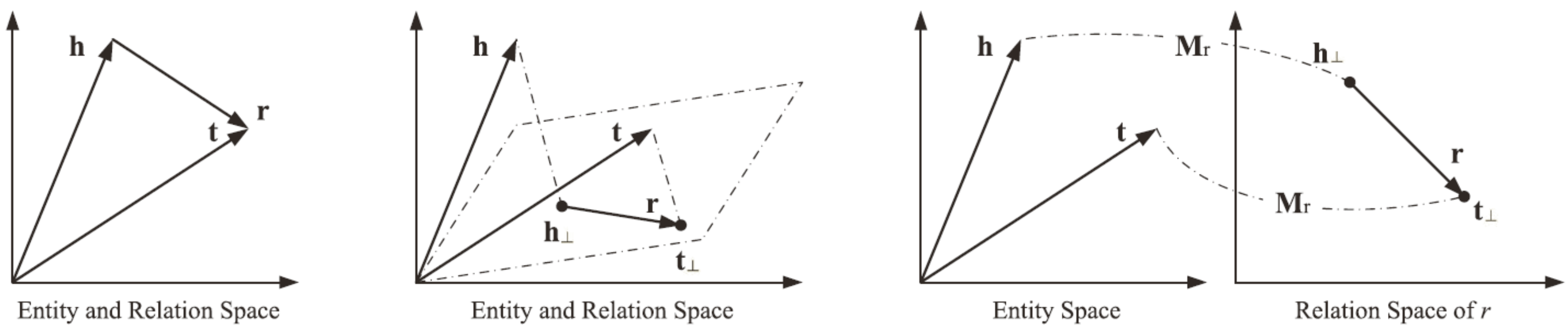
You may also like ...

The Terminal, as they share the same star;
Raiders of the Lost Ark, as they share the same genre;
Back to the Future, as they share the same director;

......

Knowledge Graph Embedding

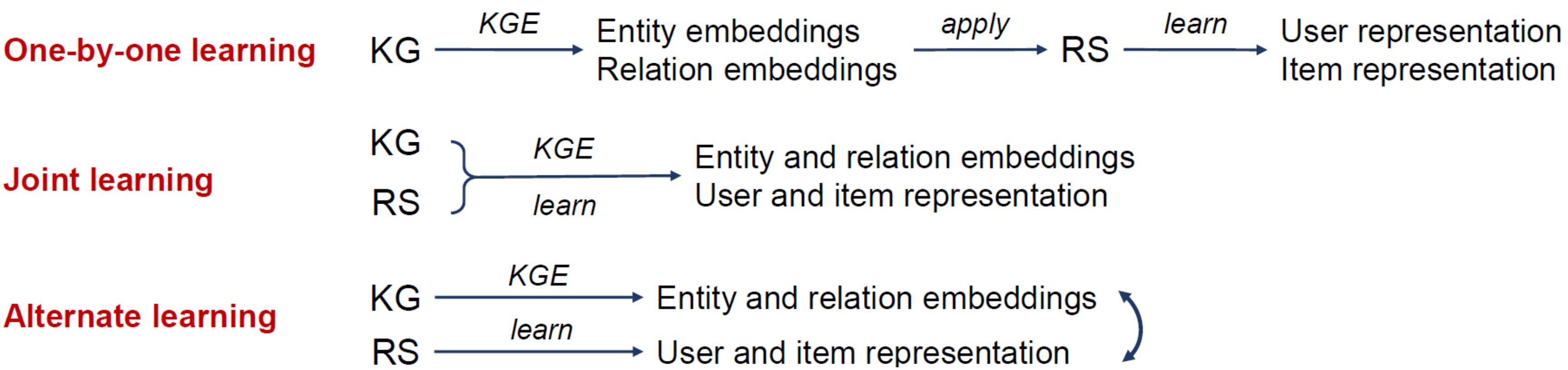
- **Knowledge graph embedding (KGE)** aims to learn a low-dimensional representation vector for each entity and relation in a KG
- Translational distance models (TransX)
  - TransE: $f_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$
  - TransH: $f_r(h,t) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2$, where $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\mathrm{T}\mathbf{h}\mathbf{w}_r$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\mathrm{T}\mathbf{t}\mathbf{w}_r$
  - TransR: $f_r(h,t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_2^2$, where $\mathbf{h}_r = \mathbf{h}\mathbf{M}_r$ and $\mathbf{t}_r = \mathbf{t}\mathbf{M}_r$



(a) TransE. (b) TransH. (c) TransR.

RS + NRL ?

- If we treat RS and NRL as two tasks, there are three possible ways of combining RS and NRL together…
  - **One-by-one learning**
  - **Alternate learning**
  - **Joint learning**



One-by-one Learning: DKN [WWW 2018]

Knowledge distillation

Trump praises **Las Vegas** medical team

Apple CEO Tim Cook: **iPhone 8** and **Apple Watch Series 3** are sold out in some places

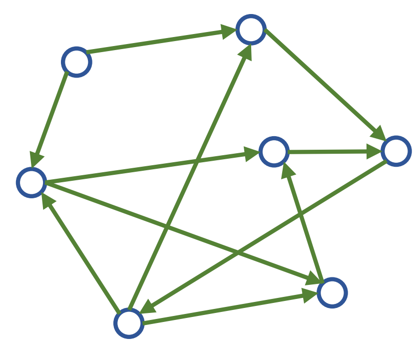
EU Spain: **Juncker** does not want **Catalonian** independence

……

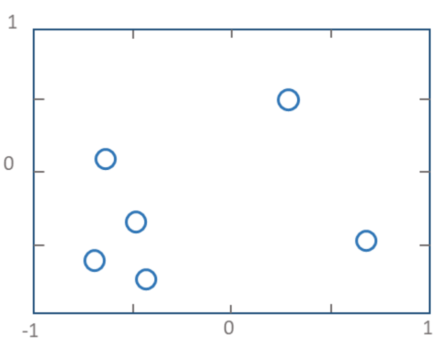
Entity linking

Donald Trump: *Donald Trump is the 45th president …*
Las Vegas: *Las Vegas is the 28th-most populated city …*
Apple Inc.: *Apple Inc. is an American multinational …*
CEO: *A chief executive officer is the position of the …*
Tim Cook: *Timothy Cook is an American business …*
iPhone 8: *iPhone 8 is smartphone designed, …*
……

Knowledge subgraph construction



Knowledge graph embedding

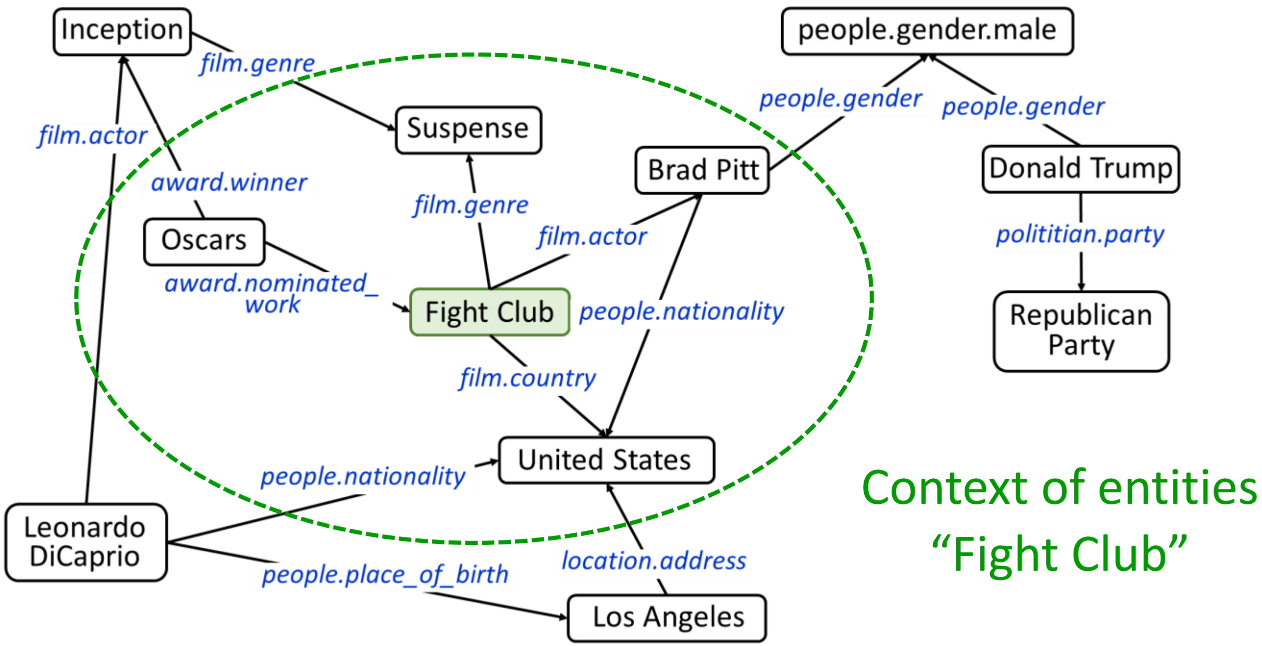


Entity embedding

Donald Trump: (0.32, 0.48)
Las Vegas: (0.71, -0.49)
Apple Inc.: (-0.48, -0.41)
CEO: (-0.57, 0.06)
Tim Cook: (-0.61, -0.59)
iPhone 8: (-0.46, -0.75)

DKN

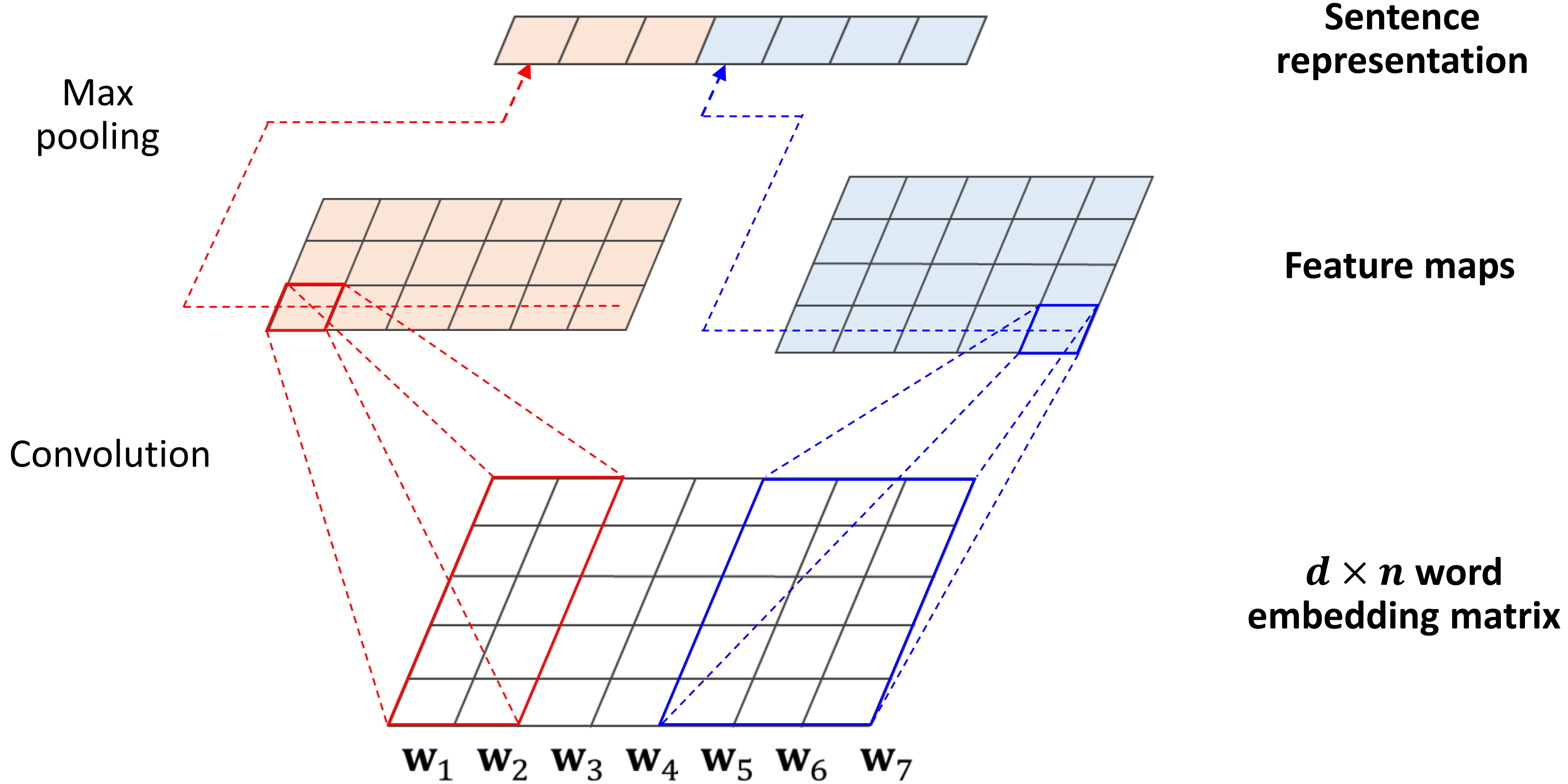
Context embedding



$$\bar{\mathbf{e}} = \frac{1}{|context(e)|} \sum_{e_i \in context(e)} \mathbf{e}_i$$

DKN

Kim CNN

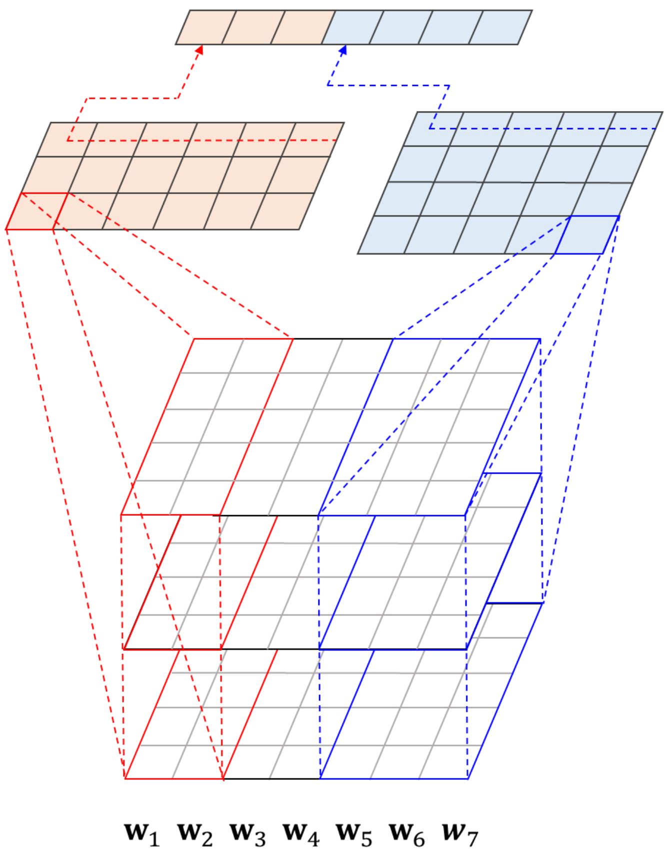


$w_{1:n}$ = [*Donald Trump praises Las Vegas medical team*]

Sentence

DKN

Knowledge-aware CNN



pooling

CNN layer

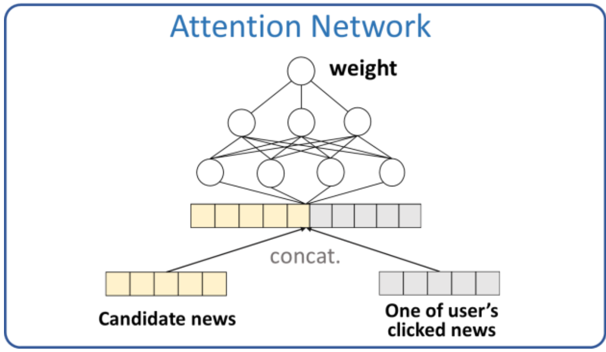
$d \times n$ transformed context embeddings

$d \times n$ transformed entity embeddings

$d \times n$ word embeddings

multiple channels

DKN



Attention net:

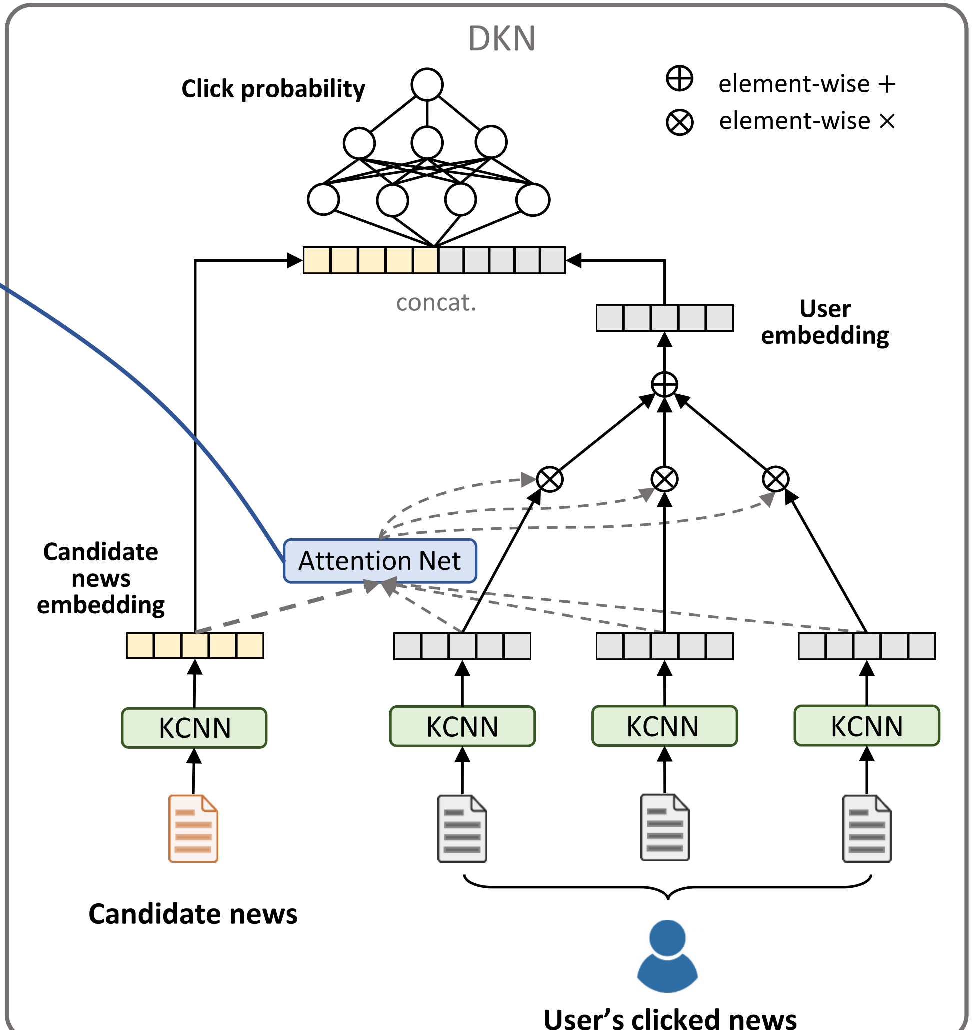
$$s_{t_k^i, t_j} = \text{softmax}\left(\mathcal{H}\left(\mathrm{e}(t_k^i), \mathrm{e}(t_j)\right)\right) = \frac{\exp\left(\mathcal{H}\left(\mathrm{e}(t_k^i), \mathrm{e}(t_j)\right)\right)}{\sum_{k=1}^{N_i} \exp\left(\mathcal{H}\left(\mathrm{e}(t_k^i), \mathrm{e}(t_j)\right)\right)}$$

User interest extraction:

$$\mathrm{e}(i) = \sum_{k=1}^{N_i} s_{t_k^i, t_j} \mathrm{e}(t_k^i).$$

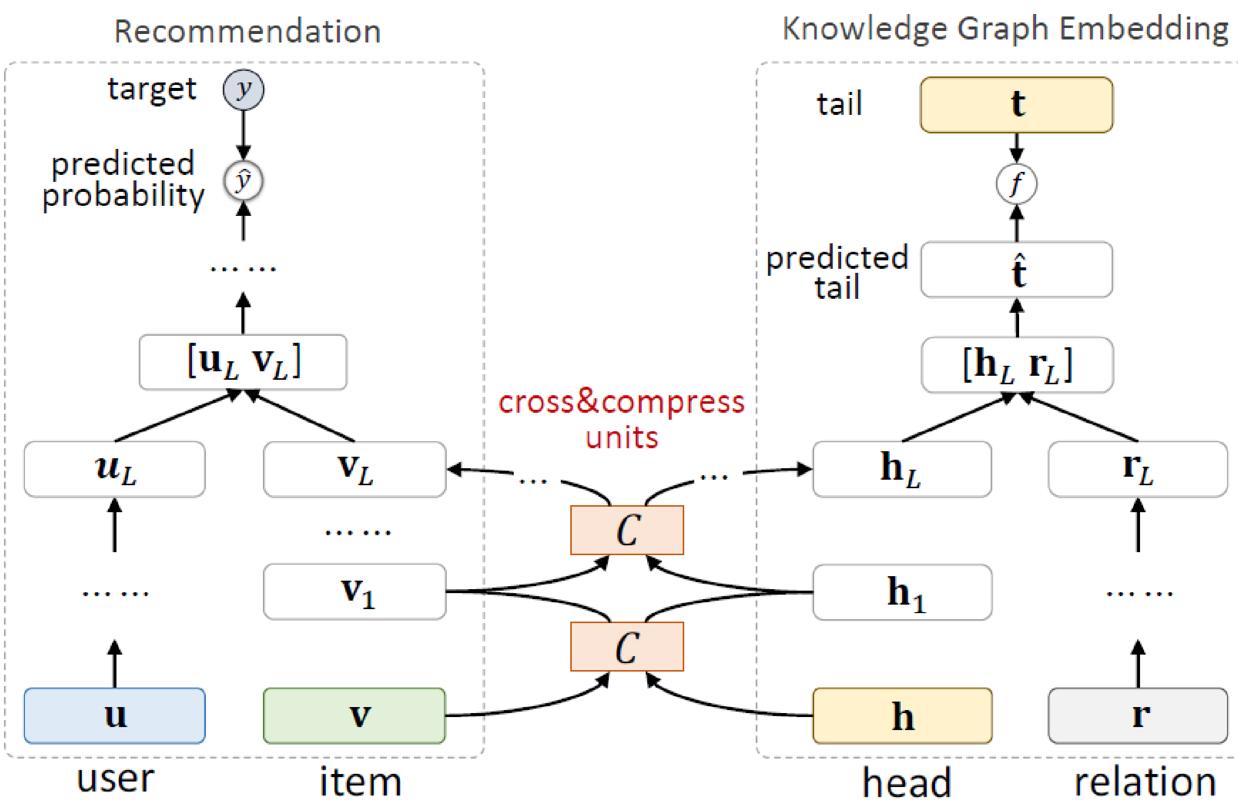
CTR prediction:

$$p_{i,t_j} = \mathcal{G}\left(\mathrm{e}(i), \mathrm{e}(t_j)\right)$$



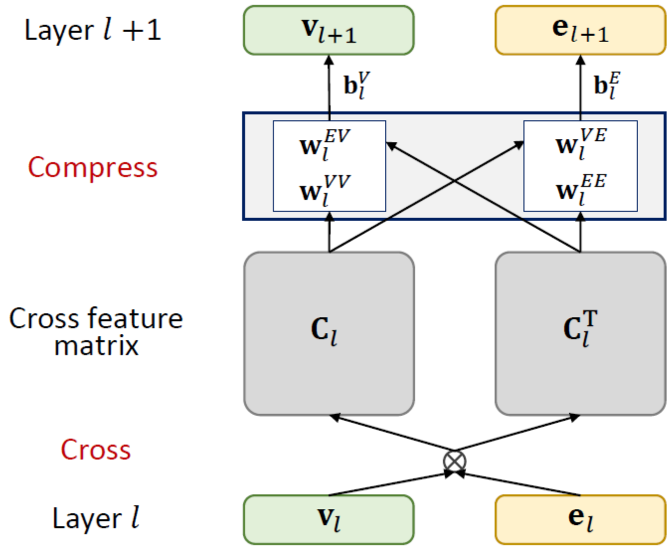
Alternate Learning: MKR [NIPS in sub.]

Multi-task learning for Knowledge graph enhanced Recommendation (MKR)



MKR

Cross&compress unit



$$\mathbf{C}_l = \mathbf{v}_l \mathbf{e}_l^\top = \begin{bmatrix} v_l^{(1)} e_l^{(1)} & \cdots & v_l^{(1)} e_l^{(d)} \\ \cdots & & \cdots \\ v_l^{(d)} e_l^{(1)} & \cdots & v_l^{(d)} e_l^{(d)} \end{bmatrix}$$

$$\mathbf{v}_{l+1} = \mathbf{C}_l \mathbf{w}_l^{VV} + \mathbf{C}_l^\top \mathbf{w}_l^{EV} + \mathbf{b}_l^V = \mathbf{v}_l \mathbf{e}_l^\top \mathbf{w}_l^{VV} + \mathbf{e}_l \mathbf{v}_l^\top \mathbf{w}_l^{EV} + \mathbf{b}_l^V$$

$$\mathbf{e}_{l+1} = \mathbf{C}_l \mathbf{w}_l^{VE} + \mathbf{C}_l^\top \mathbf{w}_l^{EE} + \mathbf{b}_l^E = \mathbf{v}_l \mathbf{e}_l^\top \mathbf{w}_l^{VE} + \mathbf{e}_l \mathbf{v}_l^\top \mathbf{w}_l^{EE} + \mathbf{b}_l^E,$$

MKR

Theoretical Analysis

- **Polynomial approximation**

Theorem 1 *Denote the input of item and entity in MKR network as* $\mathbf{v} = [v_1 \cdots v_d]^\top$ *and* $\mathbf{e} = [e_1 \cdots e_d]^\top$*, respectively. Then the cross terms about* $\mathbf{v}$ *and* $\mathbf{e}$ *in* $\|\mathbf{v}_L\|_1$ *and* $\|\mathbf{e}_L\|_1$ *(the L1-norm of* $\mathbf{v}_L$ *and* $\mathbf{e}_L$*) with maximal degree is* $k_{\boldsymbol{\alpha},\boldsymbol{\beta}} v_1^{\alpha_1} \cdots v_d^{\alpha_d} e_1^{\beta_1} \cdots e_d^{\beta_d}$*, where* $k_{\boldsymbol{\alpha},\boldsymbol{\beta}} \in \mathbb{R}$*,* $\alpha_i, \beta_i \in \mathbb{N}$ *for* $i \in \{1, \cdots, d\}$*,* $\alpha_1 + \cdots + \alpha_d = 2^{L-1}$ *and* $\beta_1 + \cdots + \beta_d = 2^{L-1}$ *(*$L \geq 1, \mathbf{v}_0 = \mathbf{v}, \mathbf{e}_0 = \mathbf{e}$*).*

sufficient approximation ability

MKR

Theoretical Analysis

- **Generalization**
  - **Factorization Machines**

Proposition 1 *The L1-norm of $\mathbf{v}_1$ and $\mathbf{e}_1$ can be written as the following form:*

$$\|\mathbf{v}_1\|_1 \; (or \; \|\mathbf{e}_1\|_1) = \left| b + \sum_{i=1}^{d} \sum_{j=1}^{d} \langle w_i, w_j \rangle v_i e_j \right|,$$

where $\langle w_i, w_j \rangle = w_i + w_j$.

  - **Deep&Cross Network**

Proposition 2 *In the formula of $\mathbf{v}_{l+1}$ in Eq. (2), if we restrict $\mathbf{w}_l^{VV}$ in the first term to satisfy $\mathbf{e}_l^\top \mathbf{w}_l^{VV} = 1$ and restrict $\mathbf{e}_l$ in the second term to be $\mathbf{e}_0$ (and impose similar restrictions on $\mathbf{e}_{l+1}$), the cross&compress unit is then conceptually equivalent to DCN layer in the sense of multi-task learning:*

$$\mathbf{v}_{l+1} = \mathbf{e}_0 \mathbf{v}_l^\top \mathbf{w}_l^{EV} + \mathbf{v}_l + \mathbf{b}_l^V, \quad \mathbf{e}_{l+1} = \mathbf{v}_0 \mathbf{e}_l^\top \mathbf{w}_l^{VE} + \mathbf{e}_l + \mathbf{b}_l^E.$$

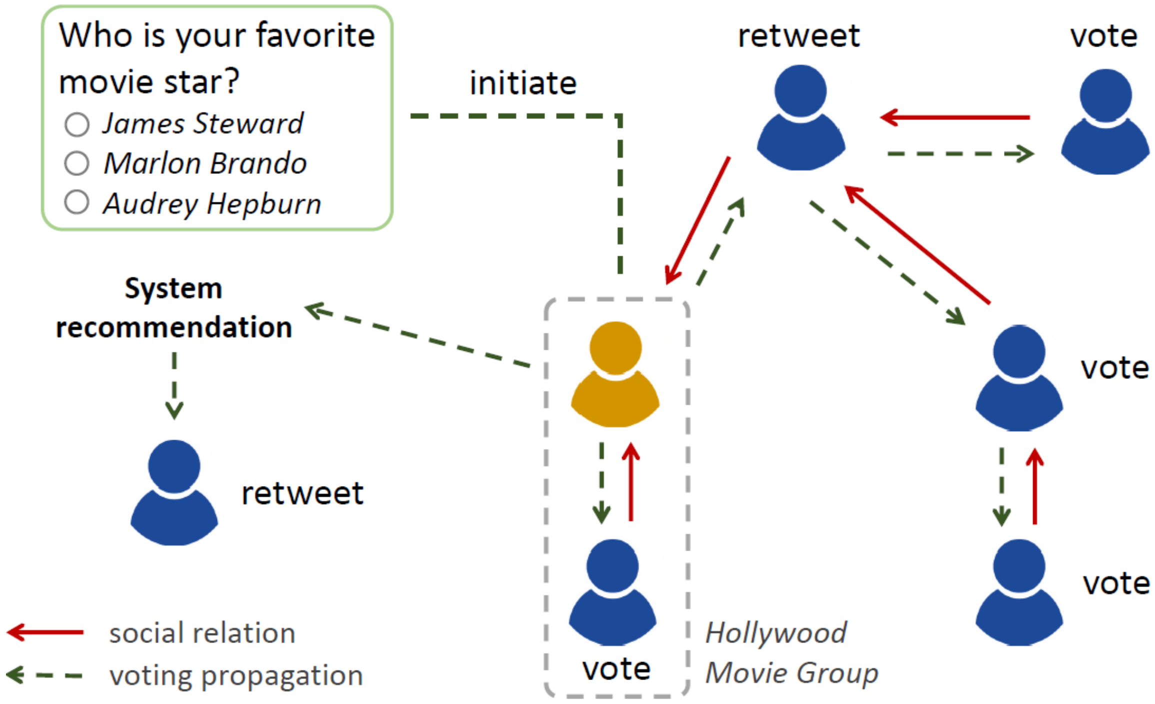
  - **Cross-stitch Network**

Proposition 3 *If we omit all biases in Eq. (2), the cross&compress unit can be written as*

$$\begin{bmatrix} \mathbf{v}_{l+1} \\ \mathbf{e}_{l+1} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_l^\top \mathbf{w}_l^{VV} & \mathbf{v}_l^\top \mathbf{w}_l^{EV} \\ \mathbf{e}_l^\top \mathbf{w}_l^{VE} & \mathbf{v}_l^\top \mathbf{w}_l^{EE} \end{bmatrix} \begin{bmatrix} \mathbf{v}_l \\ \mathbf{e}_l \end{bmatrix}.$$

Joint Learning: JTS-MF [CIKM 2017]

Joint Topic-Semantic-aware Matrix Factorization (JTS-MF)



JTS-MF

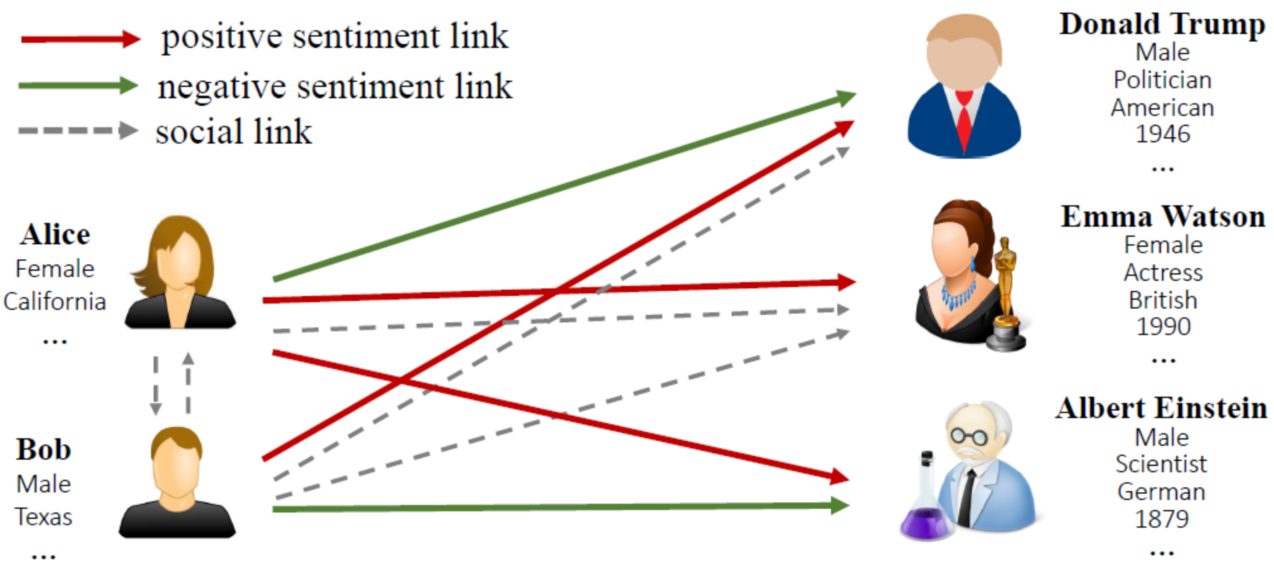
Joint Topic-Semantic-aware Matrix Factorization (JTS-MF)

$$
\begin{aligned}
L = & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I'_{i,j} \left(R_{i,j} - \boldsymbol{Q}_i \boldsymbol{P}_j^\top \right)^2 + \frac{\alpha}{2} \sum_{i=1}^{N} \left\| \boldsymbol{Q}_i - \sum_{k \in \mathcal{F}_i^+} \widehat{S}_{i,k} \boldsymbol{Q}_k \right\|_2^2 \\
& + \frac{\beta}{2} \sum_{i=1}^{N} \left\| \boldsymbol{Q}_i - \sum_{k \in \mathcal{G}_i} \widehat{G}_{i,k} \boldsymbol{Q}_k \right\|_2^2 + \frac{\gamma}{2} \sum_{j=1}^{M} \left\| \boldsymbol{P}_j - \sum_{t \in \mathcal{V}_j} \widehat{T}_{j,t} \boldsymbol{P}_t \right\|_2^2 + \frac{\lambda}{2} \left(\| \boldsymbol{Q} \|_F^2 + \| \boldsymbol{P} \|_F^2 \right)
\end{aligned}
$$

Q: user feature matrix
P: item feature matrix
$R_{i,j}$: rating
$\hat{S}, \hat{G}, \hat{T}$: similarity coefficient

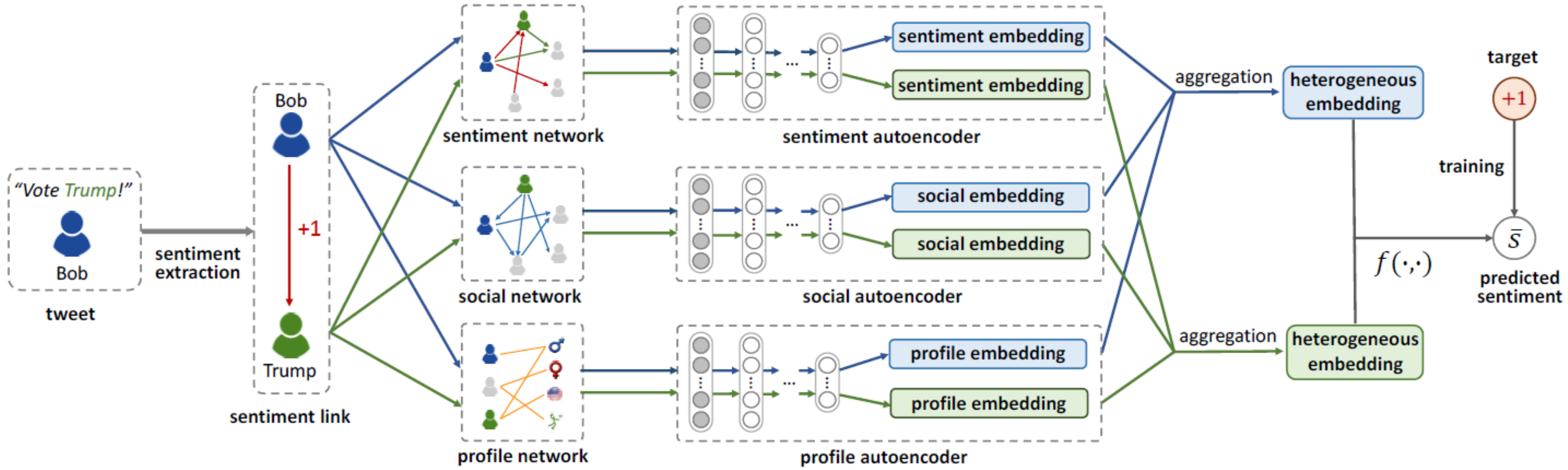
Joint Learning: SHINE [WSDM 2018]

Signed Heterogeneous Information Network Embedding (SHINE)



SHINE

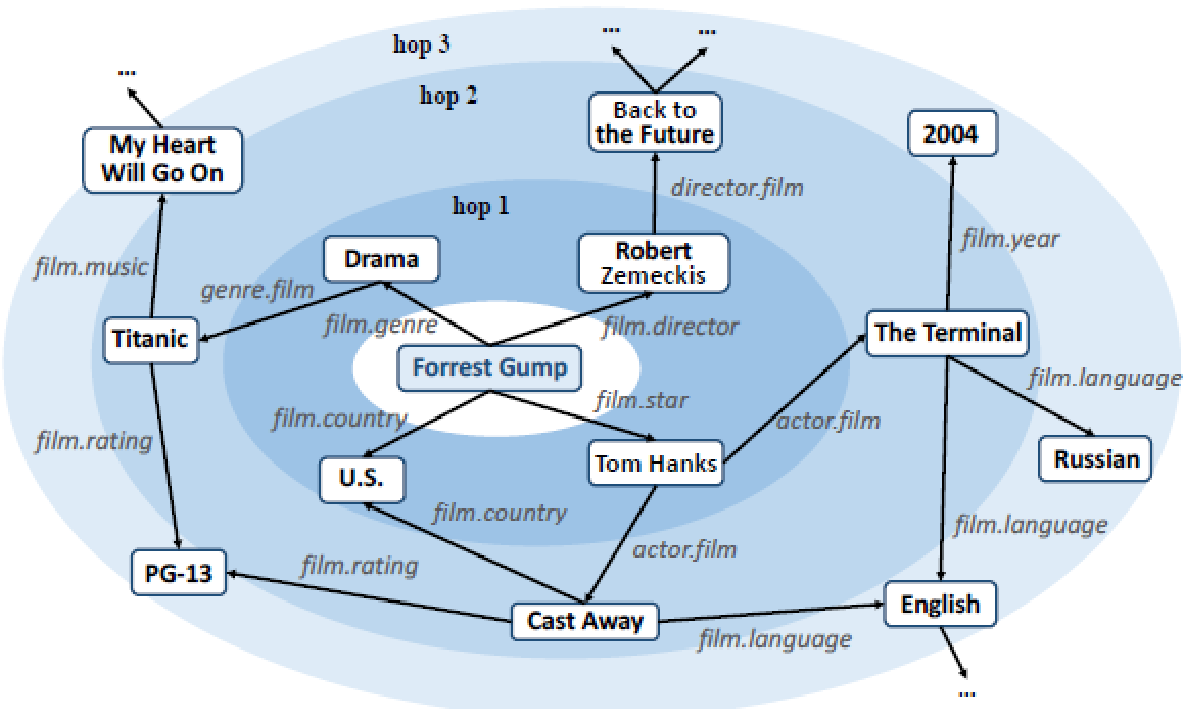
Signed Heterogeneous Information Network Embedding (SHINE)



$$
\mathcal{L} = \sum_{i \in V} \|(\mathbf{x}_i - \mathbf{x}'_i) \odot \mathbf{l}_i\|_2^2 + \lambda_1 \sum_{i \in V} \|(\mathbf{y}_i - \mathbf{y}'_i) \odot \mathbf{m}_i\|_2^2
+ \lambda_2 \sum_{i \in V} \|(\mathbf{z}_i - \mathbf{z}'_i) \odot \mathbf{n}_i\|_2^2 + \lambda_3 \sum_{s_{ij}=\pm 1} \left(f(\mathbf{e}_i, \mathbf{e}_j) - s_{ij}\right)^2
+ \lambda_4 \mathcal{L}_{reg},
$$

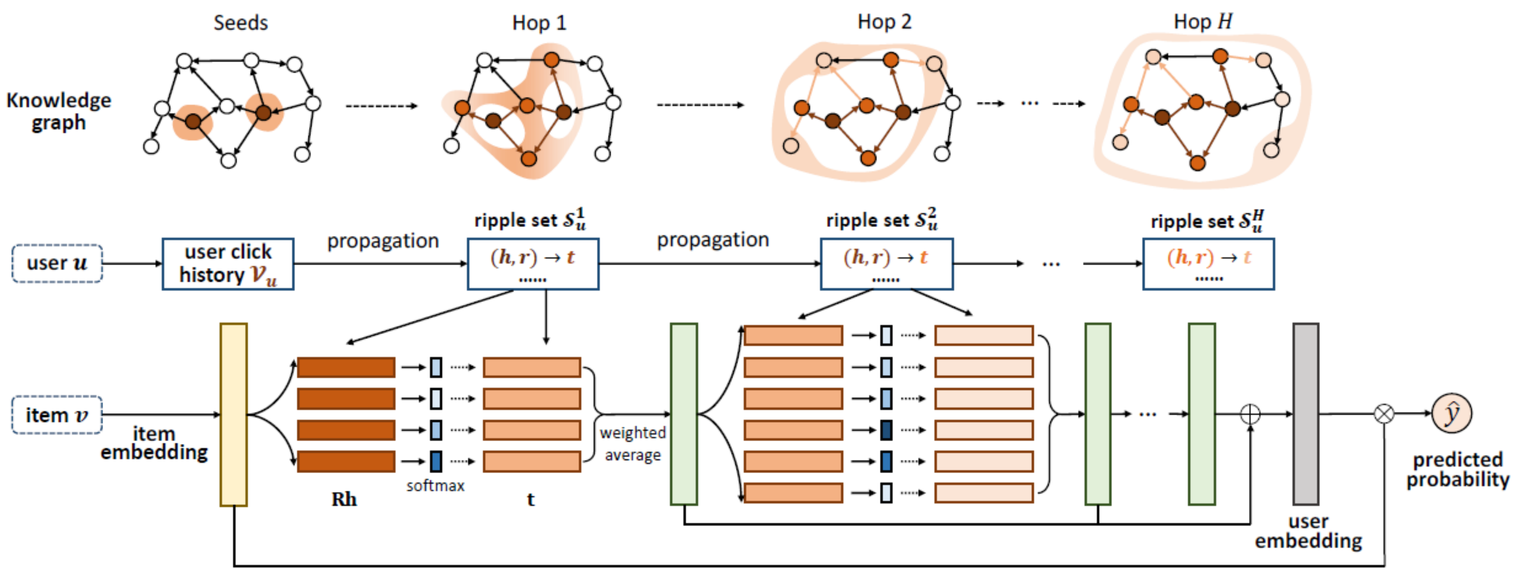
Joint Learning: RippleNet [CIKM in sub.]

Ripple Network



RippleNet

Ripple Network



$$\min \mathcal{L} = -\log\left(p(\mathrm{Y}|\Theta, \mathcal{G}) \cdot p(\mathcal{G}|\Theta) \cdot p(\Theta)\right)$$

$$= \sum_{(u,v)\in \mathrm{Y}} -\left(y_{uv} \log \sigma(\mathbf{u}^{\mathrm{T}}\mathbf{v}) + (1 - y_{uv}) \log\left(1 - \sigma(\mathbf{u}^{\mathrm{T}}\mathbf{v})\right)\right)$$

$$+ \frac{\lambda_2}{2} \sum_{r\in\mathcal{R}} \|\mathbf{I}_r - \mathbf{E}^{\mathrm{T}}\mathbf{R}\mathbf{E}\|_2^2 + \frac{\lambda_1}{2}\left(\|\mathbf{V}\|_2^2 + \|\mathbf{E}\|_2^2 + \sum_{r\in\mathcal{R}} \|\mathbf{R}\|_2^2\right)$$

Discussion

In general…

- **Efficiency: one-by-one > alternate > joint**
  - The updating frequency of KG is far less than RS…
  - All embeddings need to be learned from scratch in joint learning…

- **Performance: joint > alternate > one-by-one**
  - Joint learning methods are end-to-end…
  - Entity and relation embeddings are learned in advance in one-by-one learning, lacking the supervision from RS…

Summary

- **Recommender systems** are key technique in user-oriented web services
- **Network-structured data** are common in RS
  - User-item interaction, social network, knowledge graph
- **Network representation learning**: a popular technique of processing network-structured data
  - GraphGAN
  - Knowledge graph embedding
- RS + NRL:
  - **One-by-one learning**
  - **Alternate learning**
  - **Joint learning**

Q & A

Thanks!