Personalized Recommendations using Knowledge Graphs: A Probabilistic Logic Programming Approach

——杨子晴

Overall

- recommendation on KGs
- EntitySim : links of graph
- TypeSim : + type of entities
- GraphLF : + strengths of latent factorization with graphs
- HeteRec_p & NB(baseline)
- Dataset: Yelp; MovieLens-100K;

- Why? method cmp.; ProPPR(相关的论文的还没有来得及看...);

Preliminaries

- use binary user feedback
- regard as HIN
- from random walk to a trained walk:
  - learning a weight vector w → edge strength = f (w, ϕ_{uv})
  - optimization problem: the constraint that the PageRank computed for the positive example nodes is greater than that of the negative examples.
  - positive examples: movies that the user watched
  - negative examples: movies that the user did not watch or give an explicit negative feedback.

ProPPR

- ProPPR是受随机逻辑程序（SLP）启发的最新概率逻辑语言，它使用个性化PageRank进行有效推理。我们采用概率推理的这种观点作为随机遍历从标记逻辑程序构造的图形来研究这两种语言之间的关系。
- seedset: a set of entities that each user is interested in
- $$\begin{aligned} \texttt{seedset}(\mathrm{U},\mathrm{E}) \leftarrow & \texttt{reviewed}(\mathrm{U},\mathrm{M}), \texttt{link}(\mathrm{M},\mathrm{X}), \texttt{related}(\mathrm{X},\mathrm{E}), \\ & \texttt{isEntity}(\mathrm{E}). && (1) \\ \texttt{related}(\mathrm{X},\mathrm{X}) \leftarrow & \texttt{true}. && (2) \\ \texttt{related}(\mathrm{X},\mathrm{E}) \leftarrow & \texttt{link}(\mathrm{X},\mathrm{Z}), \texttt{related}(\mathrm{Z},\mathrm{E}). && (3) \end{aligned}$$

Figure 2: Seed Set generation

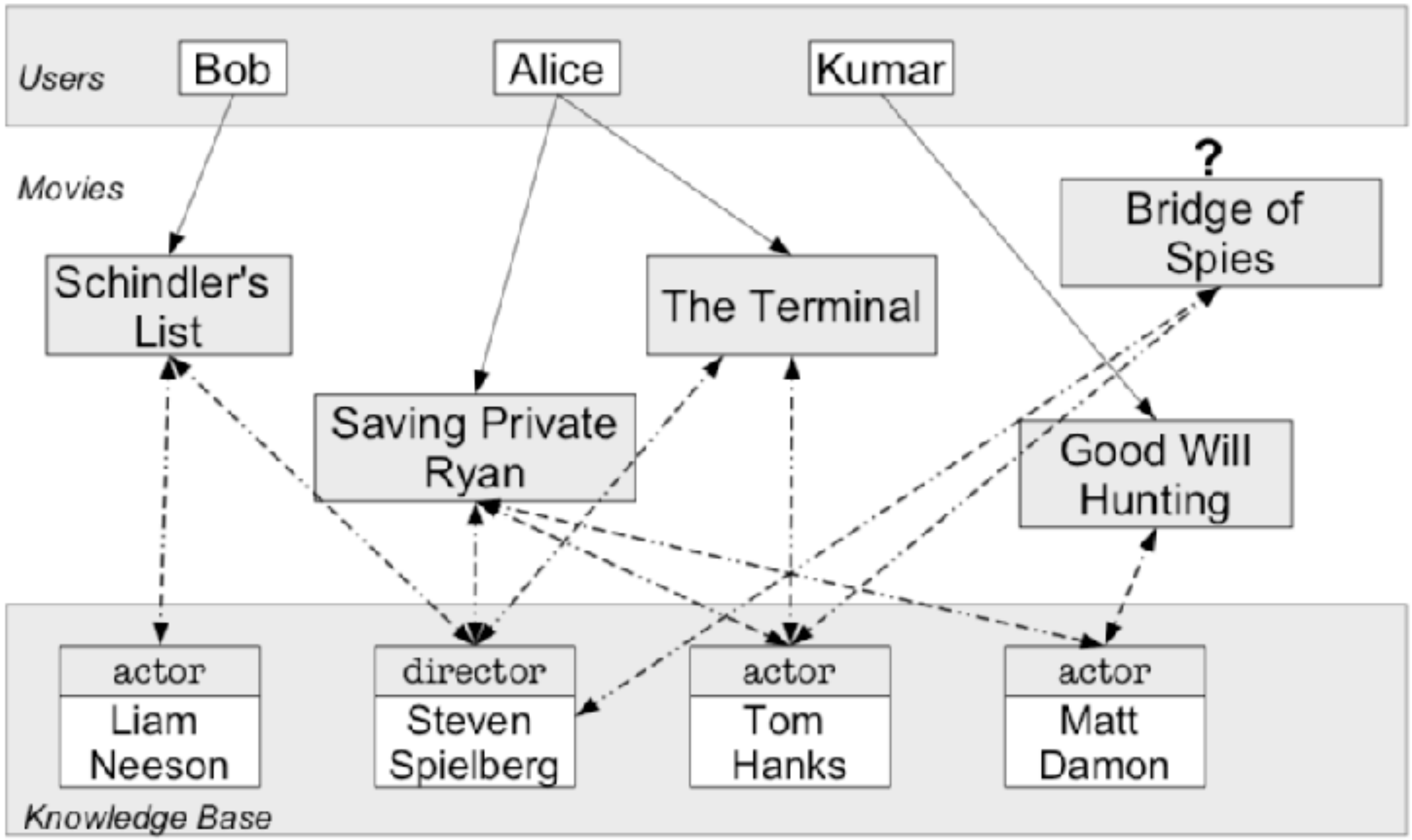


Figure 1: Example of Movie Recommendation

- infinite -> restrain the time

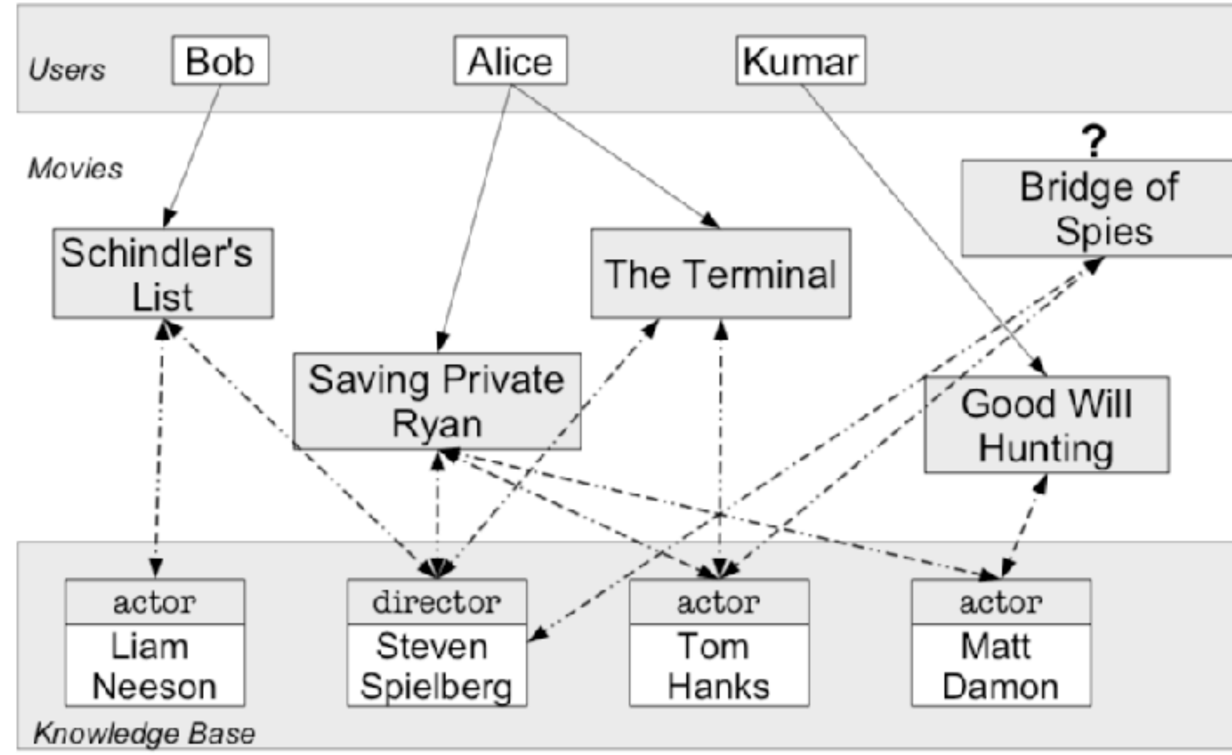
EntitySim

$$\texttt{reviewed}(\texttt{U},\texttt{M}) \leftarrow \texttt{seedset}(\texttt{U},\texttt{E}), \texttt{likesEntity}(\texttt{U},\texttt{E}),$$
$$\texttt{related}(\texttt{E},\texttt{X}), \texttt{link}(\texttt{X},\texttt{M}), \texttt{isApplicable}(\texttt{U},\texttt{M}). \quad (4)$$
$$\texttt{likesEntity}(\texttt{U},\texttt{E}) \leftarrow \{\texttt{l}(\texttt{U},\texttt{E})\}. \quad (5)$$

Figure 4: `EntitySim`**: ProPPR program for finding movies that a user may like using similarity measured using the graph links**

- the user U may like a movie M if there is an entity E belonging to U's seed set, and U likes E, and E is related to another entity X, which appears in the movie M

$$-\sum_{k=1}^{m}\left(\sum_{i=1}^{I_m}\log \mathbf{p}[u_i^{k+}] + \sum_{j=1}^{J_m}\log(1-\mathbf{p}[u_j^{k-}])\right) + \mu\|\mathbf{w}\|_2^2 \quad (6)$$



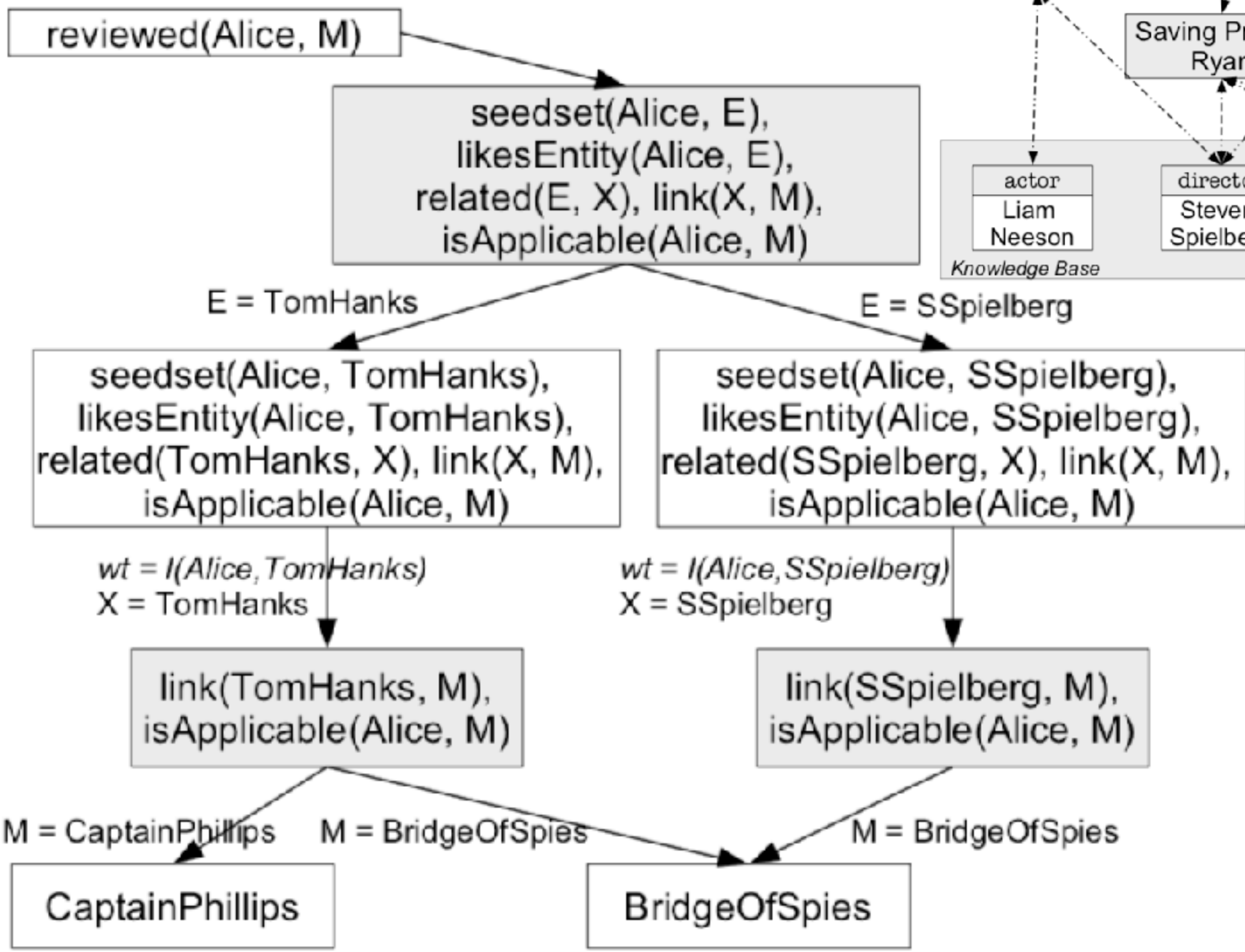


Figure 5: Sample grounding of the EntitySim ProPPR program

TypeSim

$$
\begin{aligned}
\texttt{reviewed}(\texttt{U},\texttt{R}) \leftarrow\ & \texttt{seedset}(\texttt{U},\texttt{E}), \texttt{likesEntity}(\texttt{U},\texttt{E}),\\
& \texttt{popularEntity}(\texttt{E}), \texttt{related}(\texttt{E},\texttt{X}),\\
& \texttt{link}(\texttt{X},\texttt{R}), \texttt{isApplicable}(\texttt{U},\texttt{R}). && (7)\\
\texttt{likesEntity}(\texttt{U},\texttt{E}) \leftarrow\ & \{\texttt{l}(\texttt{U},\texttt{E})\}. && (8)\\
\texttt{popularEntity}(\texttt{E}) \leftarrow\ & \texttt{entityOfType}(\texttt{E},\texttt{T}),\\
& \texttt{popularType}(\texttt{T})\{\texttt{p}(\texttt{E})\}. && (9)\\
\texttt{popularType}(\texttt{T}) \leftarrow\ & \{\texttt{p}(\texttt{T})\}. && (10)\\
\texttt{typeAssoc}(\texttt{X},\texttt{Z}) \leftarrow\ & \texttt{entityOfType}(\texttt{X},\texttt{S}), \texttt{entityOfType}(\texttt{Z},\texttt{T}),\\
& \texttt{typeSim}(\texttt{S},\texttt{T}). && (11)\\
\texttt{typeSim}(\texttt{S},\texttt{T}) \leftarrow\ & \{\texttt{t}(\texttt{S},\texttt{T})\}. && (12)
\end{aligned}
$$

Figure 6: TypeSim method for recommendations

GraphLF

- they develop a general representation of users and items based on the ratings data that are more generalizable and often indiscernible in the raw data.
- $$\texttt{reviewed}(\mathrm{U},\mathrm{R}) \leftarrow \texttt{related}(\mathrm{U},\mathrm{E}), \texttt{related}(\mathrm{E},\mathrm{X}), \texttt{link}(\mathrm{X},\mathrm{R}), \texttt{isApplicable}(\mathrm{U},\mathrm{R}). \tag{13}$$

$$\texttt{related}(\mathrm{U},\mathrm{E}) \leftarrow \texttt{seedset}(\mathrm{U},\mathrm{E}), \texttt{simLF}(\mathrm{U},\mathrm{E}). \tag{14}$$

$$\texttt{related}(\mathrm{X},\mathrm{X}) \leftarrow . \tag{15}$$

$$\texttt{related}(\mathrm{X},\mathrm{Y}) \leftarrow \texttt{link}(\mathrm{X},\mathrm{Z}), \texttt{simLF}(\mathrm{X},\mathrm{Z}), \texttt{related}(\mathrm{Z},\mathrm{Y}). \tag{16}$$

$$\texttt{simLF}(\mathrm{X},\mathrm{Y}) \leftarrow \texttt{isDim}(\mathrm{D}), \texttt{val}(\mathrm{X},\mathrm{D}), \texttt{val}(\mathrm{Y},\mathrm{D}). \tag{17}$$

$$\texttt{val}(\mathrm{X},\mathrm{D}) \leftarrow \{\mathrm{v}(\mathrm{X},\mathrm{D})\}. \tag{18}$$

Figure 7: GraphLF method for recommendations

- `EntitySim` - $\mathcal{O}(n)$: In this method, we learn one parameter per user-entity pair. However, by virtue of the rules, we constrain the entities to be chosen from the seedset of that user, which is of a constant size c.
- `TypeSim` - $\mathcal{O}(n + e + t^2)$: In addition to those parameters learned for `EntitySim`, it also learns $e + t$ weights for each of the entities and types. Moreover, it also learns the type association between pairs of types leading to an additional t^2 parameters.
- `GraphLF` - $\mathcal{O}(n + m + e)$: For each of the users, entities and items, we learn a constant d number of weights corresponding to the latent dimensions.

Method	P@1	P@5	P@10	MRR	Settings
`HeteRec_p`	0.0213	0.0171	0.0150	0.0513	*published results*
`EntitySim`	0.0221	0.0145	0.0216	0.0641	$n = 20$
`TypeSim`	0.0444	**0.0188** [$\uparrow$ 10%]	**0.0415** [$\uparrow$ 176%]	**0.0973** [$\uparrow$ 89%]	$n = 20$
`GraphLF`	**0.0482** [$\uparrow$ 126%]	0.0186	0.0407	0.0966	$n = 20$, $dim = 10$
`NB`	0	0.0012	0.0013	0.0087	

Table 2: Performance comparison on `Yelp`: The best score for each metric is highlighted in blue and the lowest score in red. [$\uparrow x\%$] gives the percent increase compared to the corresponding `HeteRec_p` score

Method	P@1	P@5	P@10	MRR	Settings
`HeteRec_p` (on `IM100K-UIUC`)	0.2121	**0.1932**	0.1681	**0.553**	*published results*
`EntitySim`	0.3485	0.1206	**0.2124** [$\uparrow$ 26.3%]	0.501 [$\downarrow -9.4\%$]	$n = 10$
`TypeSim`	**0.353** [$\uparrow$ 66.4%]	0.1207 [$\downarrow -37.5\%$]	0.2092	0.5053	$n = 10$
`GraphLF`	0.3248	0.1207 [$\downarrow -37.5\%$]	0.1999	0.4852	$n = 10$, $dim = 10$
`NB`	0.312	0.1202	0.1342	0.4069	

Table 3: Performance comparison on `IM100K` (`IM100K-UIUC` & `IM100K*`): The best score for each metric is highlighted in blue and the lowest score in red. [$\uparrow x\%$] gives the percent increase compared to the corresponding `HeteRec_p` score and [$\downarrow x\%$], the percent decrease.

Density of a dataset as $\frac{\#\texttt{reviews}}{\#\texttt{users} \times \#\texttt{items}}$

Conclusion and Question

- learned meta-path
- good at cold-start —> the redundant of type info.
- methods of combine the information

- 都是metapath的算法…为什么是16年的论文啊？是现在方法更新太快了，还是这篇论文有什么独特之处么？
- 感觉手动设计的部分也很多…

Thanks