

ページランクと埋め込みの組み合わせ

TransEなどの知識グラフ埋め込み手法とページランクを組み合わせた推薦手法を考えている。説明性に関してあまり考えられていない。一応いくつかの説明性アルゴリズムが適用可能であることはわかっている。

1. 背景

RecWalkと呼ばれる手法を拡張。推薦アルゴリズムの現状の課題に直接取り組んでいるわけではないので、この辺りの背景を詰めたいところ。

1-1. RecWalkに関して

WSDM2019の研究。ページランクの拡張。下の表のように単純なアルゴリズムながらDNNベースのモデルより高性能。

Table 2: Top-n Recommendation Quality of the Competing Approaches in Terms of HR, ARHR and NDCG

Method	movielens			yahoo			pinterest		
	HR [%]	ARHR [%]	NDCG [%]	HR [%]	ARHR [%]	NDCG [%]	HR [%]	ARHR [%]	NDCG [%]
EigenRec	45.21	20.44	26.35	48.12	23.30	29.23	33.81	13.51	18.41
PureSVD	44.14	19.33	25.36	38.68	18.30	22.62	30.97	11.85	16.30
RP3b	34.87	15.02	19.66	41.51	17.82	22.94	27.01	8.07	12.45
SLIM	46.34	21.39	27.28	52.44	26.15	32.35	34.17	13.63	18.57
Mult-DAE	44.06	18.97	24.83	45.37	21.46	27.07	35.03	13.79	18.77
Mult-VAE	44.35	19.50	25.31	45.09	21.22	26.80	35.13	13.73	18.71
RecWalk ^{K-step}	50.28	27.20	33.13	55.02	28.94	35.10	35.38	14.07	18.95
RecWalk ^{PR}	52.47	27.29	34.09	54.92	28.74	34.91	35.29	14.07	19.00

ユーザ数 U 、アイテム数 I でユーザ-アイテムの評価値行列のみが与えられている状況を考える。ユーザとアイテムをノードとした有向グラフを考える。ただしユーザ→アイテムへのリンクとアイテム→ユーザへのリンク両方が張られているとする。次の確率遷移行列にしたがって各ユーザノードに対するPersonalized Page Rank(PPR)を実行。

$$P = \alpha H + (1 - \alpha)M$$

ここで $H \in \mathbb{R}^{U+I \times U+I}$ は有向グラフの隣接行列を正規化したもの、つまりグラフの確率遷移行列。 $M \in \mathbb{R}^{U+I \times U+I}$ は $M_{ij} = 0 (i \leq U, j \leq I)$ で、 $M_{ij} (i > U, j > U)$ がアイテムどうしの(正規化された)類似度になっている行列である。アイテムどうしの類似度は任意の手法で推定する。RecWalk論文ではSLIMという手法を用いていた。RecWalkはグラフの確率遷移行列をノードどうしの類似度で補正してPPRを動かしていると解釈できる。

上のMを計算するためのアイテムどうしの類似度を推定するモデルであるSLIMは次のような最適化をする。

$$\argmin_W \frac{1}{2} \|A - AW\|_F^2 + \frac{\beta}{2} \|W\|_F^2 + \lambda \|W\|_1$$

subject to $W \geq 0$, $\text{diag}(W) = 0$ $A \in \mathbb{R}^{U \times I}$ は評価値行列、 $W \in \mathbb{R}^{I \times I}$ はアイテムどうしの類似度と解釈できる重みである。

PPRは次のようなもの。 $\pi_u = \lim_{K \rightarrow \infty} \text{e}_u^{\top} (\mu P + (1-\mu) \text{e}_u \text{e}_u^{\top})^K$

e_u はターゲットユーザのインデックスに1それ以外は0の行列。 π_u はターゲットユーザに対するノードの重要度がランキングされているベクトル。推薦の予測ではこのベクトルにもとづいて推薦アイテムをランキングする。

2. 手法

RecWalkではユーザ、アイテムのみが与えられたグラフでアイテムのみの類似度を使っていた。ユーザ、アイテム以外の補助情報を考慮したHeterogeneousなグラフで知識グラフの埋め込みを使ってノードどうしの類似度を考慮してみる。

与えられたユーザ、アイテム、ユーザ・アイテムのそれぞれの属性情報をノード数 N のグラフで表現し、その正規化された隣接行列を $H \in \mathbb{R}^{N \times N}$ とする。知識グラフ埋め込みによる各ノードの埋め込み行列どうしの内積を格納して正規化した行列を $M \in \mathbb{R}^{N \times N}$ とする。次の確率遷移行列でPPRを動かす。

$$P = \alpha H + (1 - \alpha) M$$

2-1. 知識グラフの埋め込み

今回はTransEを使用。知識グラフ埋め込み手法は何を使ってもOK。

TransEは次の目的関数を最大化する。 $L = \sum_{(h, r, t) \in S} \sum_{(h_{neg}, r, t_{neg}) \in S_{neg}} \gamma + d(\mathbf{h} + \mathbf{r}, \mathbf{t}) - d(\mathbf{h}_{neg} + \mathbf{r}, \mathbf{t}_{neg})$ またTransEにノードどうしの内積が小さくなるようなL2正則化項を加えたモデルも実験してみた。正則化項を加えるのは、上のRecWalkの場合のSLIMはアイテムの類似度行列が W スパースになっているので、埋め込みベクトルの内積が小さいorスパースになる方がいいのではないかと考えたため。(本当はL1正則化も実験したいが力不足で保留中)

実験

使ったデータ

小さめ&ある程度密なデータで実験してみた。Amazon Dataset Luxury 5coreを使用。グラフを構成する際のTriplet(head, relation, entityの三つ組)の種類は

- user ->(buy) item (34278)
- item ->(belong to) brand (1629)
- item ->(also_view) item
- item ->(also_buy) item

今回実験したデータは、

- ユーザ数: 3819
- アイテム数: 1581
- ユーザ・アイテムのペア: 34278
- ブランド数: 8

アイテムの属性として考えられるのが「ブランド」しかなく、その数も少ないのが問題か。

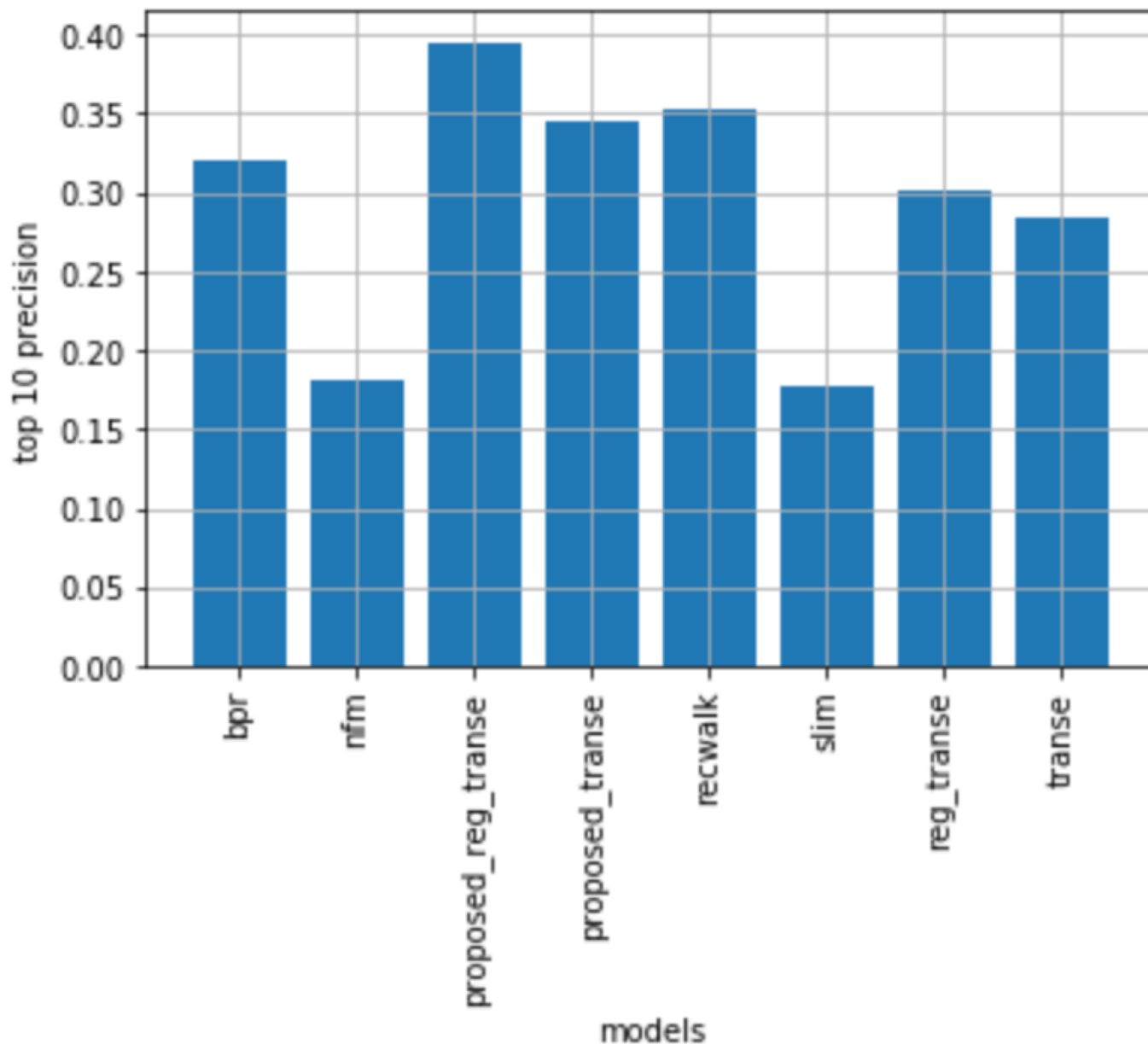
使った手法

- Bayesian Personalized Ranking(BPR)
- Neural Factorization Machine (NFM)
- TransE
- regTransE(正則化つきTransE)
- SLIM
- RecWalk

- Proposed(TransE)
- Proposed(regTransE)

結果

Top 10 Precisionで評価。下のグラフのように提案手法のスコアが一番高い。（厳密に交差検証でパラメータチューニング→テストをしていないのでちょっと怪しい）



本当はやりたいこと

page rank awareな埋め込み。次の目的関数の和を最小化。（ここではTransEを例に）

埋め込み誤差 $L_{\text{embed}} = \sum_{(h, r, t) \in S} \sum_{(h_{\text{neg}}, r, t_{\text{neg}}) \in S_{\text{neg}}} \gamma + d(\text{vec}(h) + \text{vec}(r), \text{vec}(t)) - d(\text{vec}(h_{\text{neg}}) + \text{vec}(r), \text{vec}(t_{\text{neg}}))$

ページランク誤差 $L_{\text{ppr}} = \sum_{u \in \text{User}} \pi_u - \{\mu P \pi_u + (1 - \mu) \text{vec}(e)_u\}$

$P = \alpha H + (1 - \alpha)M$ π_u はあらかじめ計算しておいた各ユーザに対するPersonalized page rank。 π_u は $\text{vec}(x) = \mu P \text{vec}(x) + (1 - \mu) \text{vec}(e)_u$

の解である。埋め込みベクトルによって構成した確率遷移行列が

$\pi_u = \mu P \pi_u + (1 - \mu) \text{e}_u$ を満たすように最適化することによって、ユーザに対するアイテムのランキングそのものを再現できるような埋め込みを学習したいという気持ち。

予測は埋め込みベクトルで構成した P での PPR で行う。

現状とToDoリスト

- 大きい & スパースなデータで実験
- 他の埋め込み手法で実験
- 埋め込み手法のL1正則化
- page rank awareな埋め込み手法を実装、実験

説明性に関して

このプロジェクトでは説明性に関しては取り組んでいない。上で実験した手法に関してはPRINCEと呼ばれる手法や、[影響関数を行列分解に応用したもの](#)を適用することができると考えられる。

Reference