

Recommenders

L2. Non-personalized

- Product association also aims for diversity. (People who buy X, buy Y)

- How to rank items:
 - "stars" rating: it is important to let the user know what each score means. use as less as possible ^{options}
 - like/dislike _{disappearing}
 - viewed, interacted...
 - confidence (knowing an item is good even without interactions)
 - tolerance: risk/reward balance.
 - number of reviews.
 - number of purchases/bookings
 - recency

- Instead of mean:

$$\mu_j = \frac{\sum_{i \in I} r_i + K(\mu)}{n + K}$$

Damped means

μ dataset mean.
K control parameter

↓ everything is mean by default (to avoid low confidence rankings with little feedback)

- How to score new stories:

↓ consider time in rankings

$$\frac{(U-D-1)^{\alpha}}{(\underbrace{t_{\text{now}} - t_{\text{past}}}_{\text{recency}})^{\beta}} \cdot P$$

P penalty

- product association:

$$\text{score}(Y|X) = \frac{X \text{ and } Y}{X}$$

↓ always recommends most popular items.

to avoid that effect:

$$\text{score}(Y|X) = \frac{\frac{X \text{ and } Y}{X}}{\frac{!X \text{ and } Y}{!X}}$$

score v.1
divided by how much you buy Y without the conditional X (big for popular Ys)

↓ it still fails, as very rare products will be enhanced.

Apriori method:

confidence who buys sugar, also buys milk in 70% cases ← "score"
support: this happens in 13.5% of all purchases ← "filter"

1) Find items with minimum support: $\forall x \in \mathcal{P}(\text{items})$, $\frac{\# \text{ purchases } / x \in \text{ purchase}}{\# \text{ purchases}}$
 and itemsets.

↓
 items = {1, 2, 3}
 $\mathcal{P}(\text{items}) = \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$

- to optimize this, we can reduce the space: If {1} has support 25% → {1, 2}, {1, 3} and {1, 2, 3} will have 25% support at max.

2) Generate rules... (check notebook)

Lecture 3. Collaborative filtering

↳ Recommendations from 'similar' users.

$R \rightarrow$ matrix rating: sparse $M \times N$ matrix
 ↓ user ↓ item

- problems: cold start, sparsity, first user rating on item, popularity bias.

- approaches: 1) memory based (neighborhood): user-based and item-based. → in terms of which similarities are considered.
 2) model based: factorization methods, NNs, ...

• User-based:
 - similarities: computable by euclidean distance, pearson correlation (corrected), spearman correlation, cosine distance
 ↓
 crucial
 ↓
 then predict
 (can be amplified)
 (bad) between rating vectors of 2 users. (more preferable: ranking focus; bias adj.) (uncertainty data)

• Factorization Models: latent vectors, embeddings.

Lecture 4. Matrix Factorization

• When testing → visualize recommendations. Sometimes a number (metric) is not enough to understand what we are doing.

• Factorization Machines:

- Linear models: (after one-hot encoding) weights the strength of each feature/variable.

- Polynomial: represents interactions between features

$\langle w_i, w_j \rangle$

→ learns affinity $\frac{n}{n \times n}$

FM → learns embedding $n \times k$
 ↓
 embedding size

↓ vectorize to reduce params.

don't forget to normalize the representations.

Lecture 5. Ranking

- Recall > Accuracy for evaluation.
- Pointwise vs. pairwise: with pointwise, some square errors may come from different rankings.
 (x, y) if $x > y$.

attention: $\begin{bmatrix} A & C \\ B & D \end{bmatrix}$
 $+1 \quad +1 \rightarrow$ some weight when first pair is irrelevant.
 with scores $A=1, B=2, C=4, D=5$

listwise
 cross-entropy

- ranknet and cross-entropy loss for pair.
- BPR pairwise approach

Lecture 6. Content-based

Good for news/music. No cold-start problem.

User independent. Over-specialization.

Pondora: counts with features extracted by an expert.

- similar items \Leftrightarrow similar representations.

- NLP: $TF \times IDF$
 $\underbrace{\quad}_{\text{term freq.}} \times \underbrace{\quad}_{\text{inverse doc freq.}}$

Lecture 7. Graph Convolutional Networks.

- Graph data is irregular \rightarrow GCN. (adjacency matrix for graph topology). sparse; 0-diagonal. $[\hat{A} = A + Id]$
- Graph Attention Network: learn the model giving priority to a particular part of the data. (GAT)
 particular zone of the graph:
- GCN: convolution of GCN $\equiv H^{(l)}$
 $\hookrightarrow H^{(0)} = X$ (feature input)

\hookrightarrow analogy to images: aggregate info of neighbours to my own info

- do not split data (you would destroy A)
 \hookrightarrow select nodes through binary masks. } one graph // three sets.

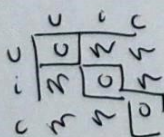
new embedding

good: embeddings contain info of all embeddings \rightarrow entities are not trained 'clone'

\hookrightarrow RSS.

topology

\hookrightarrow implemented through message passing:



\hookrightarrow include in FM

Lecture 8. GCN for RS.

1 layer \rightarrow no act function (σ)

$g(x)$ = embeddings.

- embeddings are better *even without training* just because of the influence of \hat{A} in the formula.

- GAT vs. GCN: change normalization for attention constant.
weight the average lig

(Fm 'only' catch 2-order interactions) \Rightarrow solution: graphs?

\downarrow
use GCN to capture the embeddings. implicitly capture higher order interactions.

\Rightarrow extend to context. (message passing: $j \rightarrow i$)

\downarrow
structure the interactions in

users	items
0	1
1	0

... \Rightarrow extendable to multiple context.

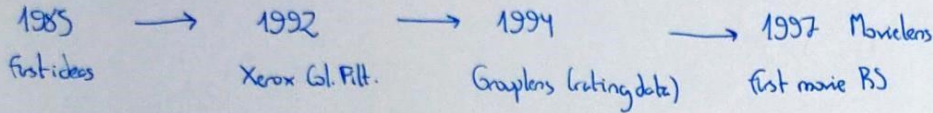
\downarrow
adjacency matrix.

$$\langle v_i, v_j \rangle \rightarrow \langle g(x_i), g(x_j) \rangle$$

L1. Introduction

- "People don't know what they want until you show them" \rightarrow RS
too many opt.

- focus on head or long tail (niche?)



1st Gen: Knowledge/content-based, collaborative filtering, hybrid

2nd Gen: MF, web usage mining, personality.

3rd Gen: DL; collab. filtering or content based

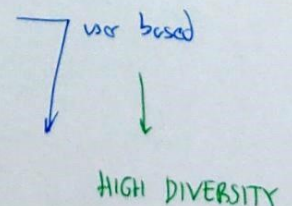
- Types of data: explicit/implicit
- How do they work: prediction or ranking problem
- Types of RS:
 - 1) Non-personalized
 - 2) Content-based
 - 3) Collaborative-based
 - 4) Knowledge-based
- Evaluation: relevance, novelty, serendipity, diversity...
Accuracy, RMSE, MAP, NCGD, sales, return rates, customer loyalty.
 \downarrow
online (A/B), offline...

L2. Non-personalized

- Some recommendations for everyone. based on
 - aggregated opinions (rankings)
 - basic product association (customers who bought also bought)
- Difficulties with ratings: reliable? shows popularity? preference change? transparent?
- A priori method: the more rules you define, the larger risk they are produced by chance. lift-of-a rule ($X \rightarrow Y$) = $\frac{\text{sup}(X \text{ and } Y)}{\text{sup}(X) \text{sup}(Y)}$

L3. Collaborative filtering

- Significance weighting: Few items in common \rightarrow reduce influence
- Predictions: $\frac{\sum \text{sim top users} \times r_{\text{user}}}{\sum \text{sim top users}}$ \rightarrow correct scale with $\bar{r}_u + \frac{\dots (r_{\text{user}} - \bar{r}_v)}{\dots}$
do not use weak users
negative correlations



- Impact of long tail!!

- Item based \rightarrow more stable similarities

generally $\#users > \#items \rightarrow$ item-based more efficient.

- Dimensionality reduction: quality, efficiency
in terms of latent factors (PCA / SVD)
- Explain recommendations (why is recommended) \rightarrow transparency + trust

L4 Factorization models. \hookrightarrow latent factor models \equiv opt problem

$$\hat{r}_{ui} = P_u Q_i^T \quad \Rightarrow \text{embeddings (each factor characterizes something)}$$

$\downarrow \quad \downarrow$
 user/item latent vector

- problem: hard to optimize with huge amount of missing data. (SVD)

\hookrightarrow fill with 0/ avg, recursive methods \rightarrow inaccurate

modeling directly the observed data only!

- Vanilla MF SVD Funk
- Regularized Vanilla MF
- Reg MF + bias
- SVD++ (has implicit data)
- Non-neg. MF
- SLM (SLIM)

model-based

- ⊕ space efficient
- ⊕ speed
- ⊕ less overfitting

L5. MF Hybrid Models

- MF with side features / temporal features
- FM: Biased MF \equiv FM with user/item only
has implicit info and like SVD++

L6. Learning 2 Rank

\hookrightarrow do not use accuracy (RMSE, MAE) \Rightarrow Precision@K, MAP, NDCG, recall@K = $\frac{\text{relevant items in top K}}{\text{total liked items}}$ \rightarrow by user

\uparrow precision in position
 \downarrow For items with diff. relevancy

Proc: RankNet, LambdaRank, BPR

List: Lambda Mart

L7. GCN: GNN nodes can have graph-like relations.

- Left normalization: $\hat{D}^{-1}A$
- Symm-norm: $\hat{D}^{-\frac{1}{2}}A\hat{D}^{-\frac{1}{2}}$