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 options as possible
  - like / dislike
    disappearing
  - viewed, interacted...
  - confidence (knowing an item is good even without iteractions)
  - tolerance: risk / reward balance.
  - number of reviews.
  - number of purchases / bookings
  - recency

- Instead of mean:

  Damped means

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

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

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

P → penalty

- product association:

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

↳ always recommends most popular items.

to avoid that effect:

$$\text{score}(Y|X) = \frac{\left.\dfrac{X \text{ and } Y}{X}\right\}\text{score v.1}}{\left.\dfrac{!X \text{ and } Y}{!X}\right\}}$$

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: $\forall x \in P(\text{items})$, $\frac{\#\text{purchases} / x \in \text{purchase}}{\#\text{purchases}}$

and itemsets.

↓

items = {1, 2, 3}

P(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 → matrix rating: sparse M×N matrix ($M \to$ user, $N \to$ 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 (bad), pearson correlation (corrected) (more preferable: ranking focus; bias adj.), spearman correlation, cosine distance (unary/binary data)

can be amplified

↳ between rating vectors of 2 users.

↓ crucial ↓ then predict

• 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:

- Lineal models: (after one-hot encoding) weighs the strength of each feature/variable.
- Polinomial: represents interactions between features $\langle w_i, w_j \rangle$ → learns affinity $n \times n$

↳ vectorize to reduce params.

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

don't forget to normalize the representations.

Lecture 5. Ranking

- Recall > Accuracy for evaluation.

- Pointwise (RMSE) vs. pairwise (AUC): with pointwise, some square errors may come from different rankings. — listwise: cross-entropy

  pairwise: (x,y) if $x>y$.

  attention: $\boxed{\begin{matrix}A\\B\end{matrix}}$ $\boxed{\begin{matrix}C\\D\end{matrix}}$
  +1 +1 → same weight when first pair is irrelevant.

  with scores $A=1$, $B=2$, $C=4$, $D=5$

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

Lecture 6. Content-based

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

User independant. Over-specialization.

Pandora: counts with features extracted by an expert.

- similar items ⇔ similar representations.
- NLP: $\underbrace{TF}_{\text{term freq.}} \times \underbrace{IDF}_{\text{inverse doc freq.}}$

Lecture 7. Graph Convolutional Networks.

- Graph data is irregular → 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 ≡ $H^{(l)}$
  ↳ $H^{(0)} = X$ (feature input)

  ↳ analogy to images: aggregate info of neighbours to my own info

  - do not split data (you would destroy A)
    ↳ select nodes through binary masks.

  } one graph // three sets.

  } new embeddings

  good: embeddings contain info of all embeddings → entities are not trained 'alone' — topology
  ↳ RRSS.

  ↓ implemented through message passing:

$$\begin{array}{c|ccc} & U & i & c \\ \hline U & \boxed{0} & \sim & \sim \\ i & \sim & \boxed{0} & \sim \\ c & \sim & \sim & \boxed{0} \end{array}$$

include in FM

Lecture 8. GCN for RS.

1 layer → 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 (ig)

(Fm 'only' catch 2-order interactions) ⇒ solution: graphs?
↓
use GCN to capture the embeddings. implicitly capture higher order interactions.

⇒ extend to context. (message passing: $\mu_{j \to i}$)
↓
structure the interactions in

$$\begin{array}{cc|c} \text{users} & \text{items} & \\ \hline 0 & & \text{users} \\ & 0 & \text{items} \end{array}$$

↓
adjacency matrix.

... ⇒ extendable to multiple context.

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

L1. Introduction

- "People don't know what they want until you show them" $\xrightarrow[\text{too many opt.}]{}$ RS

  - focus on head or long tail (niche?)

1985 → 1992 → 1994 → 1997 Movielens

first ideas — Xerox Col. Filt. — Grouplens (rating data) — first movie RS

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.
  ↓
  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 $(X \to Y) = \frac{sup(X \text{ and } Y)}{sup(X)\,sup(Y)}$

L3. Collaborative filtering

- Significance weighing: Few items in common → reduce influence
- Predictions: $\frac{\sum \text{sim top users} \times r_{users}}{\sum \text{sim top users}}$ ↳ correct scale with $\overline{r_u} + \frac{\cdots (r_{users} - \overline{r_v})}{\cdots}$
  ↓
  do not use weak users
  negative correlations

user based

↓

HIGH DIVERSITY

- Impact of long tail!!

• Item based → more stable similarities

generally #users > # items → item-based more efficient.

• Dimensionality reduction: quality, efficiency

in terms of latent factors (PCA / SVD)

• Explain recommendations (why is recommended) → transparency + trust

L4 Factorization models.

latent factor models = opt problem

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

user/item latent vector

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

fill with 0/ avg, recursive methods → 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 == FM with user/item only

has implicit info ~~and~~ like SVD++

L6. Learning 2 Rank

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

Precision@K = $\frac{\text{relevant items in top K}}{K}$

MAP ← precision in position

NDCG → for items with diff. relevancy

Pair: RankNet, LambdaRank, BPR

List: Lambda Mart

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

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