RS Basics and Classic Algorithms

密言

2025 年 5 月 30 日

Table of Content i

Table of Content ii

What is RS?

- 推荐问题的建模
  - 基于“信息过载”的问题，联系用户和信息，帮助用户获取有价值的信息，将信息展现给对应感兴趣的用户。
  - 本质：“评分预测”问题。预测评分 $\rightarrow$ Top-K 推荐
- 推荐方法分类
  - Traditional RS
    - Collaborative Filtering (User-based, Item-based, Model-based, Hybrid) / (explicit, implicit(e.g. browsing history))
    - Content-based Recommendation
  - Deep Learning based RS
- RS 面临的问题
  - 数据稀疏问题
  - Cold start 问题
  - 安全性问题
  - Others: 隐私问题、可解释问题、etc.

RS Basics: RS 的技术流程

1. 数据收集与存储 (Data Collection & Storage)
   - User and Item data
   - Interaction data
     - Explicit: rating, Review, like or unlike
     - Implicit: click, View
   - Contextual Data
2. 特征工程 (Feature Engineering)
   - 将原始数据转换为能够被推荐模型有效利用的特征（Features)。决定推荐系统性能上限的关键步骤。
3. 候选集生成 (Candidate Generation / Retrieval)
   - "召回"，筛选出与用户相关的候选物品集合
4. Ranking / Scoring
   - 预测用户对物品的打分，排序并推荐
5. 后处理 (Post-processing)
   - 多样性，解释性问题
6. Evaluation

Traditional RS: Collaborative Filtering

User-based CF

- 基本思想：相似偏好的用户在物品的打分上是相似的（“用户会喜欢相似用户所喜欢的物品）
- Method:
    - 确定目标用户
    - 根据相似度确定与目标用户最相似的 N 个用户
    - 根据相似用户预测评分
    - 根据评分对用户进行物品推荐

	item 1	item 2	item 3	item 4	item 5
user 1	5	2	1	3	?
user 2	4	1	1	2	5
user 3	1	3	4	5	2
user 4	3	2	1	2	3

Traditional RS: Collaborative Filtering

相似度的计算方式:

- Jaccard 相似度：集合交并比
- 余弦相似度：$Sim(u_1, u_2) = \frac{\vec{u_1} \cdot \vec{u_2}}{|\vec{u_1}| \cdot |\vec{u_2}|}$
- Pearson 相关系数: 减去打分均值后，计算余弦相似度（归一化的余弦相似度）
- 欧式距离：适合度量包含相似度和强度（e.g. 访问频率）的场景

评分预测方式:

- 加权预测：$R_{u_1, item_5} = \frac{\sum_{i=1}^{N} \omega_i R_{i, item_5}}{\sum_{i=1}^{N} \omega_i}$，其中$\omega$是相似度
- 基于用户评分均值增量：$R_{u_1, item_5} = \overline{R_{u_1}} + \frac{\sum_{i=1}^{N} \omega_i (R_{i, item_5} - \overline{R_i})}{\sum_{i=1}^{N} \omega_i}$，能够缓解不同用户不同打分习惯带来的影响

Traditional RS: Collaborative Filtering

Item-based CF

- 思想：用户可能喜欢与曾经喜欢物品相似的物品
- Method:
  - 确定需要预测的目标物品
  - 根据相似度确定与目标物品相似的 N 个物品
  - 根据相似物品预测评分
  - 根据预测评分推荐相关物品
- 评分计算：
  - 基于物品均分增量：$R_{u_1,item_5} = \overline{R_{item_5}} + \frac{\sum_{j=1}^{N} \omega_j (R_{u_1,j} - \overline{R_j})}{\sum_{j=1}^{N} \omega_j}$

	item 1	item 2	item 3	item 4	item 5
user 1	5	2	1	3	?
user 2	4	1	1	2	5
user 3	1	3	4	5	2
user 4	3	2	1	2	3

User-based vs Item-based

- 计算复杂性：对于用户数 » 物品数且物品更新频率低，计算物品间相似度计算量低且不必实时更新，Item-based 更优；对于 item 时效性高且数量多时，User-based 更优。
- 场景：非社交网络中，Item 间的联系是很重要的推荐原则，选用 Item-based；在社交网络中，User 间的联系是重要信息，选用 User-based
- 推荐多样性：
  - 从单个用户看，Item-based 推荐用户历史相似物品，多样性较差
  - 从系统层面看，Item-based 推荐覆盖率远远优于 User-based，User-based 倾向于推荐热门物品，Item-based 对长尾物品推荐表现好（只要某用户同时购买两个冷门物品，即可建立起联系）
- 用户对推荐算法的适应程度：
  - 有共同喜好的用户：User-based 效果好
  - 喜爱物品自相似：Item-based 效果好

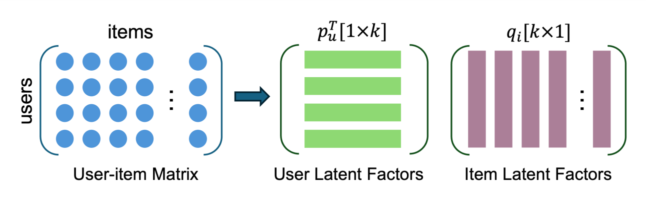
Model-based CF and Hybrid-based CF

- Model-basd CF
  - **先用历史数据训练得到一个模型，再用此模型进行预测**
  - e.g. Latent Factor Model (LFM)
- Hybrid-based CF
  - **将多种推荐技术进行混合以此弥补相互间的缺点**
  - Hybrid Method:
    - Weighted
    - Switch
    - **特征组合** (Feature Combination)
    - **级联型** (Cascade)
    - **特征递增** (Feature Augmentation)
    - **元层次混合** (Meta-level hybrid)

Matrix Factorization: Key Technique for Model-based CF

Funk-SVD (Latent Factor Model)

- 对于每个评分，有 $R_{u,i} = p_u^\top \cdot q_i = \sum_{k=1}^{K} p_{u,k}^\top \cdot q_{i,k}$
- 使用 SGD 优化目标： $min \sum_{R \in R_{train}} (\|R_{u,i} - \hat{R}_{u,i}\|_F^2 + \lambda(\|p_u^\top\|_F^2 + \|q_i\|_F^2))$
- BiasSVD： 加入偏移项
- 本质： 单层的图神经网络



Motivation: Embedding function 缺乏对协同信号的明确编码

协同信号 (Collaborative Signal)

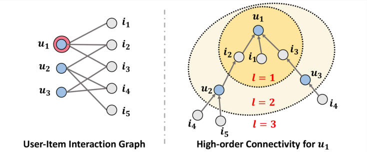
协同信号 (Collaborative Signal) 指的是隐藏在用户与项目交互数据中的潜在信息，这些信息能够揭示用户之间的行为相似性或项目之间的关联性。

Two key components in learnable CF models:

1) Embedding：将 User 和 Item 转化成 vectorized representations
2) Interact Modeling (e.g. inner product, non-linear NN, etc.)

Method of Distill collaborative Signal:

high-order connectivity（高阶连通性）
from user-item interactions



NGCF: Intro2

Contributions:

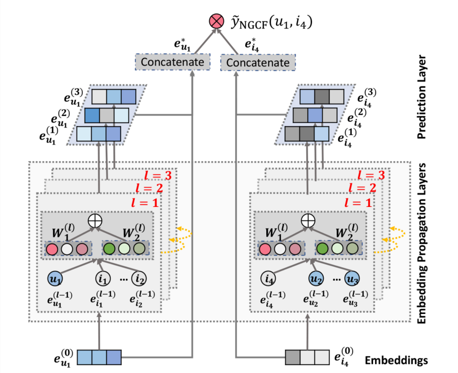
* Highlight the critical importance of **explicitly** exploiting the **collaborative signal** in the embedding function
* propose NGCF, a new recommendation framework based on graph neural network, which **explicitly encodes the collaborative signal** in the form of **high-order connectivities** by performing **embedding propagation**
* Conduct empirical studies on three million-size datasets

NGCF: Methodology

3 components:

(1) embedding layer
(2) multiple embedding propagation layers
(3) predict layer

Architecture:



Embedding layer and Embedding Propagation Layers (1)

Embedding layer:

describe each user u (or item i) with a embedding vector $e \in \mathbb{R}^d$, d is embedding size.

$$\mathbf{E} = [\underbrace{\mathbf{e}_{u_1}, \cdots, \mathbf{e}_{u_N}}_{\text{users embeddings}}, \underbrace{\mathbf{e}_{i_1}, \cdots, \mathbf{e}_{i_M}}_{\text{item embeddings}}].$$

Embedding Propagation Layers

- First-order Propagation
  - Message Construction
  - Message Aggregation
- High-order Propagation

Message Construction

W_1: item 信息，W_2： user-item 交互信息

$$\mathbf{m}_{u \leftarrow i} = \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \Big(\mathbf{W}_1 \mathbf{e}_i + \mathbf{W}_2 (\mathbf{e}_i \odot \mathbf{e}_u) \Big),$$

Message Aggregation

$$\mathbf{e}_u^{(1)} = \text{LeakyReLU}\Big(\mathbf{m}_{u \leftarrow u} + \sum_{i \in \mathcal{N}_u} \mathbf{m}_{u \leftarrow i} \Big),$$

Embedding Propagation Layers (2)

High-order Propagation

stack more embedding propagation layers to connect high-order collaborative signal.

$$\mathbf{e}_u^{(l)} = \mathrm{LeakyReLU}\left(\mathbf{m}_{u\leftarrow u}^{(l)} + \sum_{i\in\mathcal{N}_u} \mathbf{m}_{u\leftarrow i}^{(l)}\right),$$

$$\begin{cases} \mathbf{m}_{u\leftarrow i}^{(l)} = p_{ui}\left(\mathbf{W}_1^{(l)}\mathbf{e}_i^{(l-1)} + \mathbf{W}_2^{(l)}(\mathbf{e}_i^{(l-1)} \odot \mathbf{e}_u^{(l-1)})\right), \\ \mathbf{m}_{u\leftarrow u}^{(l)} = \mathbf{W}_1^{(l)}\mathbf{e}_u^{(l-1)}, \end{cases}$$

Propagation Rule in Matrix Form

$$\mathbf{E}^{(l)} = \mathrm{LeakyReLU}\left((\mathcal{L} + \mathbf{I})\mathbf{E}^{(l-1)}\mathbf{W}_1^{(l)} + \mathcal{L}\mathbf{E}^{(l-1)} \odot \mathbf{E}^{(l-1)}\mathbf{W}_2^{(l)}\right),$$

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \text{ and } \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{0} \end{bmatrix},$$

- $\mathbf{R}$: user-item 交互矩阵
- $\mathbf{D}$: 度矩阵（对角线元素是节点度数）
- $\mathcal{L}$: 拉普拉斯矩阵，非对角线元素是 $\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}$
- $\mathcal{L}+\mathbf{I}$: 为每个节点加入一个自环

Model Predict, Optimization

Model Predict

Concatenating the representations learned by different layers.

$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \| \cdots \| \mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)} \| \cdots \| \mathbf{e}_i^{(L)},$$

Estimate the user’ s preference towards the target item:

$$\hat{y}_{NGCF}(u,i) = e_u^{*\top} \cdot e_i^*$$

Optimization

$$Loss = \sum_{(u,i,j)\in O} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|_2^2, \qquad (11)$$

where $O = \{(u,i,j)|(u,i) \in \mathcal{R}^+, (u,j) \in \mathcal{R}^-\}$ denotes the pairwise training data, $\mathcal{R}^+$ indicates the observed interactions, and $\mathcal{R}^-$ is the unobserved interactions; $\sigma(\cdot)$ is the sigmoid function; $\Theta = \{\mathbf{E}, \{\mathbf{W}_1^{(l)}, \mathbf{W}_2^{(l)}\}_{l=1}^{L}\}$ denotes all trainable model parameters, and λ controls the L_2 regularization strength to prevent overfitting. We

Evaluation

Evaluation 流程

- 在训练集中未出现、但在原始数据中实际交互过的物品作为正例 (positive items); 与用户未交互过的物品为负例 (negative items)
- 对每个用户，对潜在推荐物品生成评分并排序
- 截取 Top-K 个物品
- 计算相关指标，`Recall@K` and NDCG@K

`Recall@K`

$$Recall@K = \frac{\text{Top-K 列表中属于测试集 positive items 的数量}}{\text{测试集中该用户所有 positive item 的数量}}$$

NDCG@K

CG 累积增益：Top-K 列表中的相关性总和（出现测试集 positive item 为 1，否则为 0）

DCG 折损累积增益：$DCG@K = \sum \frac{rel_i}{log_2(i+1)}$

NDCG 归一化折损累积增益：$NDCG@K = \frac{DCG@K}{IDCG@K}$,IDCG 是理想状态下的 DCG

LightGCN: Intro

Motivation: 直接基于 GCN 的 NGCF 设计复杂，其中的特征变换和非线性激活反而给模型训练带来困难。

Contributions:

- 证明了 GCN 中的 feature transformation 和 non-linear activation 对 collaborative filter 没有积极影响
- LightGCN 大大简化模型设计
- 对比 NGCF 产生显著的改进

LightGCN: Method

Light Graph Convolution (LGC)

$$\mathbf{e}_u^{(k+1)} = \sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(k)},$$
$$\mathbf{e}_i^{(k+1)} = \sum_{u\in\mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(k)}.$$

propagation rule in LightGCN

$$\mathbf{e}_u^{(k+1)} = \sigma\Big(\mathbf{W}_1\mathbf{e}_u^{(k)} + \sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}(\mathbf{W}_1\mathbf{e}_i^{(k)} + \mathbf{W}_2(\mathbf{e}_i^{(k)} \odot \mathbf{e}_u^{(k)}))\Big),$$
$$\mathbf{e}_i^{(k+1)} = \sigma\Big(\mathbf{W}_1\mathbf{e}_i^{(k)} + \sum_{u\in\mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}(\mathbf{W}_1\mathbf{e}_u^{(k)} + \mathbf{W}_2(\mathbf{e}_u^{(k)} \odot \mathbf{e}_i^{(k)}))\Big),$$

propagation rule in NGCF

Layer Combination

$$\mathbf{e}_u = \sum_{k=0}^{K} \alpha_k \mathbf{e}_u^{(k)}; \quad \mathbf{e}_i = \sum_{k=0}^{K} \alpha_k \mathbf{e}_i^{(k)},$$

$$\begin{cases} \mathbf{m}_{u\leftarrow i}^{(l)} = p_{ui}\Big(\mathbf{W}_1^{(l)}\mathbf{e}_i^{(l-1)} + \mathbf{W}_2^{(l)}(\mathbf{e}_i^{(l-1)} \odot \mathbf{e}_u^{(l-1)})\Big), \\ \mathbf{m}_{u\leftarrow u}^{(l)} = \mathbf{W}_1^{(l)}\mathbf{e}_u^{(l-1)}, \end{cases}$$

Why weighted sum?

1) 随层数增加，embedding 会过度平滑
2) 相加有利于捕获不同层次的语义
3) 加权相加实际上包含了自连接的效果

Method: Matrix Form

3.1.3 Matrix Form. We provide the matrix form of LightGCN to facilitate implementation and discussion with existing models. Let the user-item interaction matrix be $\mathbf{R} \in \mathbb{R}^{M\times N}$ where M and N denote the number of users and items, respectively, and each entry R_{ui} is 1 if u has interacted with item i otherwise 0. We then obtain the adjacency matrix of the user-item graph as

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{pmatrix}, \tag{6}$$

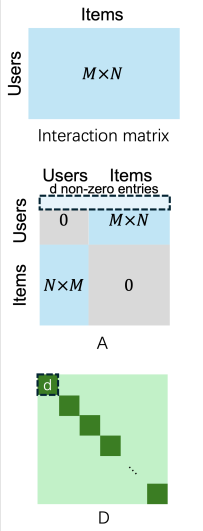
Let the 0-th layer embedding matrix be $\mathbf{E}^{(0)} \in \mathbb{R}^{(M+N)\times T}$, where T is the embedding size. Then we can obtain the matrix equivalent form of LGC as:

$$\mathbf{E}^{(k+1)} = (\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{E}^{(k)}, \tag{7}$$

where $\mathbf{D}$ is a $(M+N)\times(M+N)$ diagonal matrix, in which each entry D_{ii} denotes the number of nonzero entries in the i-th row vector of the adjacency matrix $\mathbf{A}$ (also named as degree matrix). Lastly, we get the final embedding matrix used for model prediction as:

$$\begin{aligned} \mathbf{E} &= \alpha_0\mathbf{E}^{(0)} + \alpha_1\mathbf{E}^{(1)} + \alpha_2\mathbf{E}^{(2)} + \dots + \alpha_K\mathbf{E}^{(K)} \\ &= \alpha_0\mathbf{E}^{(0)} + \alpha_1\tilde{\mathbf{A}}\mathbf{E}^{(0)} + \alpha_2\tilde{\mathbf{A}}^2\mathbf{E}^{(0)} + \dots + \alpha_K\tilde{\mathbf{A}}^K\mathbf{E}^{(0)}, \end{aligned} \tag{8}$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the symmetrically normalized matrix.



Model Analysis

通过修改 Weighted sum，LightGCN 与 SGCN、APPNP 是等价的。

SGCN 考虑了用户自连接，即度矩阵 $\mathbf{D}$ 加上单位矩阵

$$\mathbf{E}^{(k+1)} = (\mathbf{D}+\mathbf{I})^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})(\mathbf{D}+\mathbf{I})^{-\frac{1}{2}}\mathbf{E}^{(k)}, \tag{9}$$

where $\mathbf{I} \in \mathbb{R}^{(M+N)\times(M+N)}$ is an identity matrix, which is added on $\mathbf{A}$ to include self-connections. In the following analysis, we omit the $(\mathbf{D}+\mathbf{I})^{-\frac{1}{2}}$ terms for simplicity, since they only re-scale embeddings. In SGCN, the embeddings obtained at the last layer are used for downstream prediction task, which can be expressed as:

$$\begin{aligned}\mathbf{E}^{(K)} &= (\mathbf{A}+\mathbf{I})\mathbf{E}^{(K-1)} = (\mathbf{A}+\mathbf{I})^K\mathbf{E}^{(0)} \\ &= \binom{K}{0}\mathbf{E}^{(0)} + \binom{K}{1}\mathbf{A}\mathbf{E}^{(0)} + \binom{K}{2}\mathbf{A}^2\mathbf{E}^{(0)} + \ldots + \binom{K}{K}\mathbf{A}^K\mathbf{E}^{(0)}.\end{aligned} \tag{10}$$

APPNP 通过在每层补充 $E^{(0)}$ 初始特征，避免过于平滑

$$\mathbf{E}^{(k+1)} = \beta\mathbf{E}^{(0)} + (1-\beta)\tilde{\mathbf{A}}\mathbf{E}^{(k)}, \tag{11}$$

where β is the teleport probability to control the retaining of starting features in the propagation, and $\tilde{\mathbf{A}}$ denotes the normalized adjacency matrix. In APPNP, the last layer is used for final prediction, i.e.,

$$\begin{aligned}\mathbf{E}^{(K)} &= \beta\mathbf{E}^{(0)} + (1-\beta)\tilde{\mathbf{A}}\mathbf{E}^{(K-1)}, \\ &= \beta\mathbf{E}^{(0)} + \beta(1-\beta)\tilde{\mathbf{A}}\mathbf{E}^{(0)} + (1-\beta)^2\tilde{\mathbf{A}}^2\mathbf{E}^{(K-2)} \\ &= \beta\mathbf{E}^{(0)} + \beta(1-\beta)\tilde{\mathbf{A}}\mathbf{E}^{(0)} + \beta(1-\beta)^2\tilde{\mathbf{A}}^2\mathbf{E}^{(0)} + \ldots + (1-\beta)^K\tilde{\mathbf{A}}^K\mathbf{E}^{(0)}.\end{aligned} \tag{12}$$

Model Training

Employ the Bayesian Personalized Ranking (BPR) loss

$$L_{BPR} = -\sum_{u=1}^{M} \sum_{i \in \mathcal{N}_u} \sum_{j \notin \mathcal{N}_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda ||\mathbf{E}^{(0)}||^2$$

- **可学习的参数只有** $E^{(0)}$
- **不需要** dropout **机制**，L2 **正则化足够防止过拟合**

SASRec: Intro

Motivation:

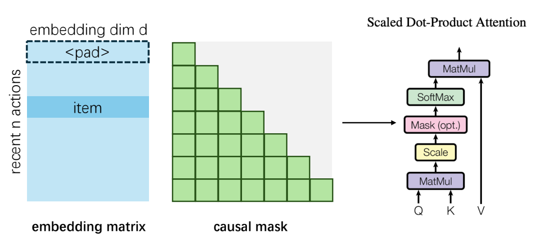
The goal of our work is to **balance these two goals**, by proposing a self-attention based sequential model (SASRec) that allows us to **capture long-term semantics** (like an RNN), but, using an attention mechanism, makes its predictions **based on relatively few actions** (like an MC).

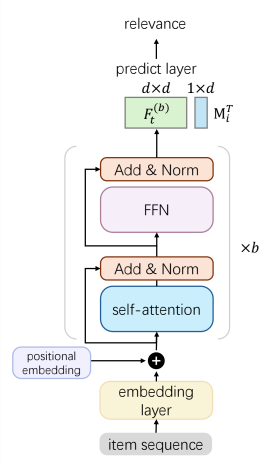
Contribution:

将注意力机制引入 Sequential Recommendation

SASRec: Method

A. Embedding layer
B. Self-attention Block
C. Stacking SA Block
D. Predict layer





Training & Complexity analysis

Training

训练使用交叉熵损失

Complexity analysis

- Space Complexity
  - $O(|\mathcal{I}|d + nd + d^2)$
- Time Complexity
  - computing: $O(n^2d + nd^2)$
  - testing: $O(n)$