Contents

1 推荐系统整体梳理

https://daiwk.github.io/posts/links-navigation-recommender-system.html

https://github.com/Doragd/Algorithm-Practice-in-Industry

王喆的机器学习笔记系列：

https://github.com/wzhe06/Reco-papers

https://github.com/wzhe06/Ad-papers

深度学习传送门系列：

https://github.com/imsheridan/DeepRec

推荐系统遇上深度学习系列：

链接：https://pan.baidu.com/s/1jZkJ2d9WckbZL48aGFudOA 密码:kme3

推荐系统技术演进趋势：召回-> 排序-> 重排

推荐系统的发展与 2019 最新论文回顾

深度推荐系统 2019 年度阅读收藏清单

推荐工业界实战角度详解 TensorFlow 中 Wide & Deep 源码（三）

2 特征工程

浅谈微视推荐系统中的特征工程

推荐系统之数据与特征工程

稠密特征加入 CTR 预估模型的方法汇总

3 工程优化

3.1 HugeCTR

点击率预估的训练传统上存在着几个困扰着广大开发者的问题：巨大的哈希表（Embedding Table），较少的矩阵计算，大量的数据吞吐。

HugeCTR 是首个全部解决以上问题的开源 GPU 训练框架，与现有 CPU 和混合 CPU / GPU 解决方案相比，它的速度提高了 12 倍至 44 倍。HugeCTR 是一种端到端训练解决方案，其所有计算都在 GPU 上执行，而 CPU 仅用于 I / O。GPU 哈希表支持动态缩放。它利用 MPI 进行多节点训练，以支持任意大的嵌入尺寸。它还还支持混合精度训练，在 Volta GPU 及其后续版本上可以利用 Tensor cores 进一步加速。

如何解决点击率预估？英伟达专家详解 HugeCTR 训练框架（二）

Merlin HugeCTR 分级参数服务器简介

3.2 BOX

大规模深度学习广告系统的分布式分层 GPU 参数服务器

Distributed Hierarchical GPU Parameter Server for Massive Scale Deep Learning Ads Systems

3.3 索引

3.3.1 ANN 索引

annoy hnsw faiss pq

3.3.2 暴力召回 ANN 加速

https://kexue.fm/archives/9336

大致思想，CUR 分解：query 和 item 的 MxN 打分矩阵，分解成 F(Mxk1), G(k1xk2), H(k2xN) 三个矩阵

- Mxk1 矩阵：原矩阵里搞 k1 列出来，即选出 k1 个种子 item，得到 F
- k2xN 矩阵：原矩阵里搞 k2 列出来，即选出 k2 个种子 query，得到 H
- k1xk2 矩阵：即矩阵 1 和矩阵 2 求交集，比如矩阵 1 是抽的第 1,23,54 列出来，矩阵 2 是抽的第 4,80 行出来，那交集元素就是 (1,4),(1,80),(23,4),(23,80),(54,4),(54,80) 这 6 个点，构成 k1xk2 矩阵，然后算一下伪逆得到 G

建索引：＋挑出种子 query，和所有 item 两两计算相似度，得到 H 矩阵 ＋挑出种子 item，和种子 query 两两计算相似度，再算伪逆，得到 G 矩阵 ＋计算 G*H，存起来

检索：＋输入的 query 和 k1 个种子 item 算一下相似度，得到 1xk1 的矩阵 q＋q 和 GH 相乘，就能得到 q 和每个 $item$ 的相似度了 ＋【这步可以 ann 化】：GH 就是 $k1$N，按列来看，就是 N 个 k1 维向量，相当于 N 个 item 向量，扔到 annlib 里去就行了，而输入的 q 也是一个 k1 维向量，就可以 ann 了

4 召回

360 展示广告召回系统的演进

推荐场景中深度召回模型的演化过程

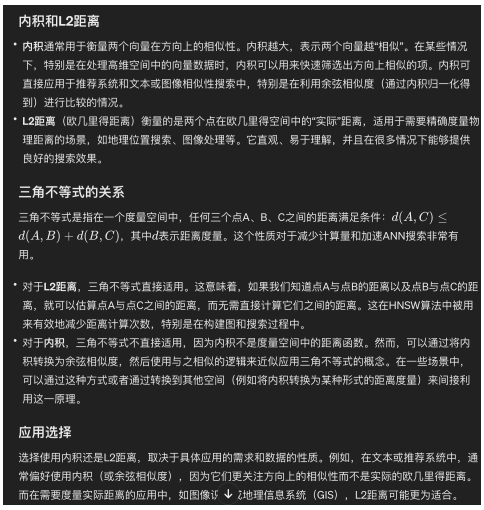
https://github.com/imsheridan/DeepRec/tree/master/Match

精准推荐的秘术：阿里解耦域适应无偏召回模型详解对应Co-training Disentangled Domain Adaptation Network for Leveraging Popularity Bias in Recommenders

推荐系统多兴趣召回论文解读

模型	年份	会议	公司	多兴趣提取阶段	训练阶段	多兴趣聚合阶段（线上阶段）	亮点	不足
MIND	2019	CIKM	阿里	胶囊网络	label-aware attention	K*N召回 取topN	• 使用胶囊网络来提取用户多兴趣表示。	• 基于target label的训练方式存在训练测试不一致的问题 • 没有考虑兴趣组合
ComiRec	2020	KDD	阿里	ComiRec-DR ComiRec-SA	使用与target item最近的兴趣	相关性和多样性权衡	• 使用胶囊网络和self-attentive来提取用户多兴趣表示。 • 线上serving时，同时考虑相关性和多样性。	• 基于target label的训练方式存在训练测试不一致的问题 • 用户都使用固定的兴趣数量（文中是4）
SINE	2021	WSDM	阿里	概念激活 + self-attentive	intention-aware attention	K*N召回 取topN	• 分别提出了新的兴趣聚类算法和兴趣聚合算法。 • 使用协方差正则化来引导概念池的学习	• 用户都使用固定的兴趣数量（文中是8）
Octopus	2020	SIGIR	MSRA	信道激活 +attention	使用与target item最近的兴趣	1.K*N召回 取topN 2. 额外构建一个多分类任务	• 自适应选取兴趣数量	• 非端到端 • 多兴趣提取比较粗暴

4.1 内积、余弦和 L2



内积和L2距离

- **内积**通常用于衡量两个向量在方向上的相似性。内积越大，表示两个向量越"相似"。在某些情况下，特别是在处理高维空间中的向量数据时，内积可以用来快速筛选出方向上相似的项。内积可直接应用于推荐系统和文本或图像相似性搜索中，特别是在利用余弦相似度（通过内积归一化得到）进行比较的情况。
- **L2距离**（欧几里得距离）衡量的是两个点在欧几里得空间中的"实际"距离，适用于需要精确度量物理距离的场景，如地理位置搜索、图像处理等。它直观、易于理解，并且在很多情况下能够提供良好的搜索效果。

三角不等式的关系

三角不等式是指在一个度量空间中，任何三个点A、B、C之间的距离满足条件：$d(A,C) \leq d(A,B) + d(B,C)$，其中$d$表示距离度量。这个性质对于减少计算量和加速ANN搜索非常有用。

- 对于**L2距离**，三角不等式直接适用。这意味着，如果我们知道点A与点B的距离以及点B与点C的距离，就可以估算点A与点C之间的距离，而无需直接计算它们之间的距离。这在HNSW算法中被用来有效地减少距离计算次数，特别是在构建图和搜索过程中。
- 对于**内积**，三角不等式不直接适用，因为内积不是度量空间中的距离函数。然而，可以通过将内积转换为余弦相似度，然后使用与之相似的逻辑来近似应用三角不等式的概念。在一些场景中，可以通过这种方式或者通过转换到其他空间（例如将内积转换为某种形式的距离度量）来间接利用这一原理。

应用选择

选择使用内积还是L2距离，取决于具体应用的需求和数据的性质。例如，在文本或推荐系统中，通常偏好使用内积（或余弦相似度），因为它们更关注方向上的相似性而不是实际的欧几里得距离。而在需要度量实际距离的应用中，如图像识 ↓ 2地理信息系统（GIS），L2距离可能更为适合。

给定 a，找到和它最像的 b

$ab = ||a||cos\theta||b||$

如果用内积，会找 $cos\theta||b||$ 最大的 b 出来，可能是夹角小，也可能是模大的 b，所以可能偏热门

4.2 采样

batch 内 shuffle 采样（有放回）

On Sampling Strategies for Neural Network-based Collaborative Filtering

浅谈个性化推荐系统中的非采样学习

Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations

https://www.tensorflow.org/extras/candidate_sampling.pdf

下载了一份：https://github.com/daiwk/collections/blob/master/assets/candidate_sampling.pdf

推荐系统遇上深度学习 (七十二)-[谷歌] 采样修正的双塔模型

4.3 突破双塔——TDM 系列

4.3.1 TDM->JTM

下一代深度召回与索引联合优化算法 JTM

4.3.2 二向箔

xx

4.4 突破双塔——DR

字节最新复杂召回模型，提出深度检索 DR 框架解决超大规模推荐系统中的匹配问题

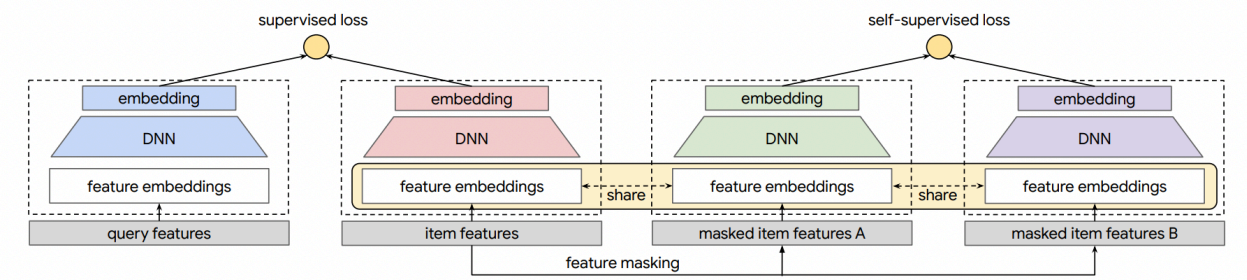
Deep Retrieval: An End-to-End Learnable Structure Model for Large-Scale Recommendations

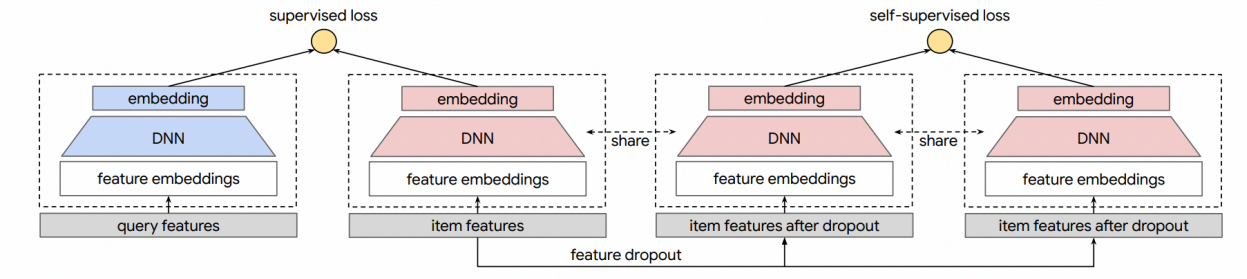
4.5 对比学习

4.5.1 自监督

Self-supervised Learning for Large-scale Item Recommendations

v3 有两个图：https://arxiv.org/pdf/2007.12865v3.pdf





4.5.2 qalign

Spherical Graph Embedding for Item Retrieval in Recommendation System

自己下载了

代码：https://github.com/WNQzhu/Q-align

自己的注释：https://github.com/daiwk/llms_new/blob/main/demos/qalign.py

假设 $N_K(u)$ 是节点 u 的 K 跳邻居，那么目标函数是最大化这些邻居的概率，即

$\max_f \sum_{u \in \mathcal{V}} \log \Pr\left(N_K(u) \mid f(u)\right)$

5 精排

5.1 传统 ctr

https://daiwk.github.io/posts/dl-traditional-ctr-models.html

5.1.1 lr for ctr

Simple and scalable response prediction for display advertising

Online Models for Content Optimization

5.1.2 gbdt for ctr

gbdt 基础知识：

https://zhuanlan.zhihu.com/p/86263786

bagging 全称叫 bootstrap aggregating，每个基学习器都会对训练集进行有放回抽样得到子训练集，比较著名的采样法为 0.632 自助法。每个基学习器基于不同子训练集进行训练，并综合所有基学习器的预测值得到最终的预测结果。bagging 常用的综合方法是投票法，票数最多的类别为预测类别。

boosting 训练过程为阶梯状，基模型的训练是有顺序的，每个基模型都会在前一个基模型学习的基础上进行学习，最终综合所有基模型的预测值产生最终的预测结果，用的比较多的综合方式为加权法。

stacking 是先用全部数据训练好基模型，然后每个基模型都对每个训练样本进行的预测，其预测值将作为训练样本的特征值，最终会得到新的训练样本，然后基于新的训练样本进行训练得到模型，然后得到最终预测结果。

bagging 和 stacking 中的基模型为强模型（偏差低，方差高），而 boosting 中的基模型为弱模型（偏差高，方差低）。

bagging 的特点：

- 整体模型的期望等于基模型的期望，这也就意味着整体模型的偏差和基模型的偏差近似。
- 整体模型的方差小于等于基模型的方差，当且仅当相关性为 1 时取等号，随着基模型数量增多，整体模型的方差减少，从而防止过拟合的能力增强，模型的准确度得到提高。

所以，bagging 中的基模型一定要为强模型，如果 bagging 使用弱模型则会导致整体模型的偏差提高，而准确度降低。

boosting 的特点：

- 整体模型的方差等于基模型的方差，如果基模型不是弱模型，其方差相对较大，这将导致整体模型的方差很大，即无法达到防止过拟合的效果。因此，boosting 框架中的基模型必须为弱模型。
- boosting 框架中采用基于贪心策略的前向加法，整体模型的期望由基模型的期望累加而成，所以随着基模型数的增多，整体模型的期望值增加，整体模型的准确度提高。

gbdt 与 Adaboost 对比

相同：

- 都是 boosting，使用弱分类器；
- 都使用前向分布算法；

不同：

- 迭代思路不同：adaboost 是通过提升错分数据点的权重来弥补模型的不足（利用错分样本），而 GBDT 是通过算梯度来弥补模型的不足（利用残差）；
- 损失函数不同：adaBoost 采用的是指数损失，GBDT 使用的是绝对损失或者 Huber 损失函数；

Learning the click-through rate for rare/new ads from similar ads

Using boosted trees for click-through rate prediction for sponsored search

Improving Ad Relevance in Sponsored Search

Stochastic Gradient Boosted Distributed Decision Trees

https://zhuanlan.zhihu.com/p/148050748

5.2 深度学习 ctr

https://daiwk.github.io/posts/dl-dl-ctr-models.html

5.3 序列建模

一文看懂序列推荐建模的最新进展与挑战

从 MLP 到 Self-Attention，一文总览用户行为序列推荐模型

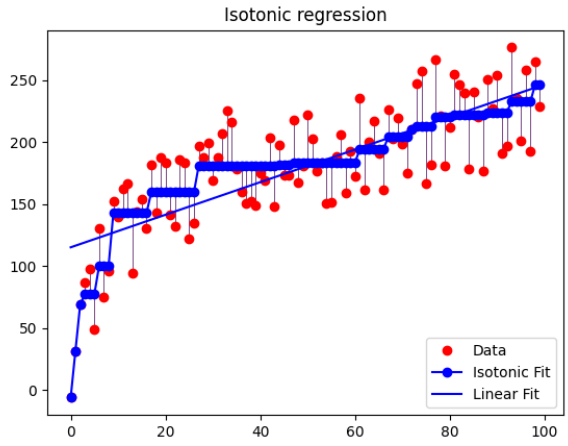
Transformer 在推荐模型中的应用总结

阿里妈妈点击率预估中的长期兴趣建模

DCN V2：Google 提出改进版 DCN，用于大规模排序系统中的特征交叉学习 (附代码)

5.4 保序回归

参考https://zhuanlan.zhihu.com/p/88623159的代码，能画出下面的图



对于二分类问题，参考https://zhuanlan.zhihu.com/p/101766505

对 lr+gbdt 的负采样校准的方法

Practical Lessons from Predicting Clicks on Ads at Facebook

5.5 cvr 预估

ecpc：用户给定一个粗粒度出价，模型可以在一定的范围内调价 ocpc：完全以模型出价为准

delay feedback https://zhuanlan.zhihu.com/p/555950153

5.6 时长预估

快手 kdd 2022

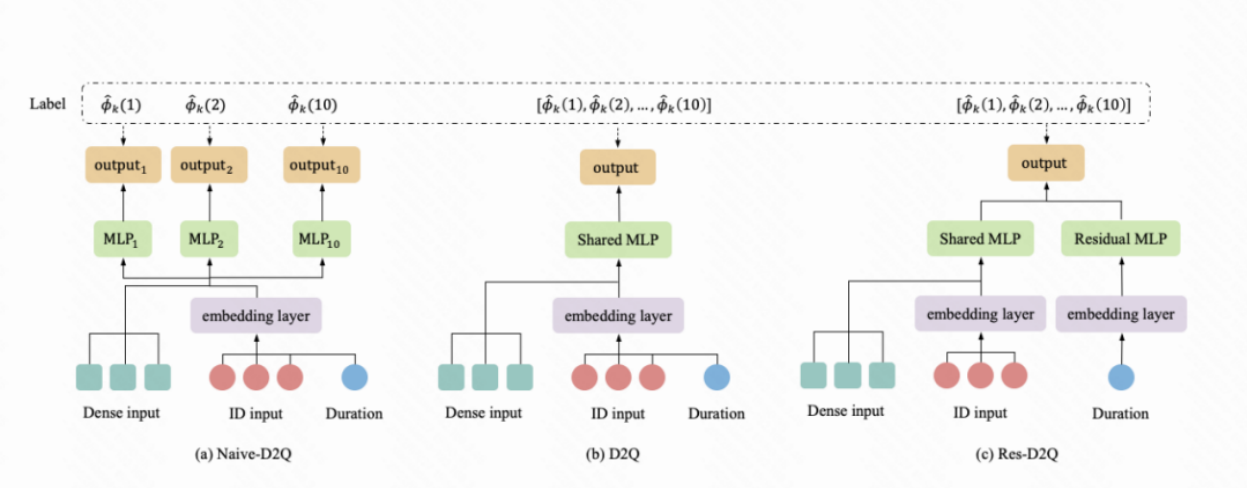
Deconfounding Duration Bias in Watch-time Prediction for Video Recommendation

短视频推荐视频时长 bias 问题

拿物理时长（duration）分桶

D2Q 算法的具体做法如下：

- 统计训练样本的 duration 分布，得到等频分桶分位点；
- 将样本按照等频分桶分位点分成 k 个相互独立的分桶 D_k；
- 对不同 duration 分桶的样本，在组内统计时长分位数作为 label，得到 Duration-Aware Watchtime-Distribution label；
- 分别在上述的分桶上训练时长预估模型 f_k；



- 图 a：M 个网络完全独立，分别学习各自的 label，不共享特征 embedding，特征 embedding 空间随着分桶维度扩大线性增加，存储、训练的资源开销随之增加，实现成本较高，不符合工业界场景的要求；
- 图 b：M 个网络共享底层特征，如果采用多输出的训练方式，则 batch 内样本分布不均的问题会导致子塔训练不稳定，收敛到局部最优。**单塔单输出**的训练方式在实际训练时效果稳定，收敛速度较快，是 D2Q 实现的基线版本。
- 图 c：单塔单输出模型中引入 Duration bias 模块，用于建模不同分桶下的样本差异（Res-D2Q），离线训练指标得到进一步的提升。

论文使用 XAUC、XGAUC 以及 MAE 等指标对时长回归效果进行评估。MAE 表示短视频预估时长与观看时长 label 的误差绝对值，表示模型回归精度，是回归任务的常用评估指标。

- XAUC：将测试集中的样本两两组合，若组合的标签和预估值的序一致则为正序，否则为逆序，XAUC 是正序对数与总组合数的比值；
- XGAUC：用户维度计算的 XAUC。

由于推荐系统主要优化候选集的排序，评估指标 XAUC 能够更加直观的反映预估时长序的好坏，与论文的优化目标更加适配。

6 多目标

6.1 多目标 + 推荐综述

Multi-task 多任务模型在推荐算法中应用总结 1

Multi-task 多任务学习在推荐算法中应用 (2)

多任务学习在推荐算法中的应用

6.2 阿里多目标

阿里提出多目标优化全新算法框架，同时提升电商 GMV 和 CTR

6.3 Youtube 多目标——MMoE

YouTube 多目标排序系统：如何推荐接下来收看的视频

https://daiwk.github.io/posts/dl-youtube-multitask.html

6.4 CGC

cgc 参考 paddle 代码：cgc_demo.py

7 多场景

7.1 APG

APG: 面向 CTR 预估的自适应参数生成网络

摘要：目前基于深度学习的 CTR 预估模型（即 Deep CTR Models）被广泛的应用于各个应用中。传统的 Deep CTR Models 的学习模式是相对静态的，即所有的样本共享相同的网络参数。然而，由于不同样本的特征分布不尽相同，这样一种静态方式很难刻画出不同样本的特性，从而限制了模型的表达能力，导致次优解。在本文中，我们提出了一个高效率、高效果的通用模块，称为自适应参数生成网络 (APG)。其可以基于不同的样本，动态的为 CTR 模型生成不同的模型参数。大量的实验表明，APG 能够被应用于各种 CTR 模型，并且显著的提升模型效果，同时能节省 38.7% 的时间开销和 96.6% 的存储。APG 已在阿里巴巴搜索广告系统部署上线，并获得 3% 的点击率增长和 1% 的广告收入增长。

APG: Adaptive Parameter Generation Network for Click-Through Rate Prediction

8 item 冷启

poso

Personalized Cold Start Modules for Large-scale Recommender Systems

https://zhuanlan.zhihu.com/p/534056942

9 用户冷启

9.1 PeterRec

仅需少量视频观看数据，即可精准推断用户习惯：腾讯、谷歌、中科大团队提出迁移学习架构 PeterRec

Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation

https://github.com/fajieyuan/sigir2020_peterrec

搞一个 pretrain-finetune 的架构，学好一套用户的表示，可以给各种下游任务用。

采用如下方式：

- **无监督**地学习用户表示：使用**序列模型**，**预测**用户的**下一次点击**。为了能建模**超长**的 u-i 交互序列，使用类似 NextItNet（A Simple Convolutional Generative Network for Next Item Recommendation）的模型
- 使用预训练好的模型去**有监督**地 finetune 下游任务
- 在各个下游任务间，想要尽可能共享更多的网络参数：参考 learning to learn，即一个网络的大部分参数可以其他参数来预测（一层里 95% 的参数可以通过剩下的 5% 的参数来预测）。文章提出了 model patch(模型补丁)，每个模型补丁的参数量不到原始预训练模型里的卷积层参数的 10%。通过加入模型补丁，不仅可以保留原来的预训练参数，还可以更好地适应下游任务。模型补丁有串行和并行两种加入方式。

序列推荐模型：

- RNN：强序列依赖
- CNN：可并行，能比 RNN 叠更多层，所以准确率更高。难以建模长序列是因为卷积核一般都比较小（如 3x3），但可以通过空洞 (dilated) 卷积来解决，可以使用不变的卷积核，指数级地扩充表示域。
- 纯 attention：可并行，例如 SASRec（Self-attentive sequential recommendation）。但因为时间和存储消耗是序列长度的平方的复杂度。

考虑到用户的点击序列往往成百上千，所以使用类似 NextItNet 的 casual 卷积，以及类似 GRec（Future Data Helps Training: Modeling Future Contexts for Session-based Recommendation）的双向 encoder 的这种 non-casual 卷积。

与推荐系统现有的 transfer learning 对比：

- DUPN：
  - 训练的时候就有多个 loss。如果没有相应的 loss 和 data，学好的用户表示效果就会很差。而本文只有一个 loss，却能用在多个 task 上，所以算是一种 multi-domain learning(Efficient parametrization of multi-domain deep neural networks)
  - DUPN 在用户和 item 特征上需要很多特征工程，并没有显式地对用户的行为序列建模
  - DUPN 要么 finetune 所有参数，要么只 finetune 最后一个分类层。PeterRec 则是对网络的一小部分进行 finetune，效果并不比全 finetune 差，比只 finetune 最后一个分类层要好很多

- CoNet：杨强提出的Conet: Collaborative cross networks for cross-domain recommendation
  - cross-domain 用于推荐的一个网络。同时训练 2 个目标函数，一个表示 source 网络，一个表示 target 网络。
  - pretrain+finetune 效果不一定好，取决于预训练的方式、用户表示的表达能力、预训练的数据质量等

预训练时没有 [TCL]，fintune 时加上。

- 原 domain$S$：有大量用户交互行为的图文或视频推荐。一条样本包括 $(u, x^u) \in \mathcal{S}$，其中，$x^u = \{x_1^u, ..., x_n^u\}$ $(x_i^u \in X)$ 表示用户的点击历史
- 目标 domain$T$：可以是用户 label 很少的一些预测任务。例如用户可能喜欢的 item、用户性别、用户年龄分桶等。一条样本包括 $(u, y) \in \mathcal{T}$，其中 $y \in \mathcal{Y}$ 是一个有监督的标签。

10 GNN+ 推荐

https://zhuanlan.zhihu.com/p/323302898

Graph Neural Networks in Recommender Systems: A Survey

Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions

11 强化学习 + 推荐

12 LLM+ 推荐

13 bias v.s. debias

推荐系统炼丹笔记：推荐系统 Bias 大全 | Debias 方法综述

13.1 position bias

搜索、推荐业务中 - position bias 的工业界、学术界发展历程 - 系列 1(共计 2)

推荐系统遇上深度学习 (七十一)-[华为] 一种消除 CTR 预估中位置偏置的框架

PAL: A Position-bias Aware Learning Framework for CTR Prediction in Live Recommender Systems

推荐系统之 Position-Bias 建模

14 工业界的一些推荐应用

14.1 dlrm

Facebook 深度个性化推荐系统经验总结 (阿里内部分享 PPT))

14.2 instagram 推荐系统

Facebook 首次揭秘：超过 10 亿用户使用的 Instagram 推荐算法是怎样炼成的?

https://venturebeat.com/2019/11/25/facebook-details-the-ai-technology-behind-instagram-explore/

Instagram 个性化推荐工程中三个关键技术是什么?

14.3 微信读书推荐系统

微信读书怎么给你做推荐的?

14.4 youtube 推荐梳理

一文总览近年来 YouTube 推荐系统算法梳理

15 其他

15.1 混合推荐架构

混合推荐系统就是多个推荐系统“大杂烩”吗?

15.2 认知推荐

NeurIPS 2019 | 从感知跃升到认知，这是阿里在认知智能推荐领域的探索与应用

Learning Disentangled Representations for Recommendation