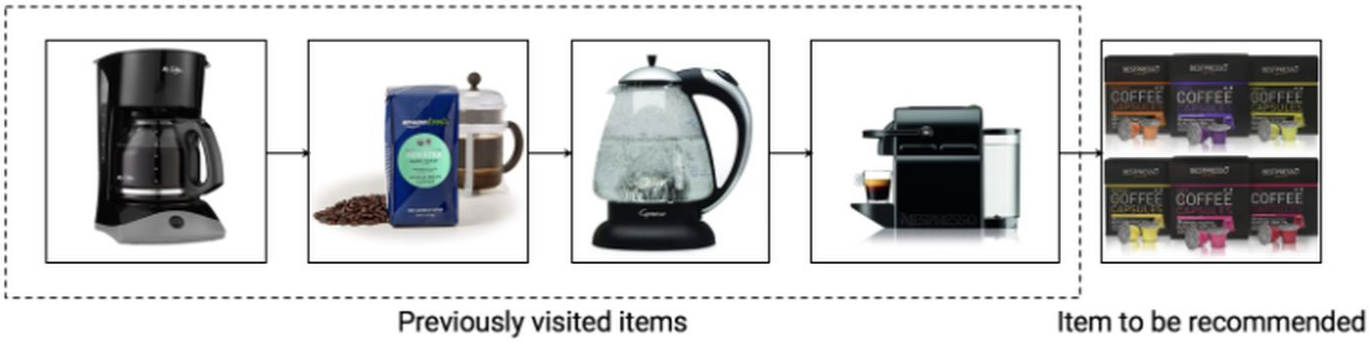


Session-based推荐算法实践与应用

icebear
中国科学技术大学 控制科学与工程硕士

背景

基于会话的推荐算法(Session-based Recommendation)是指在用户未登录状态下, 仅仅依赖匿名会话进行用户下一个行为预测的一种算法, 在许多领域(如电商、短视频、直播等)有着重要的作用.

Session-based Recommendation易与序列推荐(Sequencial Recommendation)混淆, 这里区分一下,序列推荐中常将用户长期历史序列建模表征用户, 可能还包括用户画像等信息, 典型算法如**MIND**、**SDM**等; 而基于会话的推荐中, session行为长度相对更短, 且用户长期偏好完全未知, 主要侧重于建模用户近期实时兴趣, 可以视为序列推荐的子领域.

业界主流推荐系统大致分为召回、粗排、排序和重排四个阶段. Session-based Recommendation的相关工作主要应用在我们的召回阶段. 在实际多路召回场景中, 利用SR-GNN算法作为一路召回队列, 针对纯新用户, 该队列也能够提供粗排能力.

经典算法回顾

基于会话推荐, 简单的可直接根据session内的item进行I2I扩充, 但每个item取多少?session内多个I2I队列如何融合? 往往需要拍脑袋; 传统方法如马尔科夫链(Markov chain)也可进行next item predict, 但其依赖的强假设: 下一状态只能由当前状态决定, 在时间序列中之前的行为均与之无关, 导致其在实际场景中运用受限; 近期Session-based推荐的SOTA结果都是基于神经网络模型取得的. 2016年提出的**GRU4Rec1**是该系列中经典的一篇, 首次利用RNN对session序列建模, 取得了很

item之间复杂的过渡模式.

1. GRU4Rec

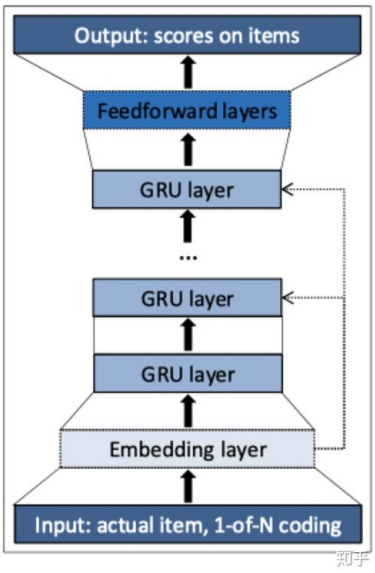
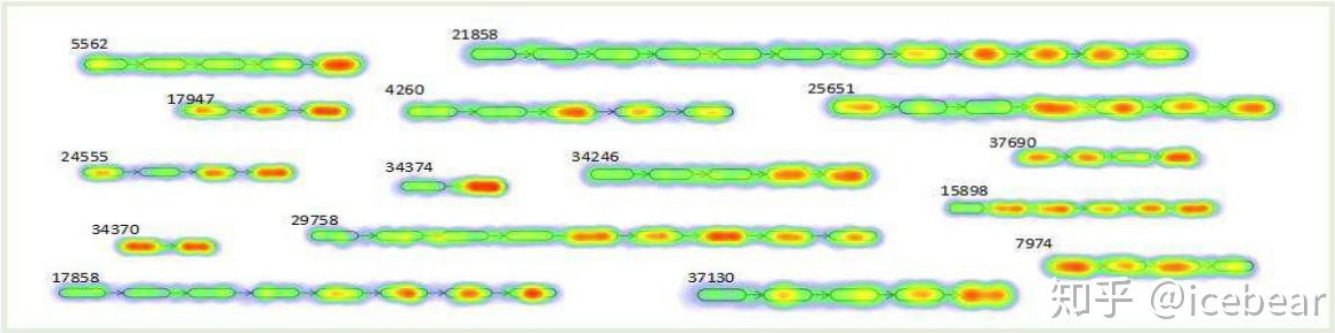


图1 GRU4Rec网络结构

现在来看, GRU4Rec网络结构比较简单, 输入是用户匿名session序列 $\boldsymbol{s}$. 首先, 对序列中item进行one-hot编码, 接着从Embedding层获得item向量表征, 经过堆叠的多层GRU更新, 最后经由全连接层计算下一个被点击item的概率. 本质上, GRU层相当于对session序列进行编码, 获得其向量表征, 在输出层计算与item向量的点积, softmax输出作为预测概率. 工程实现上有以下3个创新点, 也在后续的推荐算法落地中被大家所借鉴:

- 并行会话最小批训练(session-parallel mini-batches)
- Batch内负采样(sampling on the output)
- Pair-wise损失(rank loss)

2. NARM



等类似的session, 上文提到的GRU4Rec就不能很好地处理. NARM则设计了两路多层GRU编码器并引入注意力机制, 分别建模**user's sequential behavior**和**user's main purpose**, 具体结构如下图:

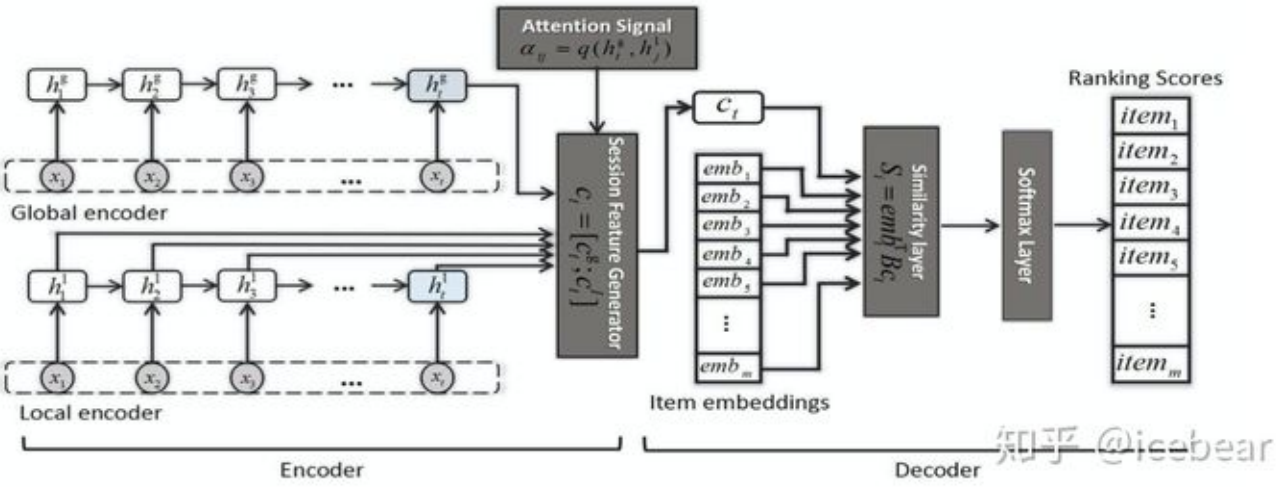


图3 NARM结构图

NARM设计为Encoder-decoder结构, 编码器部分包括Global encoder和Local encoder, 均由堆叠的多层GRU构成, 分别对用户序列行为 $\mathbf{c_t^g}$ 和主要意图 $\mathbf{c_t^l}$ 进行建模. 其中, 利用注意力机制进行主要意图学习:

$$\mathbf{c_t^l} = \sum_{j=1}^{t} \alpha_{tj} \mathbf{h_j}, \quad (1)$$
$$\alpha_{tj} = q(\mathbf{h_t}, \mathbf{h_j}).$$

Session Feature Generator模块将 $\mathbf{c_t^g}$ 和 $\mathbf{c_t^l}$ 拼接, 形成session的最终隐含表示, 其中既包含了用户序列行为, 又涵盖了用户主要意图. 后面的Top-N预测过程看做是解码器, 在NARM中, 没有采用更常见的点积运算作为相似度量, 而是提出了一种双线性(bi-linear)相似函数, 不仅能缓解网络参数量过大的问题, 还提升了模型的精度, 具体计算如下:

$$S_i = \mathbf{emb_i}^T \mathbf{B} \mathbf{c_t}. \quad (2)$$

3. STAMP

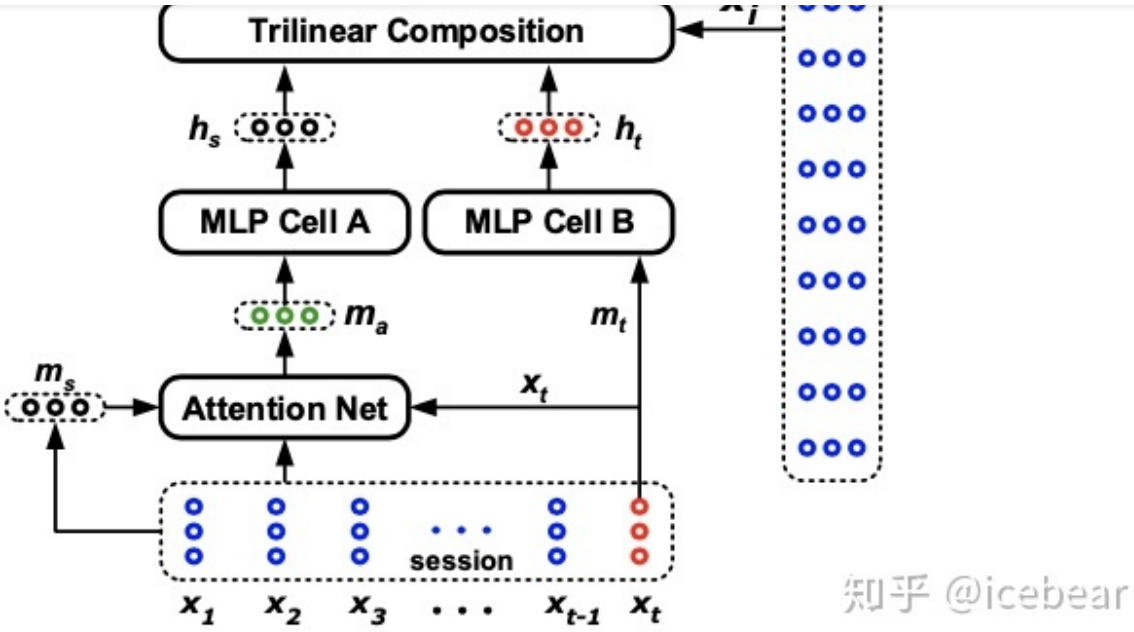


图4 STAMP结构图

对比NARM结构看, STAMP同样设计了不同结构分别对session内长期兴趣和短期兴趣建模, 也同样是取序列中最后一个交互的item表征短期兴趣. 不同的是, STAMP的网络设计地更加简单, 序列 $s = [x_1, x_2, x_3, \ldots, x_{t-1}, x_t]$, 取最后item的embedding经过MLP后得到短期兴趣表征 h_t. 同时设计了注意力网络(Attention Net)对session内长期兴趣进行提取, 该模块也比较好理解, 具体计算如下:

$$\alpha_i = \mathbf{W_0}\sigma(\mathbf{W_1}\mathbf{x_i} + \mathbf{W_2}\mathbf{x_t} + \mathbf{W_3}\mathbf{m_s} + \mathbf{b_a}), \tag{3}$$

$$\mathbf{m_s} = \frac{1}{t}\sum_{i=1}^{t}\mathbf{x_i},$$

$$\mathbf{m_a} = \sum_{i=1}^{t}\alpha_i\mathbf{x_i}.$$

$\mathbf{m_a}$ 同样经过MLP, 结构与MLP Cell B相同, 只是参数独立, 得到长期兴趣表征 h_s. 最后, 在Trilinear Composition模块中进行兴趣拼接和预测, 通过Hadamard积把长短期兴趣进行组合, 再与候选item向量点积:

$$y_i = \mathbf{x_i}^T(\mathbf{h_s} \odot \mathbf{h_t}). \tag{4}$$

4. SR-GNN

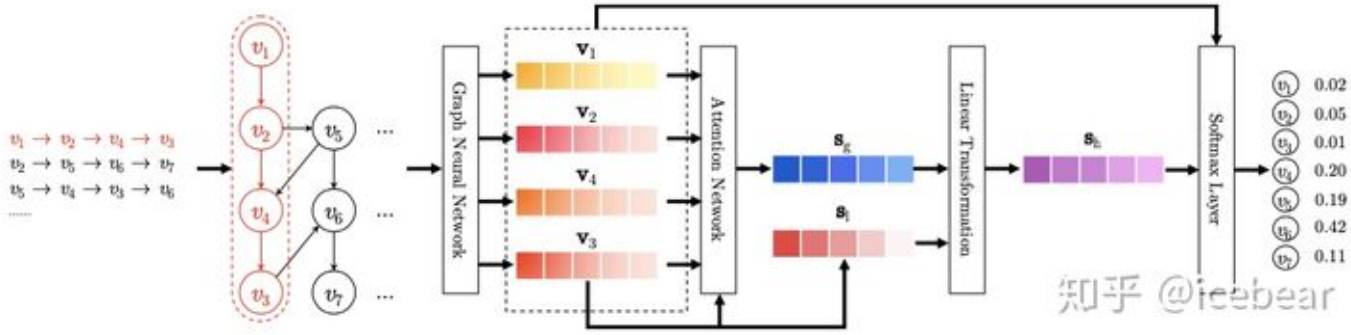
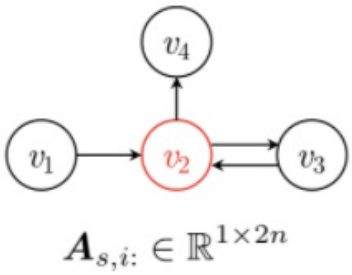


图4 SR-GNN

从左至右, 依次包含四个步骤, 分别为: Constructing session graphs、Node representation learning、Session representation generating和Making recommendation:

(1) 首先, 每个session序列 s 都可以构建为一个有向图 $G_s = (V_s, E_s)$, 对图中边的权重按照起始结点出度进行归一化. 例如, 某一session序列为 $s_i = [v_1, v_2, v_3, v_2, v_4]$, 其session图结构及连接矩阵如下:



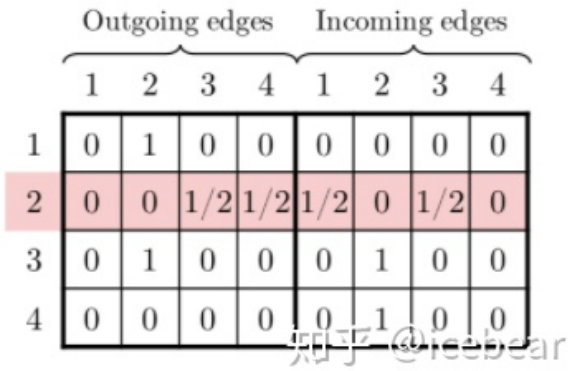


图5 SR-GNN图结构及其连接矩阵

解释一下算法中有向图的连接矩阵是如何构建的. 假设下图左侧是某一Session graph的邻接矩阵表示, 行表示源结点, 列表示目标结点. 可以看到, 结点 v_2 的出度、入度均为2. 接着分别对结点按照出度、入度进行归一化, 可以得到出度邻接矩阵和入度邻接矩阵, 拼接构成上图右侧的连通矩阵.

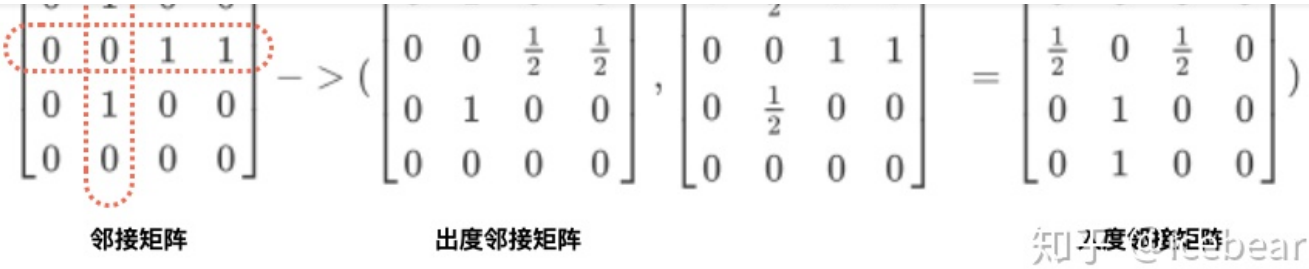


图6 有向图的出度邻接矩阵与入度邻接矩阵

(2) 采用GGNNs对session graph中的所有结点进行统一表征学习,主要传播规则如下, 公式第一行通过连接矩阵聚合邻居结点信息(包含了入和出两个方向), 剩余过程类似GRU参数更新：

$$
\begin{aligned}
a_{s,i}^t &= A_{s,i:}[v_1^{t-1},\ldots,v_n^{t-1}]^T H + b, \qquad (5)\\
z_{s,i}^t &= \sigma(W_z a_{s,i}^t + U_z v_i^{t-1}),\\
r_{s,i}^t &= \sigma(W_r a_{s,i}^t + U_r v_i^{t-1}),\\
\overline{v_i^t} &= tanh(W_o a_{s,i}^t + U_o(r_{s,i}^t \odot v_i^{t-1})),\\
v_i^t &= (1 - z_{s,i}^t) \odot v_i^{t-1}) + z_{s,i}^t \odot \overline{v_i^t}.
\end{aligned}
$$

(3) 获得item embedding向量后, 接着生成session embedding. 取session内最后一个交互的结点向量作为用户当前兴趣向量(**local embedding**), 以凸显最后交互item的重要性. 接着, 通过soft-attention网络获得**global embedding**以表征session长期兴趣. 最后, 通过一个简单线性函数做融合, 得到session的**hybird embedding**.

$$
\begin{aligned}
\alpha_i &= q^T \sigma(W_1 v_n + W_2 v_i + c), \qquad (6)\\
s_l &= v_n,\\
s_g &= \sum_{i=1}^n \alpha_i v_i,\\
s_h &= W_3[s_l : s_g].
\end{aligned}
$$

(4) session embedding与候选item embedding点积作为预测值, 进行TopN推荐.

总结

好. 比如, 利用近一个月的数据去训练模型可能要比用近三个月的, 在验证集上的指标更好. 而在实时兴趣推断时, 也存在类似情况, 当session长度超过20后, Recall、NDCG等离线指标也都发生了明显下降. 我们在SR-GNN基础上进行了一些改进(数据增强、N ormalize embedding、position embedding等), 作为线上多路召回中一路实时召回队列使用.

参考文献

1. **[2016][GRU4Rec] Session-based recommendations with recurrent neural networks**
2. **[2017][NARM] Neural Attentive Session-based Recommendation**
3. **[2018][STAMP] STAMP:Short-Term Attention/Memory Priority Model for Session-based Recommendation**
4. **[2019][SR-GNN] Session-Based Recommendation with Graph Neural Networks**
5. **[2019][NISER] NISER: Normalized Item and Session Representations to Handle Popularity Bias**
6. **[2020][TAGNN] TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation**
7. **[2020][TailNet] Long-tail Session-based Recommendation**

编辑于 2020-11-20

推荐系统 推荐系统实现 深度学习（Deep Learning）

文章被以下专栏收录

推荐+广告+搜索
千呼万唤始出来，犹抱琵琶半遮面
关注专栏

推荐召回算法
收录推荐系统召回算法
关注专栏

推荐阅读