

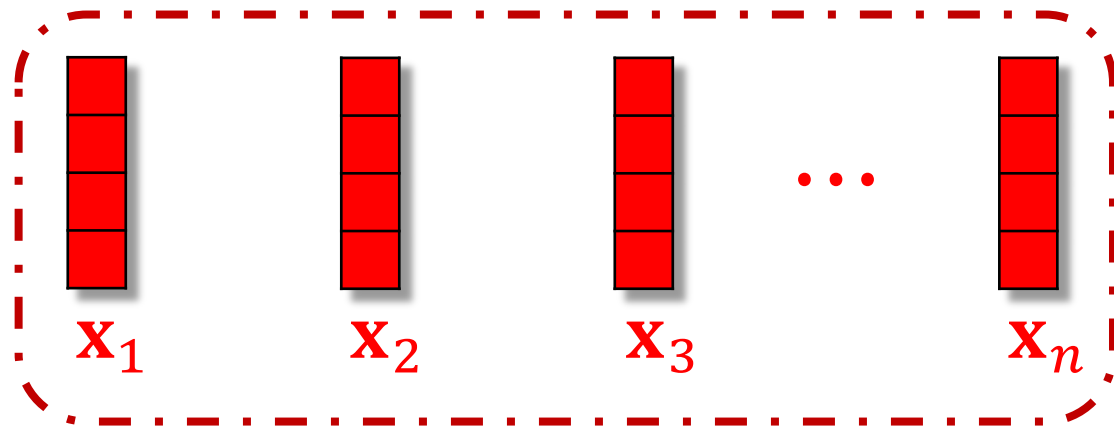
SIM模型

王树森

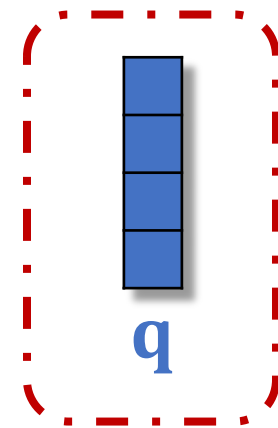
<http://wangshusen.github.io/>



DIN模型

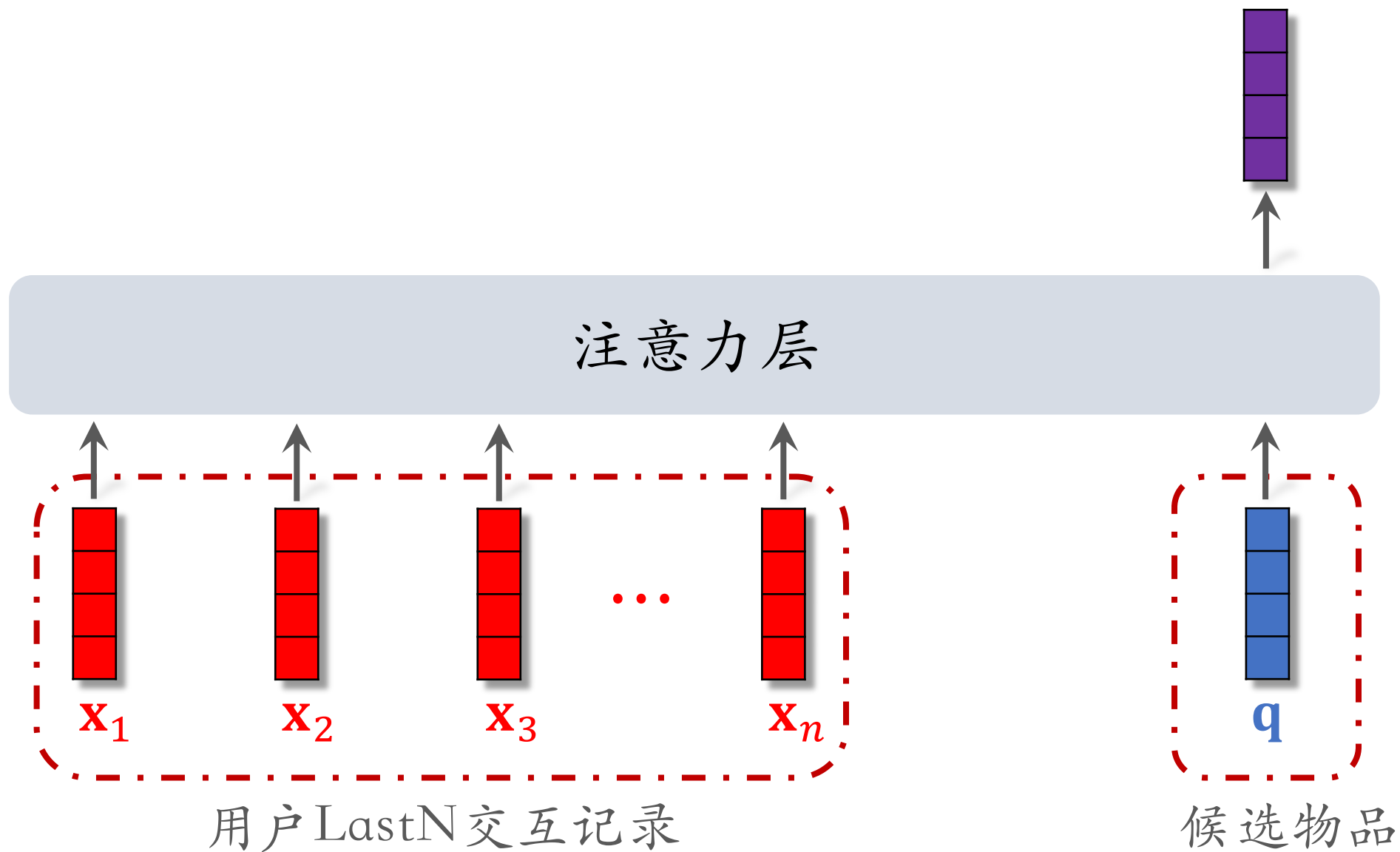


用户LastN交互记录



候选物品

DIN模型



DIN模型

- 计算用户 LastN 向量的加权平均。
- 权重是候选物品与 LastN 物品的相似度。

参考文献：

- Zhou et al. [Deep interest network for click-through rate prediction](#). In *KDD*, 2018.

DIN模型的缺点

- 注意力层的计算量 $\propto n$ （用户行为序列的长度）。
- 只能记录最近几百个物品，否则计算量太大。
- 缺点：关注短期兴趣，遗忘长期兴趣。

参考文献：

- Zhou et al. [Deep interest network for click-through rate prediction](#). In *KDD*, 2018.

如何改进DIN?

- **目标**：保留用户长期行为序列 (n 很大)，而且计算量不会过大。
- **改进 DIN**：
 - DIN 对 LastN 向量做加权平均，权重是相似度。
 - 如果某 LastN 物品与候选物品差异很大，则权重接近零。
 - 快速排除掉与候选物品无关的 LastN 物品，降低注意力层的计算量。

SIM模型

- 保留用户长期行为记录， n 的大小可以是几千。
- 对于每个候选物品，在用户 LastN 记录中做快速查找，找到 k 个相似物品。
- 把 LastN 变成 TopK，然后输入到注意力层。
- SIM 模型减小计算量（从 n 降到 k ）。

参考文献：

- Qi et al. Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. In *CIKM*, 2020.

第一步： 查找

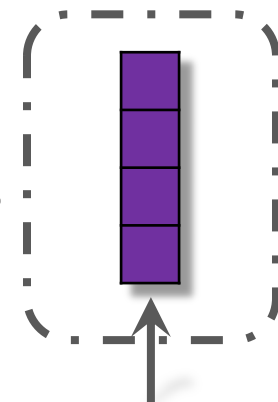
- 方法一：Hard Search
 - 根据候选物品的类目，保留 LastN 物品中类目相同的。
 - 简单，快速，无需训练。

第一步：查找

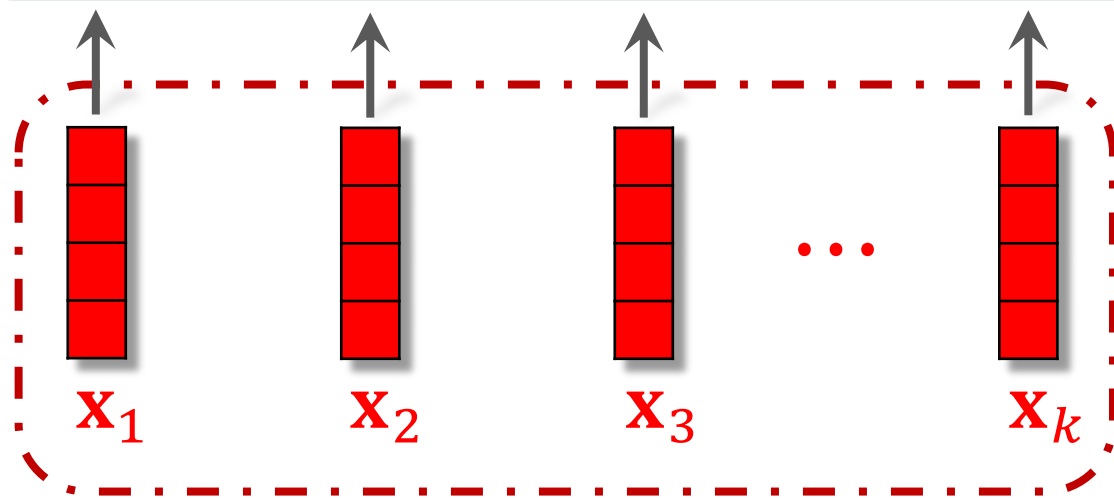
- 方法一：Hard Search
 - 根据候选物品的类目，保留 LastN 物品中类目相同的。
 - 简单，快速，无需训练。
- 方法二：Soft Search
 - 把物品做 embedding，变成向量。
 - 把候选物品向量作为query，做 k 近邻查找，保留 LastN 物品中最接近的 k 个。
 - 效果更好，编程实现更复杂。

第二步：注意力机制

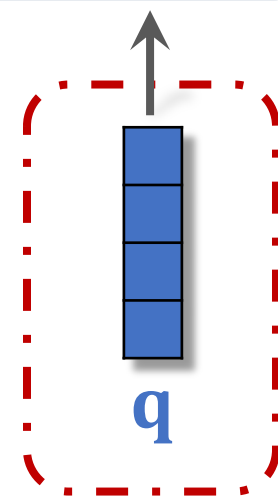
作为用户行为特征，输入模型，预估点击率等指标。



注意力层



用户TopK交互记录



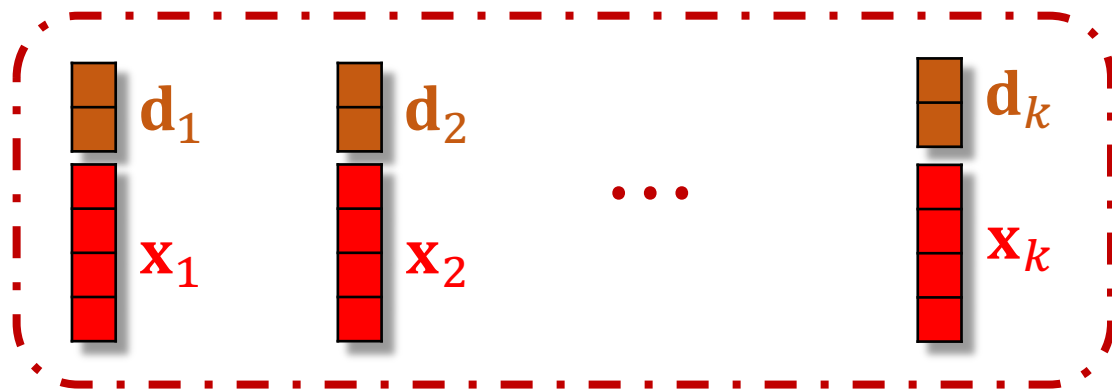
候选物品

第二步： 注意力机制

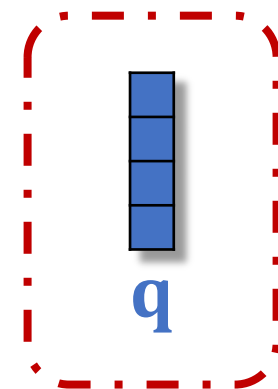
使用时间信息

- 用户与某个 LastN 物品的交互时刻距今为 δ 。
- 对 δ 做离散化，再做 embedding，变成向量 \mathbf{d} 。
- 把两个向量做 concatenation，表征一个 LastN 物品。
 - 向量 \mathbf{x} 是物品 embedding。
 - 向量 \mathbf{d} 是时间的 embedding。

第二步：注意力机制

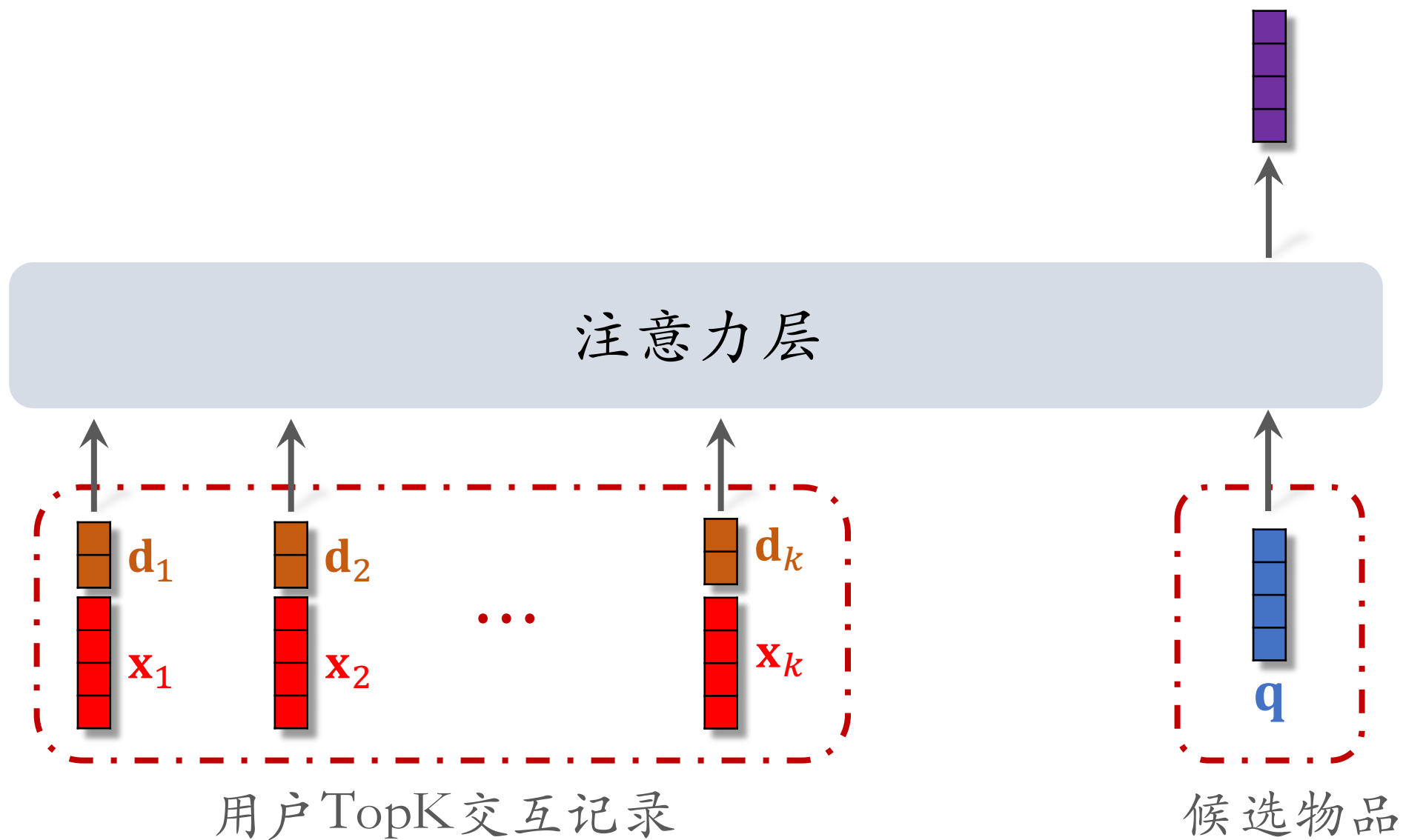


用户TopK交互记录



候选物品

第二步：注意力机制



第二步：注意力机制

为什么 SIM 使用时间信息？

- DIN 的序列短，记录用户近期行为。
- SIM 的序列长，记录用户长期行为。
- 时间越久远，重要性越低。

结论

- 长序列（长期兴趣）优于短序列（近期兴趣）。
- 注意力机制优于简单平均。
- Soft search 还是 hard search ? 取决于工程基建。
- 使用时间信息有提升。

Thank You!

<http://wangshusen.github.io/>