ATRank: An Attention-based User Behavior Modeling Framework for Recommendation

Accepted by AAAI 2018，我觉得创意不错。arXiv链接 。

摘要

在推荐系统或是社交网络中的用户建模中，可以通过他一系列的行为事件来建模。这些行为可以通过特征表示学习得到，把这些不同事件聚合起来，就能得到一个用户的表示。在人类有限的本能中，尚不明确这些融合方式的机理，因此很难全面地抓住用户的个性化特征，即这些特征向量之间的运算没有直观的解释。近期的研究中，开始用 RNN 模型来解决序列化推荐和预测，可仍然局限于这些有限的数据中，每一个事件都被通俗地编码成一个特征，我们认为只是简单的信息加和，没有利用信息之间的更新、推理等更为丰富的特征处理。例如，在当前事件的预测上，可能会失去和本次行为十分相关的历史行为的关系，而引入与当前事件没什么关系的其他噪音事件信息。

本文提出一种基于 Attention 机制的 ATRank 模型。用户的异构行为信息被映射到多个隐式语义空间中，由 self-attention，选出对当前事件的决策最具有影响的一些行为信息，从而在 Attention 时得以区别对待。ATRank 在实验中显示，训练更快，效果更好。此外，ATRank 还能对行为的类别进行预测，也比一些相关的优化模型表现好。

简介

正如一个词可以通过它的上下文得到一个表示，用户也可以根据他一系列的行为特征。**本文的任务，下面买什么（Downstream Applications Task）就像推荐系统中的一个排序任务**，过去由人标记的特征，生硬地构造一个特征空间，（注：这里的特征向量每一个或者每几个维度具有显式的意义），结合简单的机器学习方法，转化为一个分类问题来预测买（看）或不买（不看）。同时，在用户的行为序列中，每一个行为对当前的决策都具有不同程度的影响，这种简单的、基于人类本能的特征融合方式（根据我看到的一些文献研究中，主要有 sum、average、max pooling 或者 MLP 之类的特征压缩），仍然不足以刻画出全面的用户特征，难以区分出那些对当前对策有重大影响的特征（行为）。

用户的行为从时间线上天然构成了一个序列，RNN 、CNN 等模型结构，常被用来特征抽取。为了解决长距离依赖的问题，还有 LSTM、GRU 等更为精细的变体。RNN 模型还存在的一个不足，就是难以并行训练，模型的学习和测试的时间开销很长。有相关文献和实验表明，CNN 模型也能在序列化处理任务上取得不俗的效果，可以并行，但是在序列编码过程中，最长的可以依赖的距离是$\log_k(n)$ 级别的，其中 n 是序列长度，k 是卷积核的宽度（width，size）。

另一个问题是，在训练时，短序列通过补 0 等 padding 操作扩充到最大长度，RNN 或 CNN 的输入都是固定长度的，就存在对于长短序列的 Trade-off。在本文的 downstream applications 任务中，当前待排序的物品可能之和一小部分行为有比较大的关系。因此引入 Attention 机制，可以使编码器根据实际序列的情况，动态得到特征向量。简单 Attention 可表示为：$C = \sum_{i=1}^{n} a_i \vec{v}_i$，$a_i$ 是一个与 i 有关的标量，向量 $\vec{v}_i$ 的每一维都被乘上一个相同的数，这种通俗的加权平均处理方法，难以扩大 $\vec{v}_i$ 中与当前预测任务有关的隐藏语义（某几维）或者消除那些不太相关的隐藏语义特征。

第三，用户的行为数据是异构的，这些事件的类型是不同的，浏览、点击、购买，还有是否使用了优惠券、搜索关键词、评论等多元的数据，每一种信息都刻画了用户某方面的特征，这种特征对于用户比较完全的建模显然是具有正向增益的。异构数据的表示学习，可以通过最小化他们在一个隐藏语义空间的距离来实现，但缺少一个显式的监督来指导每一个特征的学习。

本文提出一种基于 Attention 的用户行为模型，来解决推荐任务，可分为三个步骤：

- 首先，把不一样长的用户交互行为记录映射到多个具有某种共性的语义空间中；
- 其次，通过 self-attention 方式，更关注和当前预测行为相关的历史行为，构建出一个更好的特征向量；
- 最后，把特征向量和 Attention 向量输入进一个排序神经网络，得到预测结果。

此外，本文通过实验验证了使用 self-attention 加上时间特征的 embedding 可以用来代替复杂的 RNN/CNN 模型，在时间角度的序列化推荐和预测的任务上，具有更快的训练学习速度和更好的性能。

相关工作

基于内容的推荐

Context Aware，即在推荐系统中，用户具有多种属性（特征），如性别、收入、购买能力、类别偏好等，这些特征的融合即为一个用户的表示。在 RNN 相关的模型中，把特征做成离散或连续的向量的输入，是较为直接且被接受的做法。可具有两个难点：

- RNN 模型的训练和预测需要输入一个序列，无法并行，在决策阶段的输出比较慢，在实际推荐系统应用中遭受诟病；
- RNN 模型的序列输入特征是固定长度（维数）的，需要对各种特征进行融合，很容易损失一些有用的信息或者引入噪音

Attention 和 Self-Attention

Attention 首先在 Seq2Seq 中提出，在机器翻译中，可以得到更为精确的序列对齐，因为它保留了编码器部分输入序列每个步骤的输出，在解码阶段都可以被使用。在其他 NLP 相关的任务，如阅读理解、广告推荐等也带来模型性能的提升。Self-attention 则更为复杂，在编码阶段还考虑到了序列中每个输入的内在联系。（个人理解：Attention 从向量变为矩阵？）相关研究显示，这样做可以把输入序列中的每个词映射到多个不同的语义空间中，在多任务模型中可以优化每一个任务相关的特征语义空间。

异构数据建模

异构数据表示学习在知识图谱相关的工作中已有许多研究，主要是通过最小化关系-类别语义子空间的特征向量距离学习特征表示。

本文模型

用户的行为抽象为一个 tuple：$\langle a, o, t \rangle$ ，a 是交互事件的类型，o 是交互的对象，t 是时间戳，其中 o 是包含了多元异构数据的特征的。从而，一个用户就可以通过 tuple 的序列来表示。模型总体如下，可以分为五个部分：

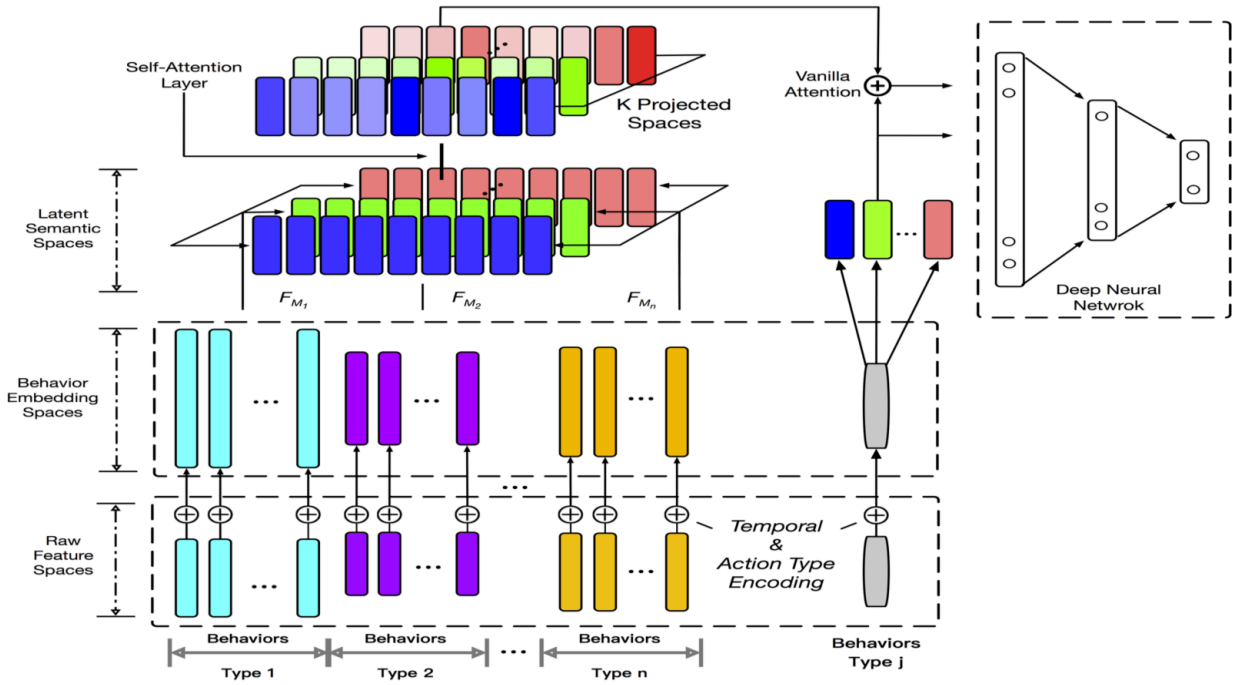

Figure 1: Attention-Based Heterogeneous Behaviors Modeling Framework

原始特征空间

将用户的行为 tuple 根据类别分组，每个组中有若干个 o 作为特征。注意，这里每个组（group）内，对于同一个物品的交互信息是相同的，包含了离散或是连续的特征。通过类型相关的 encoding，即每个组分别对应一个网络，映射到行为特征空间。

行为特征空间

这里得到用户 i 在交互过物品 j 的事件特征 $u_{ij} = f_i(a_j, o_j, t_j)$，$o_j$ 的特征向量是保留在最终的拼接向量中的，对于事件类型和时间信息也分别编码，一起拼接。考虑到每一个行为都有一个时间戳，那么可以不使用 RNN 的序列化输入结构，只要合理利用这时间特征即可。对于时间戳的编码，采用一种指数栅栏式的划分：$[0,1)$, $[1,2)$, $[2,4)$, $\cdots$ $[2^k, 2^{k+1})$, $\cdots$，每一个时间区间被赋予类别标签 k。对于事件类型的编码，也是直接采用一种查表的类型离散化标签方式。在我们得到 u_{ij} 后，对于每个类别组的表示是简单的拼接，即包含了组内所有的 o_j。最终的行为特征空间 $B = \{u_{g_1}, u_{g_2}, \cdots, u_{g_n}\}$，这里的 u_{g_i} 即为第 i 组所有的 u 向量的列表。

需要注意的是，每一个 u_{g_i} 中的特征数都是不一样多的，因为每一个类别的事件数不同，并且这些 o_j 特征的维数也不一定相同。当然，有些特征是共享的，如物品所在商铺的信息，蕴含其中的同一家商铺的特征是相同的。这里的时间特征 embedding 并不是共享的，因为每一种类别的事件对时间的敏感程度不同。（问题：时间的 embedding 之前所说的标签化方法得到的时间特征是一样的，这里又说不是共享的。）

潜在语义空间

多元异构数据的表示，他们的隐含空间应该具有不同的语义和维度。使用一个线性映射把异构的特征映射到同一空间中，这样做的好处还有，可以把这些不同视角下的特征建立某种联系或者比较。本文的做法是，将每一种行为映射到 K 种潜在语义空间中去，如模型图中的三个颜色“通道”所示。

具体来说，在一个通道（潜在语义空间）内，将每一组的特征 u_{g_i} （它是一个组内所有事件的 embedding 拼接得到的列表）通过一个线性映射 $\mathcal{F}_{M_i}$ 得到一个固定维数的特征，每一组都拼接在一个，构成一个完整的该通道内的语义总特征。经过 K 个不同通道的映射，得到了 K 个不同语义空间下的特征 S_k，每一个特征的规模是该用户的事件总数 x 某个语义空间的维数。

Self-Attention 层

在机器翻译中，率先使用这种机制【参考：Attention is all you need 】。这一层的作用是，表现出每个语义空间之间的内在联系。因为用户的每一个行为意图都会与其他的行为有关，Attention 的效果就在于可以增强相关事件的信息、消除不相关事件的影响。在每一个语义空间中计算 Attention 矩阵：A_k ，从而得到语义空间 k 下的 Attention 向量 C_k，也是一个用户的事件总数 x 语义空间 k 的维数。 把所有的 C_k 按照语义空间的视角拼接起来，就能得到一个 self-attention 特征：$C = \mathscr{F}_{self}(concat^{(1)}(C_1, C_2, \cdots, C_K))$ ，这里的 $\mathscr{F}$ 是一个具有一个隐藏层的前向网络，将 Attention作用后的特征进行非线性的、维数不变的映射。

下游应用网络

通过之前的用户行为模型得到的用户特征，就可以集成多种神经网络模型进行预测。本文选择的是推荐任务，即预测给定的事件 q_t 是否会发生的二分类问题。如模型图的右侧，前半部分的操作与模型学习过程相似，根据 q_t 的类型，映射到对应组的行为隐层空间：$h_t = \mathcal{F}_{M_{g(t)}}(q_t)$，再分别计算 K 个语义空间下的特征表示，最后通过 self-attention 整合，得到用户对于该事件的一个特征向量 e_u^t，而 e_u^t 和 h_t 的交叉熵损失即为模型优化的目标。（注：这里的 e_u^t 相当于一个用户的特征，h_t 则是对应物品的特征，它们之间做点积或其他运算，即可得到一个概率，就可以根据监督信息计算 cross entropy loss）

实验与分析

本文的实验分为两部分，首先采用同一种事件类型的设置，为的是测试 self-attention 的作用；第二是包含多种行为类型的数据，测试多元异构数据特征融合的有效性。还有一个实验是多任务测试，该模型框架还可以用来预测事件的类型。

数据集与实验设置

- Amazon 数据集的子集，使用的信息包括用户、商品及商品种类的 id 以及时间戳，这是用来预测同一事件类型（购买）下的待推荐物品。用户的购买记录，$b_1, b_2, \cdots, b_k, \cdots, b_n$，在训练集中使用前 $k-1$ 个记录来预测第 k 个物品（$k = \{1, 2, \cdots, n-2\}$），测试集则是用前 $n-1$ 条记录预测最后一个事件。

Dataset	# Users	# Items	# Cates	# Samples
Electro.	192,403	63,001	801	1,689,188
Clothing.	39,387	23,033	484	278,677

Table 1: Statistics of Amazon DataSets

- Taobao 数据集，选择了三组事件：商品操作组（ item，如购买、浏览、收藏等多种具体类型）；搜索组（ query，仅包含搜索的关键词文本特征，只有一种类型）、折扣券组（ coupon，用户接收到的抵价券、红包等信息，也只有这一种类型）

Dataset	#Users	#Items	#Cates	#Shops	#Brands	#Queries	#Coupons	#Records	Avg Length
Multi-Behavior.	30,358	447,878	4,704	109,665	49,859	111,646	64,388	247,313	19.8

Table 2: Statistics of Multi-Behavior DataSets

对比实验

- BRP-MF，针对正负样本最大化差异计算损失；
- Bi-LSTM，使用简单的双向 LSTM 模型，对用户的行为序列直接建模，得到用户的特征表示；
- Bi-LSTM + attention，在输出层使用 vanilla attention 得到最终的用户表示；
- CNN-Pooling，类似文本分类的 CNN 模型，把事件序列视作文本，用多个卷积核操作得到多维用户特征表示。

实验中的超参设置都比较常规，是经验做法，不具体介绍了。

实验的评价标准是 AUC，定义如下：

$$AUC = \frac{1}{|U^{Test}|} \sum_{u \in U^{Test}} \frac{1}{|I_u^+| \cdot |I_u^-|} \sum_{i \in I_u^+} \sum_{j \in I_u^-} \delta(\hat{p}_{u,i} > \hat{p}_{u,j})$$

该指标表示的是随机选择的正样本比随机选择的负样本具有更高 rank 的概率，越高证明模型越好。

实验结果

1. 单种事件类型

Dataset	Electro.	Clothe.
BPR	0.7982	0.7061
Bi-LSTM	0.8757	0.7869
Bi-LSTM + Attention	0.8769	0.7835
CNN + Max Pooling	0.8804	0.7786
ATRank	**0.8921**	**0.7905**

Table 3: AUC on Amazon Dataset

ATRank 模型性能得到提升的同事，收敛速度也要比其他方法快。

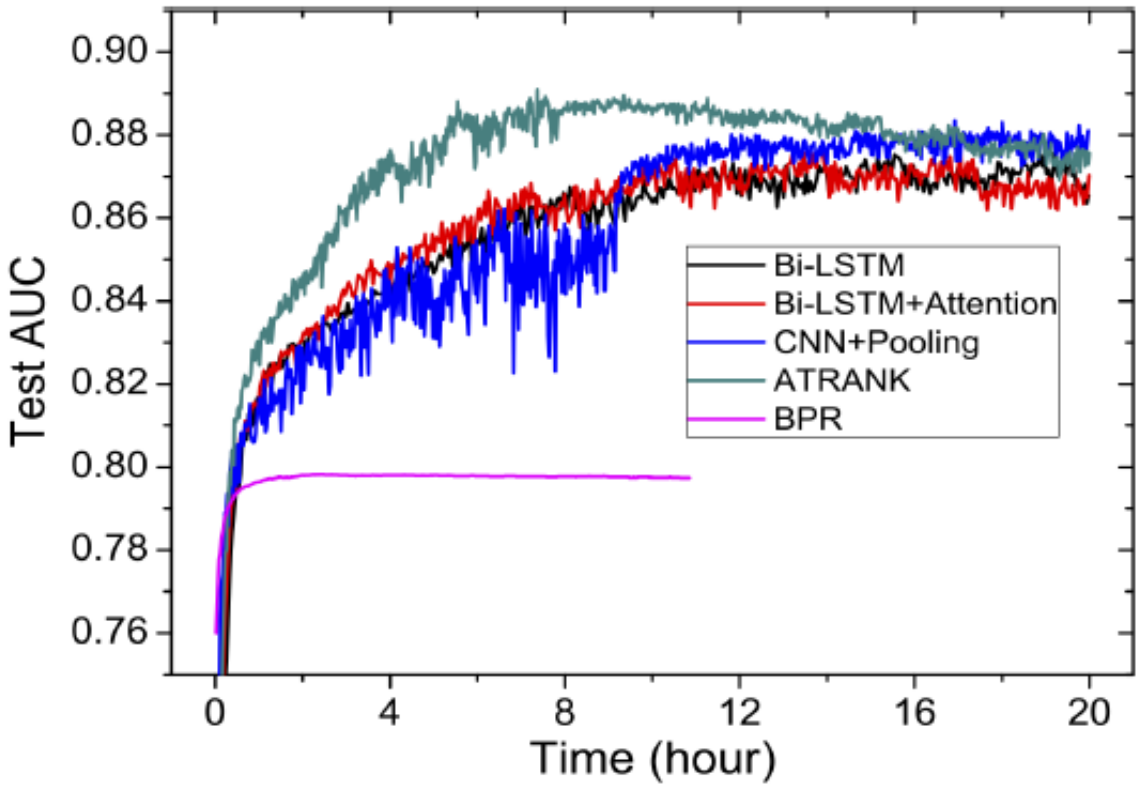

Figure 2: AUC Progress in Amazon Electro Test Set

从 Attention Score 上来分析，发现每个不同时间段内的其他事件对当前事件的影响也是有差异的。

Time Range(#Day)	[0, 2)	[2, 4)	[4, 8)	[8, 16)	[16, 32)	[32, 64)	[64, 128)
Avg Att-Score	0.2494	0.1655	0.1737	0.1770	0.1584	0.1259	0.1188

Table 4: Average Attention Score for Different Time Bucket over Amazon Dataset

2. 多种类型事件实验

   在这里又细分为三种子任务：

   - One2one，只用一种类型的事件来训练和预测，即 item2item、coupon2coupon、query2query
   - All2one，三种类型的事件放在一起作为训练集，但独立测试预测每一种事件，all2item、all2coupon、all2query；
   - All2all，这是一个多任务模型，把所有的事件放在一起作为训练集，得到一个统一的模型，最终还会预测下一个事件的类型，与 All2one 相比，是不知道预测事件的类型的。

   再提一下负样本的生成，第一种类型，和商品有关的，可以从实际网站本身提供的推荐数据中直接获取用户没有交互的项目作为负样本；第二种和第三种则是随机生成的负样本。最终 ATRank 的表现最好。

Predict Target	Item	Query	Coupon
Bi-LSTM	0.6779	0.6019	0.8500
Bi-LSTM + Attention	0.6754	0.5999	0.8413
CNN + Max Pooling	0.6762	0.6100	0.8611
ATRank-one2one	0.6785	0.6132	0.8601
ATRank-all2one	**0.6825**	**0.6297**	**0.8725**
ATRank-all2all	0.6759	0.6199	0.8587

Table 5: AUC on Ali Multi Behavior Dataset

Case Study

为了说明 ATRank 中各个模块的作用，通过一个实例来证明。一名女性用户先后买了一个包、一枚戒指、一顶男士帽子、五件首饰、一条裤子、一双长靴，待预测的物品是一件连帽衫，通过 Attention 热力图可视化，可以看到裤子的权重会高一些，因为一般来说裤子和衣服的搭配比较相关。

(a) Average Vanilla-Attention Score over All Latent Space

另一方面，通过在多个语义空间的特征分布，我们可以找到某些物品的相似性，如下图的第六个（VI）self-attention 热力图中，可以看到第二个物品（戒指）和其他的五种首饰的特征分布十分相似，因此可以判断它们属于同一个类别。可见，self-attention 结合多个潜在语义空间，能够刻画出物品之间的某些内在联系。

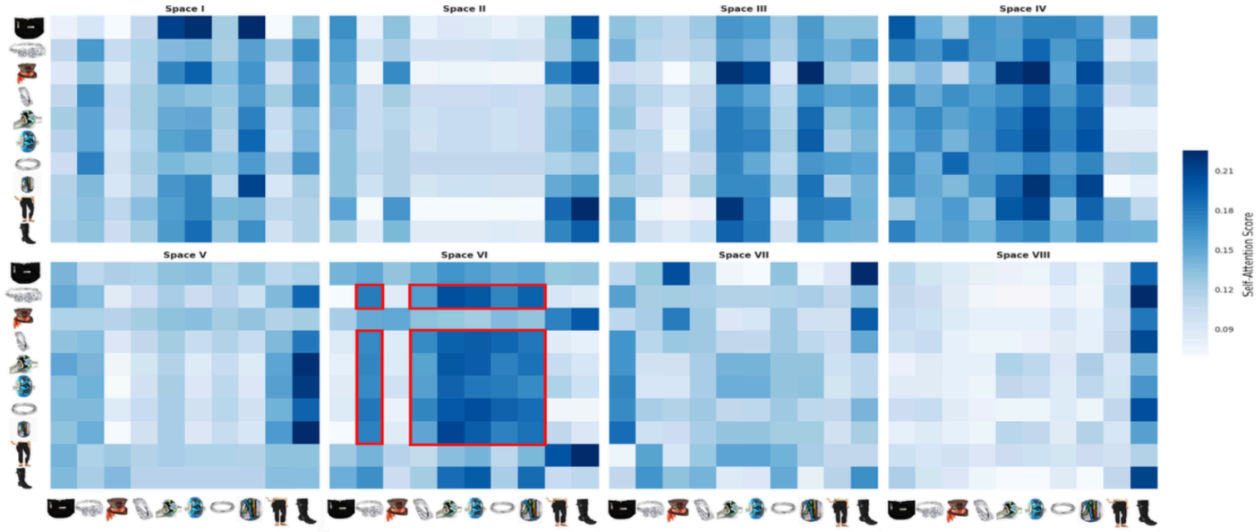

Figure 3: Case Study: Heatmap of Self-Attention in ATRank

结论

本文提出的 ATRank 模型，在序列化推荐和预测任务上，取得了不俗的效果。主要原因在于能够融合用户的多种异构数据，并且通过 Attention 机制尽可能扩大更为相关的数据信息。分散在多个语义空间的多视角下的特征，也可以帮助我们从 Multi-task 角度解释这样做的优势和道理。

本文是阿里和北大合作研究的文章，也比较注重实用性，抛弃了耗时的 RNN 框架，改为时间戳编码在特征中，基本可以取代序列化信息（我认为有点像 TimeSVD++ 的方式）。

现在关于序列化推荐，已经有一些研究了，主流还是利用 RNN 处理序列，加上简单的 Attention 是大家都能想到的改进，理论原因也是因为过去的历史记录有的和当前物品正相关，有的无关，需要通过 Attention 加以甄别，而不能一视同仁。本文中的做法，分散到多个语义空间，又通过 self-attention 进行特征的融合，其中多处使用向量的拼接，有些生硬的感觉。关于多元数据特征的学习，应该还有改进的地方，可以参考 Philip Yu 的 Broad Learning 相关研究。