# AGREE

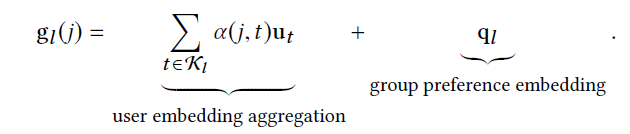
我们认为这些预定义的策略是与数据无关的，缺乏根据组成员动态调整权重的灵活性。当一个组对不同类型的项目做出决策时，这种灵活性尤为有用。受到最近神经注意力机制[2, 36]的发展启发，该机制可以从数据中学习不同模型组件的重要性，我们考虑使用注意力来学习聚合策略。其基本思想是在将一组表示压缩成一个表示时进行加权求和，其中权重由神经网络学习得到。我们发现这对于处理组推荐任务非常合适，更重要的是，现有的聚合策略可以被视为其特例。具体而言，平均值等同于给所有成员分配均匀的权重，最小痛苦和最大满意度对应于只给部分成员分配非零权重，而专业知识可以被看作是基于成员的专业知识固定注意力权重。

我们提议在表示学习（RL）框架下解决组推荐问题。在RL范 paradigm ，每个感兴趣的实体都被表示为嵌入向量，该向量编码实体的固有属性（例如一个词的语义、一个用户的兴趣等），并且可以从数据中学习得到。在项目推荐中，著名的矩阵分解（MF）方法 [17, 40] 是一种典型的RL模型，它将每个用户和项目与一个嵌入向量相关联。

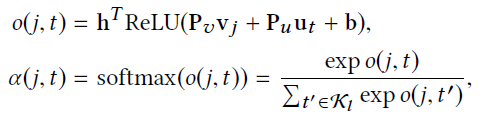
让 ui 和 vj 分别为用户 ui 和项目 vj 的嵌入向量，它们是我们 AGREE 模型中的基本表示块。我们的目标是为每个组获得一个嵌入向量，以估计其对一个项目的偏好。为了从数据中学习动态聚合策略，有必要将组嵌入定义为依赖于其成员用户和目标项目的嵌入向量，可以抽象为以下形式：



其中，gl(j) 表示针对预测组 дl 对目标项目 vj 的偏好而定制的组嵌入，Kl 包含组 дl 的用户索引，fa 是待指定的聚合函数。在 AGREE 模型中，我们将组嵌入设计为由两个组成部分组成 — 用户嵌入聚合和组偏好嵌入：



用户嵌入聚合。我们对组 дl 的成员用户的嵌入进行加权求和，其中系数 α(j, t) 是一个可学习的参数，表示成员用户 ut 对决定组对项目 vj 的选择的影响力。直观地说，如果一个用户在某个项目上有更多的专业知识（或者在类似类型的项目上有经验），她对组对该项目的选择应该有更大的影响力 [28]。为了理解这一点，让我们考虑一个例子，一个组讨论去哪个城市旅行；如果一个用户多次去过中国，那么当组考虑是否去中国的某个城市时，她应该更具影响力。由于在 RL 框架中，嵌入 ut 编码了成员用户的历史偏好，嵌入 vj 编码了目标项目的属性，我们将 α(j, t) 参数化为一个具有 ut 和 vj 作为输入的神经注意力网络：



其中，Pv 和 Pu 是注意力网络的权重矩阵，分别将项目嵌入和用户嵌入转换为隐藏层，b 是隐藏层的偏置向量。我们使用 ReLU 作为隐藏层的激活函数，然后使用权重向量 h 将其投影到得分 o(j, t)。最后，我们使用 softmax 函数对得分进行归一化，这是神经注意力网络中的常见做法 [2, 9, 36]；它使注意力网络具有概率解释，也可以处理我们这种情况下不同大小的组。

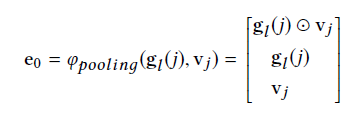
图2说明了我们设计的用户嵌入聚合组件。通过这种软注意力机制，我们允许每个成员用户对组的决策做出贡献，其中用户的贡献取决于她的历史偏好和目标项目的属性，这些都是从组-项目交互和用户-项目交互的过去数据中学习得到的（将在第2.3节中讨论）。

组偏好嵌入。除了聚合组成员的嵌入之外，我们进一步将组 дl 与一个专用的嵌入 ql 相关联。这是为了考虑到组的一般偏好。我们的考虑是，在某些情况下，当用户组成一个组时，他们可能追求与每个用户的偏好不同的目标。例如，在一个由三个人组成的家庭中，孩子喜欢卡通电影，父母喜欢浪漫电影；但当他们一起去电影院时，最终选择的电影可能是一部教育电影。因此，除了从成员中聚合得到的偏好之外，将一个组与一个嵌入相关联以表示其一般偏好是有益的。为了将组偏好嵌入的组成部分与用户嵌入聚合相结合，我们执行一个简单的加法操作，与之前的工作 [9, 37] 中将嵌入空间中的不同信号组合的方法相同。我们在第3.4节的实证结果表明，这个组件可以显著提高组推荐的性能。.

NCF（Neural Collaborative Filtering）是一种用于项目推荐的多层神经网络框架[16]。它的思想是将用户嵌入和项目嵌入输入到一个专用的神经网络（需要进行定制）中，从数据中学习交互函数。由于神经网络具有很强的数据拟合能力，NCF框架比传统的MF模型更具通用性，后者仅将一个与数据无关的内积函数作为交互函数。因此，我们选择NCF框架来对表示用户、项目和组的嵌入以及预测用户-项目和组-项目交互的交互函数进行端到端学习。

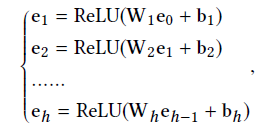
图3说明了我们定制的NCF解决方案。由于我们同时要实现对组和用户的推荐任务，我们设计了这个解决方案来同时学习用户-项目和组-项目的交互函数。具体而言，对于给定的用户-项目对（ui, vj）或组-项目对（дl, vj），表示层首先返回每个给定实体的嵌入向量（详细信息见第2.2节）。然后，将这些嵌入馈入到一个汇聚层和隐藏层（两个任务共享），以获得预测得分。接下来，我们详细说明这两个组成部分。

汇聚层。假设输入是一个组-项目对（дl, vj），汇聚层首先对它们的嵌入进行逐元素乘积，即 gl(j) ⊙ vj，并将其与原始嵌入连接起来：

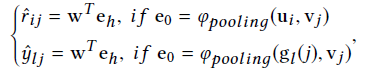


这样做的理由有两个。1）逐元素乘积包含了MF模型，它使用每个嵌入维度上的乘法来建模两个嵌入向量之间的交互；此外，逐元素乘积已被证明在低层神经架构的特征交互建模中非常有效[15]。2）然而，逐元素乘积可能会丢失一些原始嵌入中的信息，这些信息在后续的交互学习中可能很有用。为了避免这种信息损失，我们将其与原始嵌入连接起来。

需要注意的是，这样的汇聚操作部分受到了最先进的神经推荐模型NeuMF [16]的启发，该模型显示将MF与隐藏层中的MLP相结合可以获得更好的性能。由于MLP连接了原始的用户嵌入和项目嵌入，它启发我们保留原始的嵌入以促进后续隐藏层的学习。我们对（ui, vj）对的输入应用相同的汇聚操作。

共享的隐藏层。在汇聚层之上是一堆全连接层，它使模型能够捕捉用户、组和项目之间的非

在上述方程中，Wh、bh和eh分别表示第h层隐藏层的权重矩阵、偏置向量和输出神经元。我们使用ReLU函数作为非线性激活函数，经验证明它效果良好。此外，我们使用塔状结构的隐藏层，并将对结构的进一步调整作为未来的工作。最后，通过将最后一个隐藏层的输出eh转换为预测得分：



其中w表示预测层的权重；rˆi j和yˆl j分别表示用户-项目对（ui，vj）和组-项目对（дl，vj）的预测。

值得一提的是，我们有意设计了两个任务的预测共享相同的隐藏层。这是因为组嵌入是从用户嵌入聚合而来的，它们在本质上处于相同的语义空间中。此外，这可以通过用户-项目交互数据和组-项目交互数据增强组-项目交互函数的训练，并反之亦然，从而促进两个任务相互强化。

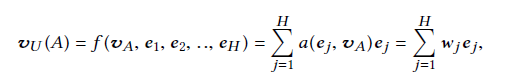
# DIN

在表1的所有特征中，用户行为特征至关重要，在电子商务应用场景中对建模用户兴趣起着关键作用。

基础模型通过对用户行为特征组中的所有嵌入向量进行汇集，得到用户兴趣的固定长度表示向量，如公式（1）所示。对于给定的用户，这个表示向量不受候选广告的影响而保持不变。通过这种方式，具有有限维度的用户表示向量将成为表达用户多样化兴趣的瓶颈。为了使其足够能力，一种简单的方法是扩展嵌入向量的维度，但不幸的是，这将大幅增加学习参数的大小。这将导致在有限的训练数据下出现过拟合，并增加计算和存储的负担，这对于工业在线系统可能是无法容忍的。

有没有一种优雅的方法来在有限维度下表示用户的多样化兴趣呢？用户兴趣的局部激活特征给我们设计一种名为深度兴趣网络（DIN）的新模型提供了灵感。想象一下，当上面在第3节中提到的年轻母亲访问电子商务网站时，她发现展示的新手提包很可爱，并点击了它。让我们来解剖点击行为的驱动力。展示的广告通过对她的历史行为进行软搜索并发现她最近浏览了类似的手提包和皮质手提包商品，命中了这位年轻母亲的相关兴趣。换句话说，与展示的广告相关的行为极大地促成了点击行为。DIN通过关注与给定广告相关的局部激活兴趣的表示来模拟这个过程。DIN不同于用同一个向量表示所有用户的多样化兴趣，而是通过考虑历史行为与候选广告相关性来自适应地计算用户兴趣的表示向量。这个表示向量在不同广告之间是变化的。

图2的右侧说明了DIN的架构。与基础模型相比，DIN引入了一个新设计的局部激活单元，并保持其他结构不变。具体而言，激活单元应用于用户行为特征，其作用是对用户行为特征进行加权汇总，以自适应地计算给定候选广告A时用户表示向量vU，如公式（3）所示。



其中 fe1; e2; :::; eH g 是用户U行为的嵌入向量列表，长度为H，vA 是广告A的嵌入向量。

通过这种方式，vU（A）在不同广告之间变化。a(·) 是一个前馈网络，其输出为激活权重，如图2所示。

除了两个输入嵌入向量之外，a(·) 将它们的外积相加作为输入传递给后续网络，这是一种显式的知识，用于帮助相关性建模。

公式（3）中的局部激活单元与在NMT任务中开发的注意力方法[1]具有类似的思想。然而，与传统的注意力方法不同，公式（3）中放松了P

i wi = 1 的约束，旨在保留用户兴趣的强度。也就是说，放弃对a(·)输出进行softmax归一化的规范化。相反，P

i wi 的值被视为激活用户兴趣强度的近似程度。例如，如果一个用户的历史行为中有90%的衣服和10%的电子产品。给定两个候选广告，一个是T恤，一个是手机，T恤激活了大部分属于衣服的历史行为，可能得到比手机更大的vU值（更高的兴趣强度）。传统的注意力方法通过对a(·)的输出进行归一化处理，在vU的数值尺度上失去了分辨率。

我们尝试过使用LSTM以顺序方式建模用户历史行为数据，但没有改进。与NLP任务中受语法约束的文本不同，用户历史行为的顺序可能包含多个并发的兴趣。在这些兴趣之间的快速跳转和突然结束导致用户行为序列数据看起来杂乱无章。一个可能的方向是设计特殊结构以顺序方式建模这种数据。我们将这留给未来的研究。

# MIND

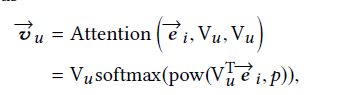
我们认为，通过一个表示向量来表示用户兴趣可能成为捕捉用户多样化兴趣的瓶颈，因为我们必须将与用户多样化兴趣相关的所有信息压缩到一个表示向量中。因此，有关用户多样化兴趣的所有信息被混合在一起，导致匹配阶段的项目检索不准确。相反，我们采用多个表示向量来分别表示用户的不同兴趣。通过这种方式，在匹配阶段将用户的多样化兴趣分别考虑，从而能够更准确地对每个兴趣方面进行项目检索。

为了学习多个表示向量，我们利用聚类过程将用户的历史行为分组成几个簇。同一簇中的项目预计密切相关，并集体表示用户兴趣的特定方面。在这里，我们为聚类历史行为和推断每个簇的表示向量设计了多兴趣提取器层。

由于多兴趣提取器层的设计受到最近提出的胶囊网络中用于表示学习的动态路由的启发[11, 12, 20]，我们首先回顾了基本原理，以使本文自成一体。

我们简要介绍一下动态路由[20]用于胶囊表示学习，胶囊是一种由向量表示的新型神经单元。假设我们有两层胶囊，我们将第一层的胶囊和第二层的胶囊分别称为低级胶囊和高级胶囊。动态路由的目标是以迭代的方式计算高级胶囊的值，给定低级胶囊的值。

通过多兴趣提取器层，从用户的行为嵌入中生成了几个兴趣胶囊。不同的兴趣胶囊代表用户兴趣的不同方面，相关的兴趣胶囊用于评估用户对特定项目的偏好。因此，在训练过程中，我们基于缩放点积注意力[23]设计了一个具有标签感知的注意力层，使目标项目选择使用哪个兴趣胶囊。具体而言，对于一个目标项目，我们计算每个兴趣胶囊与目标项目嵌入之间的兼容性，并将兴趣胶囊的加权和作为用户在目标项目上的表示向量，其中每个兴趣胶囊的权重由相应的兼容性确定。在标签感知注意力中，标签是查询，而兴趣胶囊既是键又是值，如图2所示。用户u关于项目i的输出向量计算如下：



其中pow表示逐元素的指数运算符，p是一个可调参数，用于调整注意力分布。当p接近0时，每个兴趣胶囊都会接收到相同的注意力。当p大于1时，随着p的增加，具有更大点积的值将获得越来越多的权重。考虑到极限情况，当p趋近无穷大时，注意机制变成了一种硬注意力，选择具有最大注意力的值并忽略其他值。在我们的实验中，我们发现使用硬注意力可以加快收敛速度。

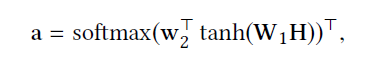
# ComiRec

由于工业推荐系统的项目池通常包含数百万甚至数十亿个项目，匹配阶段在推荐系统中起着至关重要的作用。具体而言，匹配模型首先根据用户的历史行为计算用户嵌入向量，然后基于用户嵌入向量为每个用户检索候选项目集合。我们利用快速K最近邻（KNN）算法从大规模项目池中选择最相似的项目，为每个用户生成候选集合，因此我们主要关注用户嵌入向量的计算。换句话说，匹配阶段的决定性因素是从用户的历史行为计算出的用户嵌入向量的质量。

现有的匹配模型通常使用RNN[21, 54]为用户计算嵌入向量，但其中大部分只为每个用户生成一个单一的嵌入向量。这导致单一嵌入的表达能力不足，因为现实中的用户通常在心中有多种类型的项目，这些项目通常用于不同的用途，并且在类别上有很大的差异。这些现实世界用户的行为强调了使用多个向量来表示他们的多个兴趣的必要性。基于这些观察，我们提出了一个用于顺序推荐的多兴趣框架。我们的框架的输入是用户的行为序列，其中包含一个按时间顺序排列的项目ID列表，表示用户与项目的交互。项目ID被馈送到嵌入层，并转换为项目嵌入。多兴趣提取模块接收项目嵌入并为每个用户生成多个兴趣。

构建多兴趣提取模块有许多可选方法。在本文中，我们探索了两种方法，即动态路由方法和自注意方法，作为我们的多兴趣提取模块。我们使用动态路由方法或自注意方法的框架分别被称为ComiRec-DR或ComiRec-SA。

自注意方法也可以应用于我们的多兴趣提取模块。给定用户行为的嵌入H ∈ Rd×n，其中n是用户序列的长度，我们使用自注意机制获取权重向量a ∈ Rn：



其中w2和W1是可训练参数，大小分别为da和da×d。上标⊤表示向量或矩阵的转置。大小为n的向量a表示用户行为的注意权重。根据注意权重对用户行为的嵌入进行求和，我们可以得到表示用户的向量表示vu = Ha。为了使自注意方法利用用户序列的顺序，我们将可训练的位置嵌入[51]添加到输入嵌入中。位置嵌入与项目嵌入具有相同的维度d，并且可以直接求和。

这个向量表示聚焦于并反映了用户u的特定兴趣。为了表示用户的整体兴趣，我们需要从用户行为中获得聚焦于不同兴趣的多个vu。因此，我们需要多次进行注意力计算。我们将w2扩展为一个大小为da × K的矩阵W2。然后，注意力向量a变为注意力矩阵A：



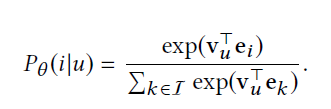
最终的用户兴趣矩阵Vu可以通过以下方式计算得到：



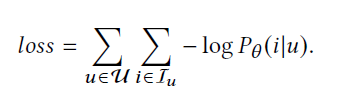
**模型训练**。通过多兴趣提取模块从用户行为中计算出兴趣嵌入后，我们使用argmax运算符为目标项i选择相应的用户嵌入向量：



其中ei表示目标项i的嵌入，Vu是由用户兴趣嵌入形成的矩阵。

给定训练样本(u, i)，其中包括用户嵌入vu和项目嵌入ei，我们可以计算用户u与项目i交互的可能性，即。其中，I表示与用户u相关的项目集合。

我们的模型的目标函数是最小化以下负对数似然损失：



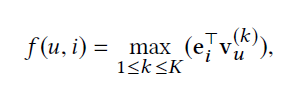
式中，U表示用户集合，Iu表示用户u的交互项目集合。

方程（9）中的求和运算在计算上是昂贵的，因此我们使用采样softmax技术[9, 24]来训练模型。

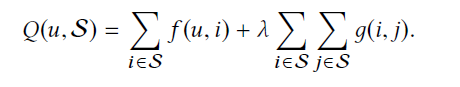
**在线服务**。对于在线服务，我们使用多兴趣提取模块为每个用户计算多个兴趣。用户的每个兴趣向量可以独立地从大规模项目池中通过最近邻库（如Faiss）检索出前N个项目。由多个兴趣检索到的项目被输入到聚合模块中，以确定整体的候选项。最终，具有较高排名得分的项目将被推荐给用户。

**聚合模块**

在多兴趣提取模块之后，我们根据用户的历史行为获得了多个兴趣嵌入。每个兴趣嵌入可以独立地基于内积相似度检索出前N个项目。但是，如何将来自不同兴趣的项目聚合起来获得整体的前N个项目呢？一种基本且直接的方法是根据项目与用户兴趣之间的内积相似度合并和过滤项目，可以表示为：



其中v(k)u是用户u的第k个兴趣嵌入。这是一种用于最大化推荐准确性的聚合方法。然而，当前推荐系统并不仅仅关注准确性。人们更倾向于推荐新颖或多样化的内容。这个问题可以形式化描述为：给定一个由用户u的K个兴趣检索得到的K·N个项目的集合M，找到一个包含N个项目的集合S，使得预定义的值函数最大化。我们的框架使用可控的过程来解决这个问题。我们使用以下值函数Q(u, S)来通过可控因子λ ≥ 0平衡推荐的准确性和多样性：



这里δ(i, j)是一个多样性或不相似性函数，例如δ(i, j) = δ(CATE(i), CATE(j))，其中CATE(i)表示项目i的类别，δ(·)是一个指示函数。对于准确性最高的情况，即λ = 0，我们只需使用上述直接的方法来获得整体项目。对于多样性最高的情况，即λ = ∞，可控模块将为用户找到最多样化的项目。我们在4.3节中研究了可控因子。我们提出了一种贪婪推理算法来近似最大化值函数Q(u, S)，算法2中列出了具体步骤。

# MVKE

基本解决方案

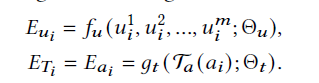
在实践中，我们分别构建两个集合T𝑉

𝑢 (𝑢)和T𝐶

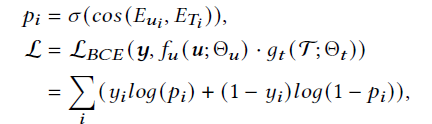
𝑢 (𝑢)，然后将它们合并到用户配置文件T𝑢 (𝑢)中。

基本上，有两个模型：兴趣模型和意图模型。前者是为用户标注兴趣标签，而后者是为意图标签进行标注。

类似于广告中的预测模型，兴趣模型用于预测<用户，标签>对的点击率（Click-through Rate），而意图模型用于预测其转化率（Conversion Rate）。如图3所示，该模型包含两个主要部分：用户塔𝑓𝑢(·)和标签塔𝑔𝑡(·)。给定用于训练的点击或转化对<𝑢𝑖, 𝑎𝑖>，用户塔接收用户的各种特征，例如基本属性（如年龄、性别）和历史行为特征（如阅读、购物）。借鉴来自在线预测模型（例如DeepFM）的设计，模型中还设计了多个特征字段，用于处理不同类型的特征输入。此外，标签塔接收广告标签集合T𝑎(𝑎𝑖)，并且所有标签都被输入到同一个特征字段中。最后，每个塔用于生成一个向量：用户嵌入𝐸𝑢和标签嵌入𝐸𝑇。形式上，我们有：



模型的估计分数由两个嵌入的余弦相似度计算得出，并且损失函数L是由标签𝑦𝑖监督的二元交叉熵损失，取决于用户𝑢𝑖是否点击或转化广告项𝑎𝑖。形式上，我们有：



其中𝜎是Sigmoid激活函数，𝑦是标签，𝑢是用户特征输入，T是标签输入，Θ是模型参数。当预测<𝑢𝑖, 𝑡𝑗>的点击率时，用户塔保持不变，但标签塔只输入一个单一的标签𝑡𝑗，表示为

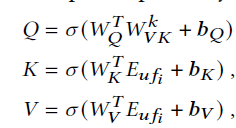
𝐸𝑡𝑗 = 𝑔𝑡(𝑡𝑗;Θ𝑡). (3)

上述解决方案被广泛使用，但无法有效地生成多目标标签。它需要构建两个独立的模型分别挖掘兴趣和意图标签，而这种传统的双塔模型在效率上牺牲了一定的准确性。在接下来的部分中，我们提出了一个名为MVKE的统一模型，以更好地实现目标，统一地挖掘多样化且准确的标签，节省计算资源并提高效率。

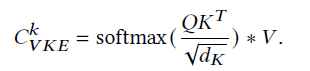
**单任务**

在单任务目标上介绍MVKE的结构清晰。MVKE的架构如图4所示，仍然类似于双塔模型，但两个塔之间有一个"桥梁"。可以发现标签塔的内部结构几乎与基本解决方案中的一致，但用户塔则有很大的区别。两个塔之间的桥梁包括两个组件：Virtual-Kernel Experts（VKE）和Virtual-Kernel Gate（VKG）。每个VKE只关注用户偏好的一方面，并通过相应的虚拟内核提供帮助。VKG是基于注意力的加权门，用于有选择地将VKE的输出组合到最终的用户表示中。换句话说，VKE的建模是不同和多样的，VKG根据不同标签对虚拟内核的注意力将VKE的输出进行组合。

用户塔中有多个VKE，但所有的VKE共享相同的特征输入层。它们的数量是一个可以灵活设置的超参数。以第k个VKE为例，其中包含一个虚拟内核，表示为𝑊𝑘𝑉𝐾。虚拟内核是一个可学习的变量，并作为VKE中注意力机制中的"Query"输入。这里的"Key"和"Value"都是用户特征嵌入𝐸𝑢𝑓𝑖。具体而言，VKE中的第一步是对每个输入进行非线性变换，定义为：

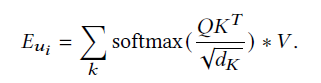


其中𝜎(·)表示应用激活函数，𝑊和𝑏表示线性层中的权重和偏置。然后，它计算注意力权重以引导输入特征的组合：



在顶层，一种简单的聚合方法是对每个特征字段嵌入进行加权求和。由于VKE的高度灵活性，我们还可以选择在加权用户字段嵌入上叠加一些复杂的交互结构，例如DeepFM、xDeepFM，表示为. 每个VKE在虚拟内核的引导下输出一个紧凑的嵌入𝐸𝑘𝑢𝑖，但它是隐式的并且没有实际意义。

为了生成特定于标签的用户表示，设计了一种基于注意力机制的软门控网络，称为VKG。VKG还是基于注意力机制的，输入包括所有虚拟内核𝑊𝑉𝐾（"Key"），标签嵌入𝐸𝑇𝑖（"Query"）和所有VKE的输出𝐸𝑢𝑖（"Value"）。形式上，通过对非线性变换的𝑄(𝐸𝑇𝑖)和𝐾(𝑊𝑘𝑉𝐾)进行计算得到注意力权重，然后输出加权求和的𝑉(𝐸𝑘𝑢𝑖)，作为最终的用户嵌入𝐸𝑢𝑖，定义如下：

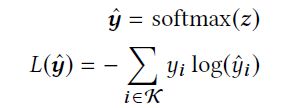


损失函数与基本解决方案中定义的类似，通过用户嵌入𝐸𝑢𝑖和标签嵌入𝐸𝑇进行计算。回顾整体结构，可以很容易地发现虚拟内核在MVKE中起着重要的作用。它既在VKE中使用，又在VKG中使用，因此可以看作是连接用户和标签的桥梁。VKE专注于用户的隐式偏好，这些偏好在VKG中聚合成与标签对应的具有实际含义的用户表示（例如之前提到的"Sports"）。在某种意义上，虚拟内核确定了用户偏好的具体学习空间。此外，所有的虚拟内核可以结合并映射到不同的真实标签空间。在虚拟内核作为桥梁的帮助下，用户和标签特征可以更好地相互交互。

# SDM

Training and Online Serving

在训练过程中，时间t上的正标签是下一个互动的物品iu t+1。负标签是从I中（排除iu t+1）使用对数均匀采样器进行采样的，考虑到实际应用中物品的大量性质。然后，通过softmax层生成预测类别概率。这称为采样softmax[10]，我们将交叉熵应用为损失函数：



其中，K是包括正负标签在内的采样子集，z = [z1, ..., z|K|]是输出和每个vi（i∈K）之间的内积得分，ŷ = [ŷ1, ..., ŷ|K|]是对每个采样物品的预测概率分布，yi是物品i的真实概率分布。

我们将模型部署在我们的在线推荐系统上。物品嵌入向量V被导入到高效的K最近邻（KNN）相似度搜索系统[11]中。同时，用户预测网络部署在高性能的实时机器学习推理系统上。这种架构类似于YouTube的视频推荐[4]。当用户使用我们的在线服务时，他们会与许多物品进行交互，并且他们对物品的反馈将被后端系统记录下来。这些信息将被处理，然后存储在数据库中作为用户行为日志。从海量日志中提取有用的信息，构建成我们模型所需的结构化数据。在时间t，用户的历史行为（Su和Lu）被输入到推理系统中。然后预测用户行为向量out。KNN搜索系统根据它们的内积检索与out最相似的物品。然后推荐出前N个物品。现在我们详细说明Su和Lu如何在网络中编码，并如图2所示融合这两种表示。

Input Embedding with Side Information

在淘宝的推荐场景中，顾客不仅关注特定物品本身，还关注品牌、店铺、价格等方面。例如，有些人倾向于购买特定品牌的物品，而其他人则喜欢从值得信赖的店铺购买物品。此外，由于行业中大规模在线物品引起的稀疏性，仅通过物品ID特征级别对物品进行编码远远不够。因此，我们从不同的特征尺度描述物品，例如物品ID、叶子类别、一级类别、品牌和店铺，它们被表示为附加信息集合F。每个输入物品iu t ∈ Su都被表示为通过嵌入层转换得到的稠密向量eiu t ∈ Rd×1，以便可以直接输入深度神经网络中。



其中，ef i = Wf xf i ∈ Rdf ×1是特征f的嵌入大小为df 的物品i的输入嵌入，Wf 是特征f的转换矩阵，而xf i是一个独热向量。

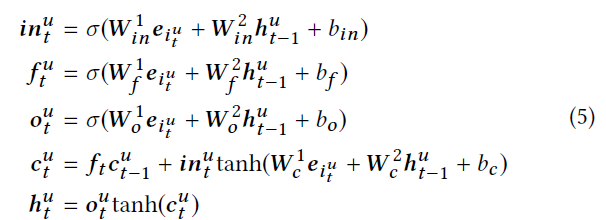
类似地，用户配置文件可以从不同的特征尺度描述用户u，例如年龄、性别和生命周期阶段。用户u的配置文件信息输入表示为一个稠密向量eu ∈ Rd×1：



其中，P是配置文件特征集合，ep u是特征p的嵌入。

**Recurrent Layer**

为了捕捉和表征短期序列数据中的全局时间依赖关系，给定用户u的嵌入的短期序列[eiu1, ..., eiut]，我们采用长短期记忆（LSTM）网络作为递归单元，这是基于会话推荐[7, 14, 15]的方法。LSTM可以描述为：



其中，inut、fut和out分别表示输入门、遗忘门和输出门。LSTM将用户u的短期交互序列编码为时间t处的隐藏输出向量hut，我们称之为序列偏好表示。cut是细胞状态向量，携带着hut-1的信息并在细胞之间流动。我们将hut传递给注意力网络，以获得更高阶的表示。

**注意力机制**

在在线购物场景下，客户通常会交替浏览一些不相关的商品，这些被称为因果点击。不相关的行为会在序列中影响hut的表示。我们使用自注意力网络来减少这些不相关行为的影响。注意力网络[1, 18, 28]可以通过为每个组成部分分配不同的权重来将各种向量聚合成一个整体表示。

**多头自注意力。**自注意力是注意力机制的一种特殊情况，它将序列本身作为d维的查询、键和值向量。经过自注意力之后，输出向量ˆhut可以从LSTM的先前隐藏输出Xu = [hu1, ..., hut]中聚合得到。

用户可能有多个兴趣点。例如，当用户正在浏览一条裙子时，颜色和新颖的款式都是做出决策的关键因素。单个注意力网络很难捕捉到多个方面的表示。多头注意力允许模型同时关注不同表示子空间和不同位置上的信息[25]，可以从多个兴趣视角对用户偏好ˆhut进行建模。因此，我们在注意力机制中引入了多头注意力。输出矩阵ˆXu = [ˆhu1, ..., ˆhut]的计算如下：

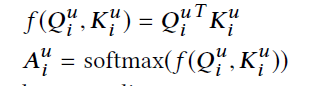


其中，WO ∈ Rd×hdk表示输出线性变换的权重矩阵，h表示头的数量，dk = 1/(h\*d)。

具体而言，每个头headui ∈ Rdk×t表示一个单独的潜在兴趣头：



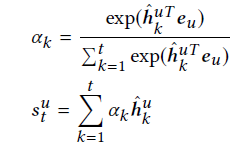
其中，WQi、WKi、WVi ∈ Rdk×d分别表示查询、键和值的线性变换权重矩阵。令Qui = WQi Xu，Kui = WKi Xu，Vui = WVi Xu。注意力得分矩阵定义如下：



最后，通过加权求和池化，我们得到：



**用户注意力。**对于不同的用户，即使是相似的商品集合，他们通常也会表现出不同的偏好。因此，在自注意力网络之上，我们添加了一个用户注意力模块，用于挖掘更细粒度的个性化信息，其中eu被用作查询向量，注意到ˆXu = [ˆhu1, ..., ˆhut]。时间t处的短期行为表示sut ∈ Rd×1计算如下：



**长期行为融合**

从长期来看，用户通常在各个维度上积累了不同层次的兴趣。一个用户可能经常访问一组相似的店铺，并重复购买属于同一类别的商品。因此，我们还需要对来自不同特征尺度的长期行为Lu进行编码。Lu = {Luf | f ∈ F}由多个子集组成：Lu

id（商品ID），Lu

leaf（叶子类别），Lu

cate（一级类别），Lu

shop（店铺）和Lu

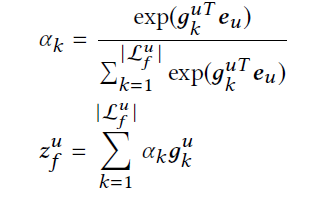
brand（品牌），如图2所示。例如，Lu

shop包含了用户在过去一周内互动过的店铺。每个子集中的条目被嵌入并通过基于注意力的池化操作聚合成一个整体向量，考虑到在线环境下的快速响应。

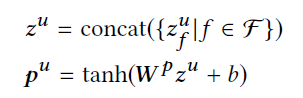
每个f u

k ∈ Luf通过Wf转换为一个稠密向量дu

k ∈ Rd×1，如第3.3节所述。然后，我们使用用户属性嵌入eu作为查询向量来计算注意力分数，并获得Luf的表示：



其中{zuf | f ∈ F}被连接起来并输入到一个全连接神经网络中：



其中pu ∈ Rd×1是长期行为的表示。

为了与短期行为相结合，我们精心设计了一个门控神经网络，它以eu、su

t和pu作为输入，如图2所示。门向量Gut ∈ Rd×1用于决定时间t时短期和长期行为的贡献比例：



最终的输出，即用户行为向量out ∈ Rd×1，由以下计算得出：



其中⊙表示逐元素乘法。

# SINE

由于现实世界的推荐系统中商品池通常由数百万甚至数十亿个商品组成，匹配阶段在现代推荐系统中至关重要。具体而言，匹配阶段的深度序列模型通常包括一个序列编码器 𝜙𝜃(·) 和一个项目嵌入表 H ∈ R𝑀×𝐷，其中 𝜃 是包含所有可训练参数（包括 H）的集合。编码器将用户的历史行为序列 x(𝑢) 作为输入，并输出序列 𝜙𝜃(x(𝑢)) 的表示，可以视为用户意图的表示。用户的意图嵌入然后被用作查询，通过快速的 K 最近邻算法（如 faiss [20]）从项目池中生成候选项目。文献中大多数编码器 𝜙𝜃(·) 输出一个单一的 𝐷 维嵌入向量，但也有输出 𝐾 个 𝐷 维嵌入向量以在 𝐾 个潜在类别下保留用户的意图的模型。我们主要关注后一种方法，旨在准确捕捉用户的多样意图。

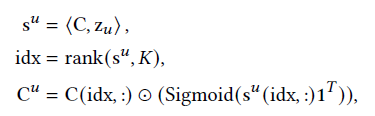
用于捕捉用户多个意图的最先进序列编码器可以总结为两类。第一类方法借助强大的序列编码器隐式提取用户的多个意图，例如基于多头自注意力的模型（也称为Transformer [43]）。另一类方法依赖于潜在原型显式捕捉用户的多个意图。一般来说，前一种方法可能由于实践中意图检测和嵌入的混合性质而限制其捕捉多个意图的能力。例如，实证结果表明，Transformer学习到的多个向量表示似乎在推荐中与单头实现[21]相比没有明显优势。相反，后一种方法可以通过聚类识别的概念的帮助有效地提取用户的多样兴趣，这在[27, 29]中经验证明。然而，这些方法的可扩展性较差，因为它们要求每个用户在每个概念下都有一个意图嵌入，而在工业应用中很容易扩展到数千个。例如，在中国天猫电商平台中，数百万甚至数十亿个商品属于超过10,000个专家标记的叶子类别[24]。在实际系统中有大量的兴趣概念，需要一个可扩展的多兴趣提取模块。

因此，我们在这里提出了一种稀疏兴趣网络，它能够自适应地激活来自大量概念池的一部分概念，以适应用户。我们模型的输入是用户的行为序列 x(𝑢)，然后将其输入到一个嵌入层，并转换为项目嵌入矩阵 X𝑢 ∈ R𝑛×𝐷。设 C ∈ R𝐿×𝐷 为总体概念原型矩阵，C𝑢 ∈ R𝐾×𝐷 表示用户 𝑢 的激活概念原型嵌入矩阵在 𝐾 个潜在概念上。其中 𝐿 是概念的总数。

**概念激活**。我们的稀疏兴趣层首先推断出每个用户𝑢的激活概念原型C𝑢。给定X𝑢 ∈ R𝑛×𝐷，首先应用自注意力方法[26]有选择地聚合输入序列。

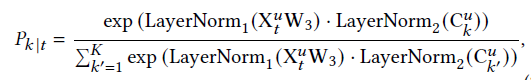


其中 W1 ∈ R𝐷×𝐷 和 W2 ∈ R𝐷 是可训练参数。向量 a ∈ R𝑛 是用户行为的注意权重向量。当根据注意权重对输入序列的嵌入进行求和时，我们可以获得用户的虚拟概念向量 z𝑢 = (a⊤X𝑢)⊤。z𝑢 ∈ R𝐷 反映了用户的一般意图，并可用于激活感兴趣的概念原型，如下所示：



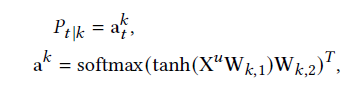
其中 rank(s𝑢,𝐾) 是前 K 个排名运算符，返回 s𝑢 中𝐾个最大值的索引。rank(s𝑢,𝐾) 返回的索引包含为用户𝑢选择的原型的索引。C(idx, :) 执行行提取以形成子原型矩阵，而 s(idx, :) 提取 s𝑢 中索引为 idx 的值。1 ∈ R𝐾 是所有元素都为 1 的向量。 ⊙ 表示哈达玛积，⟨·, ·⟩ 是内积。C𝑢 ∈ R𝐾×𝐷 是用户𝑢的最终激活的 𝐾 个潜在概念嵌入矩阵。方程2是一种𝐾个最高选择的技巧，使得离散选择操作可微分。先前的工作[8]发现它在逼近最高𝐾选择问题时非常有效。

**意图分配**。在推断出当前的概念原型C𝑢之后，我们可以根据它们与行为序列中每个项目之间的距离来估计用户与每个项目相关的意图。



其中 𝑃𝑘 |𝑡 衡量在位置 𝑡 处的主要意图与第 𝑘 个潜在概念相关的可能性。C𝑢𝑘 ∈ R𝐷 是用户 𝑢 的第 𝑘 个激活概念原型的嵌入。W3 ∈ R𝐷×𝐷 是可训练的权重矩阵。LayerNorm𝑙 (·) 表示层归一化层。请注意，由于归一化，我们在这里使用余弦相似度而不是内积。这个选择是基于余弦相似度在模型坍塌[29]（例如，模型忽略大多数原型的退化情况）方面比点积更加稳健的事实。

**注意权重**。除了从概念角度计算出的注意权重 𝑃𝑘 |𝑡，我们还考虑另一个注意权重 𝑃𝑡 |𝑘，用于估计位置 𝑡 处的项目对预测用户下一个意图的重要性。

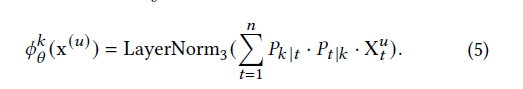


其中 a𝑘 ∈ R𝑛 是所有位置的注意向量。上标 𝑘 表示它是第 𝑘 个激活意图的注意层。与方程 1 类似，上述方程是另一个自注意层。主要的区别在于我们在这里尝试利用用户序列的顺序，并为输入嵌入添加了额外的可训练位置嵌入 [43]。位置嵌入的维度与项目嵌入相同，因此它们可以直接相加。

**兴趣嵌入生成**。现在，我们可以根据 𝑃𝑘 |𝑡 和 𝑃𝑡 |𝑘 从用户的行为序列 X𝑢 生成多个兴趣嵌入向量。具体来说，我们稀疏兴趣编码器的第 𝑘 个输出 𝜙𝑘

𝜃

(x(𝑢) ) ∈ R𝐷 的计算如下：



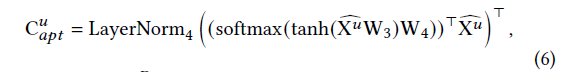
到目前为止，我们已经介绍了稀疏兴趣网络的整个过程。给定一个用户的行为序列，我们首先从概念池中激活他/她首选的概念原型。然后进行意图分配，估计输入序列中每个项目与用户意图的相关性。然后，应用自注意层计算所有项目的注意权重，用于下一个项目的预测。最后，根据公式 5，通过加权求和生成用户的多个兴趣嵌入向量。

**Interest Aggregation Module**

在稀疏兴趣提取模块之后，我们为每个用户获得了多个兴趣嵌入。一个自然的问题是如何利用各种兴趣进行实际推断。一种直观的解决方案是将下一个预测的项目作为目标标签，在训练过程中选择不同的兴趣嵌入，就像 MIND [24] 中所做的那样。尽管这种方法简单，但主要的缺点是在推断过程中没有目标标签，导致训练和测试之间存在差距，并可能导致性能下降。

为了解决这个问题，我们提出了一个基于主动预测的自适应兴趣聚合模块。这里的动机是相比于找到理想标签，更容易预测用户的基于时间的下一个意图。具体而言，基于方程式 3 中计算的意图分配分数 𝑃𝑘 |𝑡，我们可以得到一个意图分布矩阵 P𝑢 ∈ R𝑛×𝐾，用于描述行为序列中所有项目的意图。然后，输入行为序列 x𝑢 可以从意图的角度进行重新定义，记为 cX𝑢 = P𝑢C𝑢，其中 cX𝑢 ∈ R𝑛×𝐷 被视为用户 𝑢 的意图序列。通过 cX𝑢，用户的下一个意图 C𝑢𝑎

𝑝𝑡 被自适应地计算为



其中 C𝑢𝑎

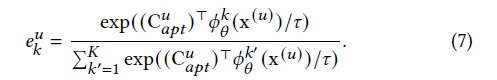
𝑝𝑡 ∈ R𝐷 是用户 𝑢 对下一个项目的预测意图。W3 ∈ R𝐷×𝐷 和 W4 ∈ R𝐷 是可训练参数。给定 C𝑢𝑎

𝑝𝑡 和多个兴趣嵌入 {𝜙𝑘

𝜃

(x(𝑢) )}𝐾

𝑘=1，不同兴趣的聚合权重计算如下：

 (x(𝑢) )/𝜏)

其中 𝑒𝑢 = [𝑒𝑢 1 , 𝑒𝑢 2 , · · · , 𝑒𝑢𝐾]𝑇 ∈ R𝐾 是用于不同兴趣的注意向量。𝜏 是一个用于调节的温度参数。当 𝜏 较大 (𝜏 → ∞) 时，𝑒𝑢 近似为均匀分布的向量。当 𝜏 较小 (𝜏 → 0+) 时，𝑒𝑢 近似为一个独热向量。在实验中，我们使用 𝜏 = 0.1 来确保聚合器选择最受用户喜爱的意图进行推断。最终用户表示 v𝑢 ∈ R𝐷 的计算方式如下：

