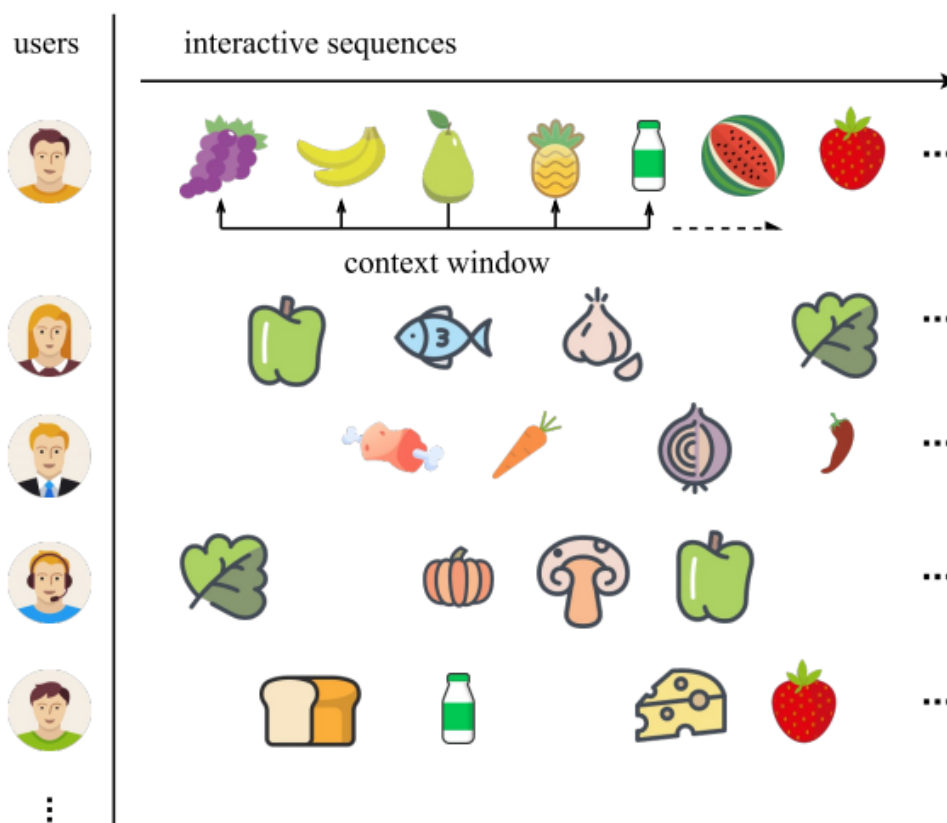


提出一个基于嵌入的框架来学习和转移IDs的表示，所有类型的ID可以嵌入到一个低维的语义空间。

### 3 学习IDS的表示

#### 3.1 在用户交互序列上的Skip-gram



**Figure 3: The skip-gram model with context window of length  $C = 2$ .**

用target item预测context item

$$\mathcal{J} = \frac{1}{N} \sum_{n=1}^N \sum_{\substack{1 \leq n+j \leq N, j \neq 0 \\ -C \leq j \leq C}} \log p(\text{item}_{n+j} | \text{item}_n), \quad (1)$$

$$p(\text{item}_j | \text{item}_i) = \frac{\exp(\mathbf{e}'_j \mathbf{T} \mathbf{e}_i)}{\sum_{d=1}^D \exp(\mathbf{e}'_d \mathbf{T} \mathbf{e}_i)}, \quad (2)$$

## 3.2 负采样 Log-uniform

由于softmax分母的计算成本是很高的，采用NCE ( Noise Contrastive Estimation )，进行负采样：

$$p(\text{item}_j|\text{item}_i) = \sigma(\mathbf{e}_j'^T \mathbf{e}_i) \prod_{s=1}^S \sigma(-\mathbf{e}_s'^T \mathbf{e}_i), \quad (3)$$

没有使用均匀分布负采样，采用 log-uniform负采样，因为越流行的ID提供更少的信息。为了加速负采样的过程，采用近似Zipfian分布（这还没看懂）

## 3.3 IDs及他们的结构连接

两类ID

- Item ID和其属性ID，本文中 item ID, product ID, store ID, brand ID, cate-level1 ID, cate-level2 ID and cate-level3 ID. (不同store中的相同Item，共享product ID,但是item ID不同)
- User ID,比如cookie，设备IMEI，登录用户名等

## 3.4 联合嵌入属性IDs

(1) item ID的共现也涉及到属性ID的共现

假设K种ID，将公式3由公式7代替：

$$\begin{aligned} & p(\text{IDs}(\text{item}_j)|\text{IDs}(\text{item}_i)) \\ &= \sigma \left( \sum_{k=1}^K (w_{jk} \mathbf{e}_{jk}')^T (w_{ik} \mathbf{e}_{ik}) \right) \\ & \prod_{s=1}^S \sigma \left( - \sum_{k=1}^K (w_{sk} \mathbf{e}_{sk}')^T (w_{ik} \mathbf{e}_{ik}) \right), \end{aligned} \quad (7)$$

- 相当于K个ID，对应的ID嵌入做内积，但是item i 和item j每个ID内积时，都有一个权重  $w_{ik}$   $w_{jk}$
- 不同的ID嵌入维度可以不同，因为只有对应类型的ID做内积
- $w_{ik}$  是对于item i来说第k种ID的权重， $\text{id}_k(\text{item}_i)$  包含  $V_{ik}$  个item，设想  $\text{id}_k(\text{item}_i)$  中的每一项对于其贡献是相同的。

$$\mathbf{I}(x) = \begin{cases} 0 & , x \text{ is False} \\ 1 & , x \text{ is True} \end{cases}, \quad (8)$$

$$V_{ik} = \sum_{j=1}^D \mathbf{I}(\text{id}_k(\text{item}_i) = \text{id}_k(\text{item}_j)), \quad (9)$$

$$w_{ik} = \frac{1}{V_{ik}} (k = 1, \dots, K). \quad (10)$$

- 有点绕。。就是item i的第K种ID嵌入的权重相当于：①先计算item i的第k种ID取值，在所有样本中出现的次数count ②然后权重= 1/count 。比如男生出现了40次，女生出现了60次，那么男生对于某item的权重是1/40，乘上男生的嵌入。

## (2) item ID和属性ID之间的连接意味着约束


- 两个item ID共现，不仅意味着两个item ID嵌入要接近，意味着两个item相同属性的ID嵌入也要接近。
- 以store ID为例，应该是此商店内所有item ID的合理summary（到底是这个例子这种意思，还是说，item ID应该是所有属性id的summary呢，看损失函数应该是后者）

$$p(\text{item}_i | \text{IDs}(\text{item}_i)) = \sigma \left( \sum_{k=2}^K w_{ik} \underbrace{\mathbf{e}_{i1}^T \mathbf{M}_k \mathbf{e}_{ik}}_{\text{双线性插值}} \right), \quad (11)$$

- $M_k$  的作用是将  $e_1$  嵌入转化(transform) 到  $e_k$  的维度
- 最大化以下公式，代替公式1：

$$\mathcal{J} = \frac{1}{N} \sum_{n=1}^N \left( \sum_{\substack{1 \leq n+j \leq N, j \neq 0 \\ -C \leq j \leq C}} \log p(\text{IDs}(\text{item}_{n+j}) | \text{IDs}(\text{item}_n)) \right) \quad (12)$$

- 这是关于item n 的集合，是什么集合呢  

$$+ \alpha \log p(\text{item}_n | \text{IDs}(\text{item}_n)) - \beta \sum_{k=1}^K \|\mathbf{M}_k\|_2 \Bigg),$$


- $\alpha$  表示了约束的强度， $\beta$  表示了变换矩阵的L2正则化

## 3.5 用户ID的嵌入

通过聚合用户的交互序列item id集合来表示用户，比如平均、RNN等

本文采用平均

$$\text{Embedding}(u) = \frac{1}{T} \sum_{t=1}^T e_t, \quad (13)$$

## 3.6 模型学习

- Xavier初始化所有可训练的参数
- SGD，Adam优化器，shuffled mini-batches
- 超参数：
  - windows size C=4，左4个，右4个
  - 负样本个数S=2
  - batch size 128，5个epoch

## 4 利用 IDS 表示

## 4.1 衡量物品相似度

- item相似物品推荐
- 召回，每个用户最近的物品作为seed集，每个物品的top N相似物品作为候选集：

the **candidate set** for user  $u$  is:

$$\text{candidate}(u) = \bigcup_{\text{seed}_i \in \text{SEED}(u)} (\text{top-N similar items of seed}_i). \quad (14)$$

item-item相似度也可以用到item based CF中

## 4.2 从已知物品转化到未知物品

- 物品冷启动问题，由于新的item id没有交互历史，而一些基于内容的方法解决了冷启动问题。
- 本文解决冷启动的方法是：为新Item构造近似的嵌入向量。
  - 基本思想是，新物品连接的IDs（不是item ID）有历史纪录
  - 根据新物品连接的IDs为新物品构造近似嵌入

公式11，推导出公式15：

$$\begin{aligned} & p(\text{item}_i | \text{IDs}(\text{item}_i)) \\ &= \sigma \left( \sum_{k=2}^K w_{ik} \mathbf{e}_{i1}^T \mathbf{M}_k \mathbf{e}_{ik} \right) \\ &\propto \sum_{k=2}^K w_{ik} \mathbf{e}_{i1}^T \mathbf{M}_k \mathbf{e}_{ik} \\ &= \mathbf{e}_{i1}^T \left( \sum_{k=2}^K w_{ik} \mathbf{M}_k \mathbf{e}_{ik} \right). \end{aligned} \quad (15)$$

最大化公式12，导致  $p(\text{item}_i | \text{IDs}(\text{item}_i)) \rightarrow 1$ ，因此近似为：

$$\begin{aligned}
& p(\text{item}_i | \text{IDs}(\text{item}_i)) \rightarrow 1 \\
& \Rightarrow \mathbf{e}_{i1}^T \left( \sum_{k=2}^K w_{ik} \mathbf{M}_k \mathbf{e}_{ik} \right) \text{ is relatively large} \\
& \Rightarrow \mathbf{e}_{i1} \approx \sum_{k=2}^K w_{ik} \mathbf{e}_{ik}^T \mathbf{M}_k^T.
\end{aligned} \tag{16}$$

(个人认为, 和content-based类似, 就是用属性id来重构新item的嵌入, 暂时忽略了item id)

### 4.3 在不同领域之间转化

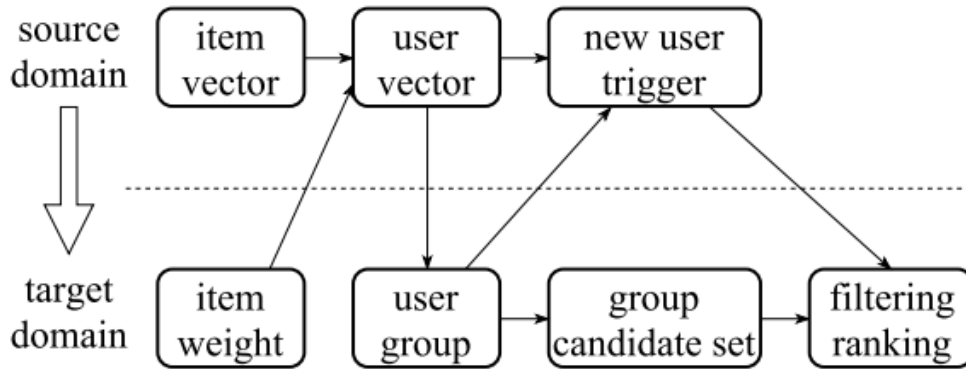
(产品矩阵, 比如河马, 新的APP用户都是新用户, 可以将这些用户在其他APP上累积的数据学到的偏好迁移过来)

将用户在source domain的偏好迁移到target domain

源领域用户集 $U^s$ , 目标领域用户集 $U^t$ , 共同用户集 $U^i$

- $U^s$  中的所有用户通过(平均)聚合交互item嵌入, 获取用户嵌入
- $U^i$  用户k-means聚类成1000组
- 对于每个聚类组, 最流行的N个河马item作为候选集
- $U^s$ 、 $U^i$  中的新用户根据和聚类中心的嵌入相似度分配给某个聚类组
- 将对应的候选集进行过滤和排名

(没太看懂??)



**Figure 5: The process of transferring user vectors across different domains.**

每种隐行为可以赋予权重, 公式13可以转化成:

$$\text{Embedding}(u) = \frac{\sum_{t=1}^T w_t e_t}{\sum_{t=1}^T w_t}. \tag{18}$$

## 4.4 在不同任务之间转化

将store ID 的嵌入和历史销量数据作为模型输入，预测下一阶段的销量。