知乎 写文章

【推荐系统经典论文(九)】谷歌双塔模型

努力搬砖...

46 人赞同了该文章

Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations

背景介绍

• 文章核心思想?

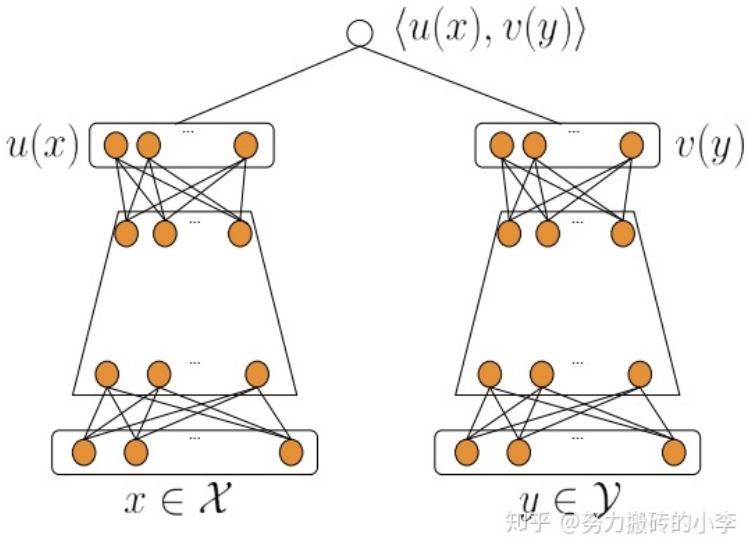
在大规模的推荐系统中，利用双塔模型对user-item对的交互关系进行建模，学习【用户，上下文】向量与【item】向量，针对大规模流数据，提出**in-batch softmax损失函数**与**流数据频率估计方法(Streaming Frequency Estimation)**，可以更好的适应item的多种数据分布。

• 文章贡献

1. 提出流数据频率估计方法：针对流数据来估计item出现的频率，并利用实验分析估计结果的偏差与方差，模拟实验证明该方法在数据动态变化时的功效
2. 提出模型架构：提供了一个针对大规模的检索推荐系统，包括了in-batch softmax损失函数与流数据频率估计方法，减少了负采样在每个batch中可能会出现的采样偏差问题
3. YouTube推荐：将大规模的检索推荐系统用户YouTube，端对端实现推荐
4. 线下和真实实验：利用两个数据集实验，检验模型效果

算法原理

利用双塔模型构架推荐系统，Queries特征向量 $\{x_i\}_{i=1}^N$，item*特征向量* $\{y_j\}_{j=1}^M$ ，目标是给定一个query，检索到一系列item子集用于后续排序推荐任务。模型结构如图所示：



首先建立两个参数embedding函数， $u : X \times R^d \rightarrow R^k, v : Y \times R^d \rightarrow R^k$ ，把query和候选item映射到k维向量空间，模型的输出为二者的embedding内积，即：$s(x, y) =< u(x, \theta), v(y, \theta) >$ ，我们的目标是根据训练集 $T := \{(x_i, y_i, r_i)\}_{i=1}^T$ 来学习参数 θ (其中 r_i 为用户反馈，比如说用户花在一个视频上的时间等)

In-batch loss function

▲ 赞同 46 ▼ 4 条评论 分享 喜欢 收藏 申请转载 ···

直觉上，检索问题可以看作是一个多分类问题，给定query X，从M个item中得到y的概率可以利用softmax函数计算：

$$P(y|x,\theta)=\frac{e^{s(x,y)}}{\sum_{j\in[M]}e^{s(x,y_j)}}$$ ，

考虑反馈 r_i ，加权对数似然损失函数为：

$$L_T(\theta):=-\frac{1}{T}\sum_{i\in|T|}r_i log(P(y_i|x_i,\theta))$$

当M非常大时，我们通常可以利用负采样算法进行计算。然而对于流数据，我们考虑在同一个batch中采样负样本，batch-softmax函数为：

$$P_B(y_i|x_i,\theta)=\frac{e^{s(x_i,y_i)}}{\sum_{j\in[B]}e^{s(x_i,y_j)}}$$

在每个batch中，由于存在幂律分布现象，即如果在每个batch中随机采样负样本，会使热门商品更容易被采样到，在损失函数中就“过度”惩罚了这些热门商品，因此考虑用频率对采样进行修正，即：

$$s^c(x_i,y_j)=s(x_i,y_j)-log(p_j)$$

其中 p_j 是在每个batch中随机采样到item j的概率(将在下一节中介绍)，因此修正后的条件概率函数为：

$$P_B^c(y_i|x_i,\theta)=\frac{e^{s^c(x_i,y_i)}}{e^{s^c(x_i,y_i)}+\sum_{j\in[B],j\neq i}e^{s^c(x_i,y_j)}}$$

则损失函数为：

$$L_B(\theta):=-\frac{1}{B}\sum_{i\in[B]}r_i log(P_B^c(y_i|x_i;\theta))$$

即为batch loss function，然后可以利用SGD来更新参数 θ

一些tricks

- 最近邻搜索：当embedding映射函数u和v学习好后，预测包含两步：1)计算query的向量 $u(x,\theta)$ 2)从事先训练好的函数v中找到最邻近的item。考虑到耗时问题，此处利用hash技术采用近邻搜索等方法进行处理
- 归一化：经验表明，对函数归一化效果更好，即 $u(x,\theta)=u(x,\theta)/||u(x,\theta)||_2, v(x,\theta)=v(x,\theta)/||v(x,\theta)||_2$,对每个logit函数，利用超参数 τ 进行处理： $s(x,y)=s(x,y)/\tau$

Streaming Frequency Estimation

此方法用于估计在流数据中，每个batch下item出现的概率。

如果一个item每50步出现一次，那么该item出现的概率p=1/50=0.02。按照这样的想法，针对流数据，利用哈希序列来记录采样id(暂时不考虑hash collision的问题)。

定义两个大小为H的数组A，B，哈希函数h可以把每个item映射为[H]内的整数。

- A[h(y)]表示item y上次被采样到的时刻

▲ 赞同 46 ▼ 4 条评论 分享 喜欢 收藏 申请转载 ···

先说结论，当第t步y被采样到时，利用迭代可更新A，B：

$$B[h(y)] = (1-\alpha)B[h(y)] + \alpha(t - A[h(y)])$$
$$A[h(y)] = t$$

α 可看作学习率。通过上式更新后，则在每个batch中item y出现的概率为 $1/B[h(y)]$ 。

直观上，上式可以看作利用SGD算法和固定的学习率 α 来学习“可以多久被采样到一次”这个随机变量的均值。

下面，可以从**数学理论上**证明这种迭代更新的有效性：

假设item y被采样到的时间间隔序列为 $\Delta = \{\Delta_1, \ldots, \Delta_t\}$,满足独立同分布，这个随机变量的均值为 $\delta = E[\Delta]$,对于迭代： $\delta_i = (1-\alpha)\delta_{i-1} + \alpha\Delta_i$,可以证明对于这个序列(可以看作上文提到的数组B)均值和方差：

$$E(\delta_t) - \delta = (1-\alpha)^t\delta_0 - (1-\alpha)^{t-1}\delta$$
$$E[(\delta_t - E[\delta_t])^2] \le (1-\alpha)^{2t}(\delta_0 - \delta)^2 + \alpha E[(\Delta_1 - \alpha)^2]$$

证明：**对于均值**：

$$\begin{aligned}
E[\delta_i] &= (1-\alpha)E[\delta_{i-1}] + \alpha\delta \\
&= (1-\alpha)[(1-\alpha)E[\delta_{i-2}] + \alpha\delta] + \alpha\delta \\
&= (1-\alpha)^2E[\delta_{i-2}] + [(1-\alpha)^1 + (1-\alpha)^0]\alpha\delta \\
&= (1-\alpha)^3E[\delta_{i-3}] + [(1-\alpha)^2 + (1-\alpha)^1 + (1-\alpha)^0]\alpha\delta \quad \text{即} \\
&= \ldots\ldots \\
&= (1-\alpha)^t\delta_0 + [(1-\alpha)^{t-1} + \ldots + (1-\alpha)^1 + (1-\alpha)^0]\alpha\delta \\
&= (1-\alpha)^t\delta_0 + [1 - (1-\alpha)^{t-1}]\delta
\end{aligned}$$

则 $E(\delta_t) - \delta = (1-\alpha)^t\delta_0 - (1-\alpha)^{t-1}\delta$

对于方差：

$$\begin{aligned}
E[(\delta_t - E[\delta_t])^2] &= E[(\delta_t - \delta + \delta - E[\delta_t])^2] \\
&= E[(\delta_t - \delta)^2] + 2E[(\delta_t - \delta)(\delta - E[\delta_t])] + (\delta - E[\delta_t])^2 \\
&= E[(\delta_t - \delta)^2] - (E[\delta_t] - \delta)^2 \\
&\le E[(\delta_t - \delta)^2]
\end{aligned}$$

对于最后一项，

$$\begin{aligned}
E[(\delta_i - \delta)^2] &= E[((1-\alpha)\delta_{i-1} + \alpha\Delta_i - \delta)^2] \\
&= E[((1-\alpha)\delta_{i-1} + \alpha\Delta_i - (1-\alpha+\alpha)\delta)^2] \\
&= E[((1-\alpha)(\delta_{i-1} - \delta) + \alpha(\Delta_i - \delta))^2] \\
&= (1-\alpha)^2E[(\delta_{i-1} - \delta)^2] + \alpha^2E[\Delta_i - \delta]^2 + 2\alpha(1-\alpha)E[(\delta_{i-1} - \delta)(\Delta_i - \delta)]
\end{aligned}$$

由于 δ_{i-1} 和 Δ_i 独立，所以上式最后一项为0，则

$E[(\delta_i - \delta)^2] = (1-\alpha)^2E[(\delta_{i-1} - \delta)^2] + \alpha^2E[\Delta_i - \delta]^2$ ；

与均值的推导类似，可得：

$$\begin{aligned}
E[(\delta_t - \delta)^2] &= (1-\alpha)^{2t}(\delta_0 - \delta)^2 + \alpha^2\frac{1 - (1-\alpha)^{2t-2}}{1 - (1-\alpha)^2}E[(\Delta_1 - \delta)^2] \\
&\le (1-\alpha)^{2t}(\delta_0 - \delta)^2 + \alpha E[(\Delta_1 - \delta)^2]
\end{aligned}$$

即 $E[(\delta_t - E[\delta_t])^2] \le (1-\alpha)^{2t}(\delta_0 - \delta)^2 + \alpha E[(\Delta_1 - \alpha)^2]$

证毕。

对于上述均值，当 $t \to \infty$ 时， $|E[\delta_t] - \delta| \to 0$ ，即当采样数据足够多的时候，数组B(每多少步采样一次)趋于真实采样频率。递推式合理。且当初始值 $\delta_0 = \delta/(1-\alpha)$ ，递推式为无偏

▲ 赞同 46 ▼ 4 条评论 分享 喜欢 收藏 申请转载 ···

对于方差，上式给了一个估计方差的上界。

利用上述的**In-batch loss function**与**Streaming Frequency Estimation**可建立双塔模型：

算法一：训练算法

输入： 参数embedding函数$u(\cdot,\theta)$和$v(\cdot,\theta)$，学习率γ

1. 迭代：
2. 从数据流中采样数据表示$\{(x_i,y_i,r_i)\}_{i=1}^{B}$
3. 利用频率估计算法计算每个item y_i的概率p_i
4. 计算损失函数$L_B(\theta)=-\frac{1}{B}\sum_{i\epsilon[B]} r_i\log(P_B^c(y_i|x_i,\theta))$
5. 更新参数$\theta=\theta-\gamma\nabla L_B(\theta)$

知乎 @努力搬砖的小李

算法二：频率估计算法

输入： 学习率α，大小为H的数组A, B, 哈希函数h

1. 对于每一步$t=1,2,\ldots$:
2. 对于每个item y:
3. $B[h(y)]=(1-\alpha)B[h(y)]+\alpha(t-A[h(y)])$
4. $A[h(y)]=t$
5. 对于每个item y，采样概率为$1/B[h(y)]$

知乎 @努力搬砖的小李

为了解决hash collision的问题，可以建立多个数组 A_i, B_i，最终在多个数组中取最大：

算法三：改进的多元数组-频率估计算法

输入： 学习率α，大小为H的数组$\{A\}_{i=1}^{m},\{B\}_{i=1}^{m}$，哈希函数$\{h\}_{i=1}^{m}$

1. 对于每一步$t=1,2,\ldots$:
2. 对于每个item y:
3. $B_i[h(y)]=(1-\alpha)B_i[h(y)]+\alpha(t-A_i[h(y)])$
4. $A_i[h(y)]=t$
5. 对于每个item y，采样概率为$1/max_i\{B_i[h(y)]\}$

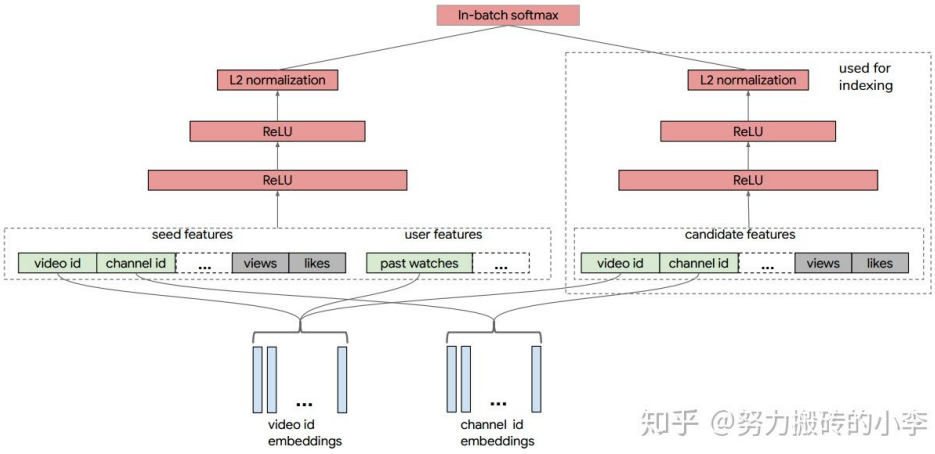
知乎 @努力搬砖的小李

模型架构

利用双塔模型训练，对YouTube的视频推荐，模型架构如下图所示。

- 训练标签：当点击了video并观看完整，则 $r_i=1$,否则 $r_i=0$
- 视频特征：视频 id，频道id等，特征转化为embedding，对于多值类别时，对embedding加权平均

▲ 赞同 46 ▼ 4 条评论 分享 喜欢 收藏 申请转载 ···



参考文献：

Yi X , Yang J , Hong L , et al. Sampling-bias-corrected neural modeling for large corpus item recommendations[C]// the 13th ACM Conference. ACM, 2019.

编辑于 2020-05-02

机器学习 推荐系统 深度学习（Deep Learning）

推荐阅读

论文 | 从DSSM语义匹配到Google的双塔深度模型召回...

Thinkgamer

【推荐系统经典论文(十)】阿里SDM模型

SDM: Sequential Deep Mmatching Model for Online Large-scale Recommender Systemm背景介绍文章核心思想?用户存在短期偏好与长期偏好，文章认为，一方面，在一个session...

努力搬砖的小李

多目标

主要调研
Represe
Assisted
Predicti
https://a
阿里提出

雪的味道

4 条评论 切换为时间排序

写下你的评论...

fendouxiaoquan 2020-11-12

厉害，理解了那块的证明

赞

凯风 2020-09-30

，按照它的描述，

▲ 赞同 46 ▼ 4 条评论 分享 喜欢 收藏 申请转载 ···