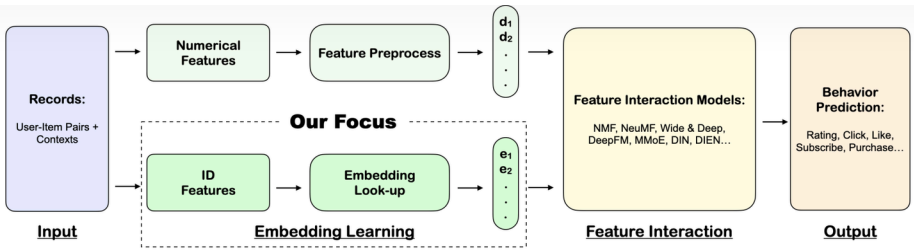
知乎



shopee 2023 -CEL：ID embedding自动聚类，助力解决用户和物品冷启动问题

SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我
已关注

17 人赞同了该文章

Introduction

为了解决冷用户和冷物品问题，一种常见做法是训练嵌入时加入正则化，如强制所有嵌入遵守给定的聚类结构。但固定预定义的聚类结构难以优化，且现有方法需在全嵌入上训练，内存开销大。为降低内存成本，通常采用哈希将实体映射到固定大小的嵌入表。然而，强制随机哈希会导致性能不佳，学习到的哈希方法仍有提升空间。

此外，预训练哈希函数或绑定每个哈希桶到热用户会使这种方法失去灵活性。CEL是一种动态自动聚类的解决方案，通过共享嵌入降低内存成本。它采用分层法进行聚类，并参考相对嵌入的梯度进行细分。优化后的集群需要进行自动重排，可通过最小化直接损失来完成。该框架适用于零数据启动和自动调整簇数和排序。

理论与实验验证了提出的聚类有效性因子(CEL)的性质：唯一性和存在最优簇数。同时，CEL-Lite具有最优时间复杂度，使其在线更新更为高效。多项实验证明了CEL在真实世界数据集及商业数据集上的优越性能，并且将CEL集成到商业模型中可带来+0.6%的AUC提升和显著收入增长，同时嵌入表的大小也减少至2650倍。

The Proposed Framework

Clustered Embedding Learning (CEL)框架可以与多种差异特征交互模型集成，如NMF, NeuMF, Wide&Deep, DeepFM, MMoE, DIN, DIEN。通过图展示嵌入学习在推荐系统中的作用。

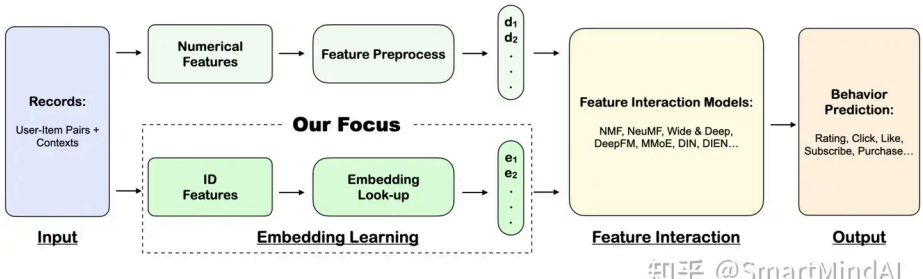


Figure 1: The standard behavior prediction paradigm in recommender system.

设用户-物品交互数据矩阵为
$\mathbf{X} \in \mathbb{R}^{N \times M}$,
其中每个元素代表用户对特定项目的交互（评分、点击等）。目标是找到两个低秩矩阵
$\mathbf{A} \in \mathbb{R}^{N \times R}$
$\mathbf{B} \in \mathbb{R}^{M \times R}$,
使得
$\mathbf{X} \approx Y(\mathbf{A}, \mathbf{B})$,
其中$\mathbf{A}$和$\mathbf{B}$分别表示用户和项目嵌入。 每行$\mathbf{A}$或$\mathbf{B}$对应一个用户的嵌入。R为嵌入维度。 预测
$Y(\mathbf{A}, \mathbf{B}) = [y_{ij}] = \mathbf{y}(\mathbf{A}_i, \mathbf{B}_j) \in \mathbb{R}^{N \times M}$

化问题。

$$\arg\min_{\mathbf{A},\mathbf{B}} \mathcal{L},\ \mathcal{L} = \left\|\mathbf{W} \odot (\mathbf{X} - Y(\mathbf{A},\mathbf{B}))\right\|_F^2,\ \hat{\mathbf{W}} = \mathbf{C}_i^T \cdot \mathbf{U}^T$$

其中，$\mathbf{C}_i$是物品i的聚类向量，$\mathbf{U}$是所有物品的潜在表示矩阵。

$$\mathbf{B} = \mathbf{S}_q\mathbf{B}_q,\ \ q = \{0,1,2,\ldots\},$$

$\mathbf{S}_q$是当前的簇分配矩阵，由0和1构成；$\mathbf{B}_q$是当前的聚类嵌入矩阵。M_q表示当前簇的数量，q表示当前分裂的次数。在硬聚类分配中，每一行的$\mathbf{S}_q$都只有一个1，其余位置为0。

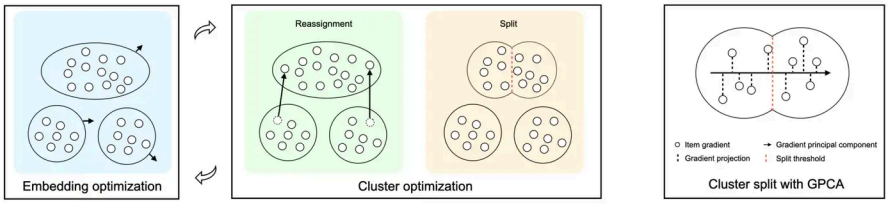


Figure 2: (a) An overview of CEL: it alternatively performs embedding optimization and cluster optimization. (b) The illustration of GPCA: it splits clusters along the first principal component of item gradients.

Embedding Optimization

$$\begin{aligned}\mathcal{L}_k = \big\|\mathbf{W} \odot \big[&\mathbf{X} \cdot \mathrm{diag}(\mathbf{S}_q(:,k)) \\ &- Y(\mathbf{A}, \mathbf{S}_q(:,k)\mathbf{B}_q(k,:))\big]\big\|_F^2,\end{aligned}$$

冒号$(:)$用于切片操作，$\mathbf{diag}(\cdot)$将向量映射为对角矩阵+。 $\mathcal{L}$是由$\{\mathcal{L}_k\}_{k=1}^{M_q}$的和构成的。

Cluster Optimization

提出两种聚类优化操作：聚类重新分配和聚类分裂。它们的每步嵌入优化分别为t_1和t_2。

Cluster Reassignment

项与簇交互有数据，簇内嵌入聚类。簇嵌入对项适应度衡量自然。目标是直接最小化项对簇赋值损失。

$$\mathbf{S}_q = \arg\min_{\mathbf{S}_q} \left\|\mathbf{W} \odot (\mathbf{X} - Y(\mathbf{A}, \mathbf{S}_q\mathbf{B}_q))\right\|_F^2.$$

对于固定的嵌入情况，每个项目的重新分配都是独立的。

$$\mathbf{S}_q(j,:) = \arg\min_{\mathbf{S}_q(j,:)} \left\|\mathbf{W}(:,j) \odot (\mathbf{X}(:,j) - Y(\mathbf{A}, \mathbf{S}_q(j,:)\mathbf{B}_q))\right\|_F^2.$$

Cluster Split with GPCA

$$\mathbf{p} = \arg\max_{\|\mathbf{p}\|_2=1} \mathbf{p}^\top \mathbf{G}^\top \mathbf{G}\mathbf{p}.$$

将集群分为两组，根据第一主成分得分$g_j^\top \mathbf{p}$

$$\begin{cases}\mathbf{S}_{q+1}(j,k) = 1, & \text{if } g_j^\top \mathbf{p} < \delta; \\ \mathbf{S}_{q+1}(j, M_q+1) = 1, & \text{if } g_j^\top \mathbf{p} \geq \delta.\end{cases}$$

调整阈值δ以平衡簇中物品的数量是通过CEL伪代码+实现的。

Theoretical Analysis of CEL under NMF

知乎

（CEL的可识别性）假设数据矩阵$\mathbf{X}$符合

$$\mathbf{X} = \mathbf{A}\mathbf{B}_q^\top \mathbf{S}_q^\top,$$

其中

$\mathbf{A} \in \mathbb{R}^{N\times R}$，$\mathbf{B}_q \in \mathbb{R}^{M_q\times R}$，$\mathbf{S}_q \in \{0,1\}^{M\times M_q}$，$\|\mathbf{S}_q(j,:)\|_0 = 1, \forall j \in [1,M]$，
$rank(\mathbf{X}) = rank(\mathbf{A}) = R$，$\mathbf{B}_q$ 没有重复行，并且不失去一般性地满足 $M_q \geq R$。

如果 $\mathbf{S}_q$ 的列满秩并且 $\mathbf{B}_q$ 的行足够分散，则 $\mathbf{A}$、$\mathbf{B}_q$ 和 $\mathbf{S}_q$ 基本上是唯一的。

在我们的实现中，强制这种条件改善了模型性能。假设$\mathbf{B}_q$足够分散是常见的，并且很可能满足于$\lambda(\mathbf{B}_q) > \lambda(\mathbf{S}_q)$。根据定理，我们可以进一步推导出唯一最优聚类数量的存在。

（CEL最优簇数）在 [定理：CEL可识别性]的假设下，并进一步假设数据矩阵还可以分解为

$$\mathbf{X} = \mathbf{A}'\mathbf{B}_{q+1}^\top \mathbf{S}_{q+1}^\top,$$

其中 $\mathbf{A}' \in \mathbb{R}^{N\times R}$，$\mathbf{B}_{q+1} \in \mathbb{R}^{M_{q+1}\times R}$，$\mathbf{S}_{q+1} \in \{0,1\}^{M\times M_{q+1}}$，
$\|\mathbf{S}_{q+1}(j,:)\|_0 = 1, \forall j \in [1,M]$，$rank(\mathbf{A}') = R$，

并且 $M_{q+1} > M_q \geq R$。

如果 $\mathbf{S}_{q+1}$ 的列满秩并且 $\mathbf{B}_{q+1}$ 的行足够分散，则 $\mathbf{B}_{q+1}$ 只有 M_q 种不同的行，可以通过对 $\mathbf{B}_q$ 中的行进行排列和缩放来获得。

根据CEL的唯一性，$\mathbf{S}_{q+1}\mathbf{B}_{q+1}$等于$\mathbf{S}_q\mathbf{B}_q$的置换和缩放。这意味着存在唯一的最优聚类数以及一旦达到该数，继续分裂是不必要的，因为重复的聚类嵌入会引入。鉴于此，我们在分裂时始终将新分裂集群的嵌入初始化为已选择的集群以保留前一解决方案的优化性。在线学习环境下，随着最优聚类数的增加，分裂变得更有动机，使得CEL成为一个可持续的学习框架。

CEL-Lite for Efficient Online Update

Online Learning.

现代推荐系统通常处理大量数据，所以它们往往是在线更新的：随着时间的推移，数据不断涌入，并在积累到一定程度时进行更新。这个框架遵循在线学习范式，使得模型可以快速适应新来的数据。 我们通过优化批量数据处理效率将CEL改编为CEL-Lite，使之适用于在线学习环境。CEL-Lite具有理论上最优的时间复杂度，并继承了CEL的许多优点。当我们可以获取完整的数据但又希望在线训练模型时，CEL-Lite也是适用的。

Data and Embedding Optimization.

在线环境中，新的批次 $\mathbf{X}_b$ 由用户-项目交互集 $\{\mathbf{X}(i,j)\}$ 构成。为了解决"灾难性遗忘"问题，我们从每个用户的最新交互中缓存 $\leq n$ 个交互项 (每项为用户 ID、项目 ID 和评分的三元组) 进行经验重放，这是在线系统常用的方法。由于每个交互都以用户 ID、项目 ID 和评级的三元组表示，因此存储开销是可接受的。此外，这些交互可以存储在分布式文件系统中，而不是像嵌入或模型参数那样必须存储在内存中以实现高效更新。在 CEL-Lite 中，每次优化一个批次的嵌入时都会执行随机梯度下降，并使用相关的缓冲交互。总体来说，嵌入优化的时间复杂度为 $\mathcal{O}(nDR)$，其中 D 表示总交互数 $\|\mathbf{W}\|_0$。

Efficient Cluster Reassignment.

只考虑每轮所需的新随机采样集群数和当前批次中的物品重新分配。这些策略只能局部优化集群分配。然而，随着足够多的更新，可以收敛到全局最优。实验显示，这大大提高了效率，同时仅引起微小的性能下降。每轮重分配的时间复杂度为$\mathcal{O}(nmbR)$，其中b是批量大小，假设存在D/b批，最多有D/b个集群需要重新分配。因此，总时间复杂度为$\mathcal{O}(nmDR)$。

如果一个项目有超过 d 次交互，它会直接分裂并单独形成一个集群。[[策略：1]]

只有当一个集群拥有超过 $2d$ 个相关交互（$\mathbf{WS}_q(:,k)||_0$）时才会分裂。拆分应保持平衡，使得结果集群之间的关联交互数量差不超过 d。 [[策略：2]]

策略 用于处理大型项目的极端情况。该策略确保每个聚类之间的交互次数不超过d。为了平衡分割并调整阈值δ，可以轻松实现。 使用策略 和，每个集群至少包含$d/2$的交互。因此，最多可有$2D/d$个聚类，总计最多有$2D/d$个GPCA分裂。每项只考虑其n个最相关的交互，计算梯度所需的时间复杂性约为$\mathcal{O}(n\|\mathbf{S}_q(:,k)\|_0 R)$，这是$\mathcal{O}(2ndR)$的上界。总的分裂时间复杂性为$\mathcal{O}(4nDR)$。 我们通过将$d$设置为一个常数来简化并方便实际操作。实际上，它可以随着D的增长而以一种与D单调非递增的子线性函数*进行扩展，但不会影响整体的时间复杂性。以上策略允许最大簇的数量根据D的数量增加，从而实现了我们的理论结果在Section所述。

The Optimal Time Complexity

当n和m为常数时，CEL-Lite的优化算法时间复杂度均为$\mathcal{O}(DR)$，故整体时间复杂度也为$\mathcal{O}(DR)$。需要注意的是，如果需要处理所有数据并每次调用都会更新嵌入，则时间复杂度将为$\mathcal{O}(DR)$。因此，$\mathcal{O}(DR)$可视为嵌入学习的下界时间复杂度。

Experiments

Cel与嵌入学习基准的比较：Vanilla full embedding, JNKM, eTree, Modulus, BH, AE, HashGNN和PreHash。在不同特征交互模型上进行测试，如NMF、DeepFM等，并考虑了MovieLens、Video Games and Electronics、One-day Sales三个数据集。eTree和JNKM在簇内不共享嵌入，所以进行了调整，以便在测试阶段叶节点与父节点共享嵌入。同样，我们也调整了JNKM，使其集群共享其聚类的嵌入在测试阶段。PreHash有超参数K，用于选择每个用户或项目的前K个相关簇。对于上述内容，我们重新定义了M_q，表示eTree中的父节点数量，JNKM中的簇的数量，hash方法中的哈希桶的数量。我们将M_q/M称为"压缩比"。

Table 1: Datasets summary.

Dataset	Users	Items	No. interactions
MovieLens-1m	6, 040	3, 952	1, 000, 209
Video Games	826, 767	50, 210	1, 324, 753
Electronics	4, 201, 696	476, 002	7, 824, 482
One-day Sales	≈ 13m	≈ 40m	≈ 700m

Performance Versus Baselines

Feature Interaction Model				NMF				DIN			
Compression Ratio				100%	5%	1%	0.5%	100%	5%	1%	0.5%
	LTC	LMC	SIT								
Vanilla full embedding	✓	✗	✓	0.8357	-	-	-	0.7903	-	-	-
eTree	✗	✗	✗	0.7841	0.8399	0.8482	0.8614	0.7640	0.8306	0.8379	0.8470
JNKM	✗	✗	✗	0.7626	0.8903	0.8616	0.8498	0.7486	0.8868	0.8560	0.8466
Modulo	✓	✓	✓	-	1.0227	1.0688	1.0732	-	1.0206	1.0647	1.0692
BH	✓	✓	✓	-	0.9753	1.0170	1.0526	-	0.9907	1.0040	1.0495
AE	✓	✓	✓	-	0.9959	1.0343	1.0630	-	0.9895	1.0280	1.0624
HashGNN	✗	✗	✗	-	0.8491	0.8829	0.8882	-	0.8467	0.8663	0.8723
PreHash ($K = 4$)	✓	✓	✗	-	0.8169	0.8832	0.8987	-	0.8057	0.8678	0.8897
PreHash ($K = M_q$)	✗	✓	✗	-	0.7958	0.8047	0.8101	-	0.7871	0.7893	0.7942
CEL (Ours)	✗	✓	✓	**0.7507**^{p}	**0.7784**	**0.7858**	**0.7926**	**0.7513**^{p}	**0.7713**	**0.7767**	**0.7808**
CEL-Lite (Ours)	✓	✓	✓	**0.7519**^{p}	**0.7820**	**0.7901**	**0.8014**	**0.7421**^{p}	**0.7766**	**0.7789**	**0.7906**

Online Conversion Prediction

该研究旨在预测用户是否会执行可接受的操作，以促进他们的行为，如点击和购买。使用的数据集包括视频游戏和电子设备数据集。ROC曲线*下面积（AUC）被用来评估性能，并发现HashGNN

CEL-Lite可以显著提高转化预测的性能。

Table 3: The test AUC (presented in % for a better interpretation) on Video Games and Electronics datasets.

Dataset	Video Games								Electronics							
Feature Interaction	DeepFM				DIN				DeepFM				DIN			
Compression Ratio	100%	10%	5%	2%	100%	10%	5%	2%	100%	10%	5%	2%	100%	10%	5%	2%
Vanilla full embedding	68.19	-	-	-	70.73	-	-	-	66.95	-	-	-	67.62	-	-	-
Modulo	-	61.68	58.81	56.34	-	63.35	61.94	60.25	-	63.13	54.54	48.61	-	63.70	61.69	58.93
HashGNN	-	68.51	64.43	61.84	-	71.59	71.11	70.82	-	64.49	61.41	59.45	-	65.50	63.49	60.40
PreHash ($K = M_q$)	-	69.19	66.21	63.89	-	72.72	71.63	71.46	-	67.50	6[illegible]	66.9[illegible]	-	70.[illegible]	68.8[illegible]	[illegible].17
CEL-Lite	**70.85**^{p}	**70.33**	**70.21**	**69.82**	**73.85**^{p}	**73.80**	**73.50**	**73.39**	**68.98**^{p}	**68.90**	**68.69**	**68.03**	**71.63**^{p}	**71.56**	**70.18**	**69.81**

Integrated with Business Model

我们通过将CEL-Lite集成到一个复杂的企业级模型中，在Shopee电商平台上进行了实验，以验证其性能。结果显示，相较于使用全向嵌入的企业级模型，使用CEL-Lite的小型簇可以获得更好的性能，尤其是当嵌入表的大小是全向嵌入的2650倍时。此外，使用压缩的企业级模型也能获得更好的性能，使得AUC提高了0.61%。需要注意的是，在实践中，对于转化率任务来说，绝对AUC的增加0.1%被认为是重要的。

Table 4: The test AUC (%) on One-day Sales dataset.

Dataset	One-day Sales			
Embedding size	Full	1m	100k	10k
Business Model	87.09	86.80	86.55	86.46
CEL-Lite	-	-	**87.78**	**87.70**

Personalization and Cold-start Problems

Personalization.

在使用共享嵌入进行训练时，可以选择在特定点解绑聚类分配，为每个用户和物品学习一个个性化嵌入。解绑后，可以根据历史数据进一步优化嵌入。完全的嵌入需要足够的数据和内存，此处将此过程称为"个性化"，类似联邦学习的个性化。有各种个性化技术，如我们在后续工作中将详细探讨。在此，我们简单地在原始目标中添加距离正则化来引导个性化。

$$\mathcal{L}_p = \frac{\lambda_p}{2} \sum_{j=1}^{M} \|\mathbf{B}(j,:) - \mathbf{S}_q(j,:)\mathbf{B}_q\|_2^2.$$

实验设置λ_p为50，表1表明个性化后的CEL方法在最佳情况下效果最好，且优于所有其他全嵌入方法。

Mitigating Cold-start Problems.

使用MovieLens-1m数据集，压缩比为1%。结果表明，CEL在冷项目上的性能优于PreHash。这可能是因为PreHash倾向于暖项目，而冷项目通常是由暖项目的一个加权平均值组成的。在区间$[800, 2000]$内，PreHash的性能稍好。然而，如果我们对CEL进行个性化，冷项目的表现会下降。这表明冷项目不能仅仅通过自己的数据进行优化，而需要CEL的帮助。此外，个性化在温暖项目上的效果显著提高，总性能更好。最后，只针对温暖项目进行个性化的结果显示，测试MSE提高了1.2%。

Sparsity and Compression Ratio.

创建一个稠密的MovieLens-1m数据集，通过过滤掉小于100次的交互得到。接着，创建一个稀疏的MovieLens-1m数据集，通过对每个用户随机删除50%的交互进行随机丢弃。实验结果表明，无论在哪种情况下，增加聚类数量都能提高CEL的整体性能；然而，当压缩比超过某个阈值时，增加聚类数量反而会导致性能下降。此外，在稠密和原始数据集上，增加聚类数量可以显著提升个性化性能，而在原始和稀疏数据集上，个性化则存在一个优化的压缩比范围为1%至2%。总体来

知乎

Ablation Studies on Clustering

Interpretation of Clustering.

为了评估CEL如何及在多大程度上将项目聚类在一起，我们使用MovieLens-1m的$M_q = 10$级别数据集可视化了CEL聚类结果。结果显示，这些聚类主要根据电影的评级进行划分，且各集群内的标准差较小，因为我们的任务是预测评级。 此外，我们还观察到电影在某些聚类中呈现出内在的相关性。例如，集群6主要包含动作/科幻电影；集群5主要包含适合儿童的音乐/动画电影；集群9主要包含带黑帮元素的犯罪电影；而集群8则包含了Stanley Kubrick执导的大部分电影（约为70%）。这些关系并非事先输入给模型，而是由CEL通过自动学习获得。 进一步的分析表明，与其它方法相比，CEL在区分不同类型电影方面的性能更好。对此，我们对每个集群中不同类型电影的平均熵进行了计算，以进一步验证这一结论。 令人惊讶的是，我们发现同一类型的电影可能会形成多个具有不同平均评分的聚类。例如，集群4和6都是主要的动作片，但它们的评分却存在明显差异。

Rule-based Clusterings

比较CEL与其他两种基于规则的聚类方法（类型和评级）。结果表明这两种规则方法无效。同时，尝试将这些规则方法用作CEL的初始化，发现效果不佳。

Split Methods

我们比较了随机分割、Bisecting 2-Means、随机投影和GPCA等四种分隔方法。这些方法包括随机划分，它通过随机选择物品将它们分为两组；Bisecting 2-Means，它是基于DHC的一种高级分隔方法，它根据物品之间的距离将物品分为两个簇；随机投影，它是类似GPCA的方法，但在投影梯度时而不是在第一个主成分方向上进行。最后是GPCA。Bisecting 2-Means可以在梯度矩阵或用户启发式表示上执行。对于所有方法，我们都将其压缩到1%。 结果已经在表格中展示。从结果可以看出，随机分割产生的效果最差，这强调了分裂方法的重要性；Bisecting 2-Means在梯度上的表现优于用户启发式的表示，这表明梯度更具信息性；Bisecting 2-Means和随机投影的梯度比GPCA更有效，这表明按照第一主成分分隔的合理性。

Conclusions

本文提出了一种名为Clustered Embedding Learning (CEL)的新颖嵌入学习框架，可在任何不同特征交互模型中插入使用，同时提高性能并减少内存消耗。理论证明了CEL的识别性，而CEL-Lite则达到最优的时间复杂度。大量的实验验证表明，CEL在实际推荐系统中具有强大实力，特别是针对用户和项目等嵌入特征。此外，CEL也可适用于其他ID特征或分类特征，如商店ID、卖家ID、搜索词、对象类别和地理位置。这方面的研究将在后续工作中展开。

原文《Clustered Embedding Learning for Recommender Systems》

关注我，追踪最新技术

编辑于 2024-02-19 14:45 · IP 属地北京

Shopee 推荐系统 K – 均值聚类