Rank-GeoFM: A Ranking based Geographical Factorization

Method for Point of Interest Recommendation

摘要--随着基于位置的社交网络的快速增长，兴趣点（POI）推荐已成为一个重要的研究问题。 然而，登记数据的稀缺性（一种隐式反馈数据）对现有POI推荐方法提出了严峻挑战。此外，有关POI的不同类型的上下文信息是可用的，如何利用它们成为另一个挑战。在本文中，我们提出了一种基于排名的地理因子分解方法，称为Rank-GeoFM，用于POI推荐，它解决了这两个挑战。在所提出的模型中，我们认为登记频率表示用户的访问偏好并通过正确地对POI进行排序来学习因子分解。在我们的模型中，无论是签入还是不签入，POI都将有助于学习排名，从而缓解数据的稀疏性问题。此外，我们的模型可以很容易地结合不同的语境信息的类型，如地理影响和时间影响。我们提出了一种基于随机梯度下降的算法来学习分解。在用户POI设置和用户时间POI设置下对公开可用数据集进行了实验，以测试所提方法的有效性。两种设置下的实验结果表明，所提出的方法优于最先进的方法。在推荐准确性方面显着。

1. **INTRODUCTION**

最近，基于位置的社交网络（LBSN）已经出现，例如Foursquare。这些在线系统使人们能够在他们访问兴趣点（POI）（例如餐馆和购物中心）时登记并与朋友分享他们的体验。这些网络以前所未有的速度增长。以Foursquare为例，截至2014年1月，它吸引了超过50亿次签到的4500万用户。大量数据包含有关POI和人类偏好的有价值信息，可用于POI推荐[1]。

POI推荐旨在了解用户的访问偏好并向用户推荐他/她可能感兴趣但从未访问过的POI。 这项任务非常重要且有意义，因为它不仅可以帮助当地居民或游客探索城市中有趣的未知地点，还可以为POI所有者创造机会，通过寻找和吸引潜在的访客来增加收入。

由于两个原因，POI推荐具有挑战性。首先，LBSN中的登记数据非常稀疏，因此推荐方法存在数据稀缺问题。登记数据通常表示为user-POI矩阵，如图1（a）和1（b）所示。正如我们在实验中所看到的，登记矩阵的密度通常小于0.5％。此外，当考虑上下文感知POI推荐时，用户POI登记数据（矩阵）需要被分离并表示为张量，例如，如图1（c）和1（d）所示，用于时间感知 POI推荐。这将使数据更加稀疏，并且实验中的登记张量密度小于0.05％，与Netflix数据的1.2％相比非常小[2]。更糟糕的是，签到是一种隐式反馈[11]，这使得POI推荐更加困难。与传统的电影评级数据不同，用户明确表示他们对具有不同评级分数的项目“喜欢”或“不喜欢”，签到仅提供用户喜欢的正面示例，以及没有签到的POI，标记为 图1（a）和1（c）中的“？”要么没有吸引力，要么未被发现但可能具有吸引力。换句话说，我们需要根据登记数据推断他/她的偏好和非偏好。大多数现有的POI推荐方法[22,3,14,9,7,12]忽略了数据稀缺性和隐式反馈事实，并使常规内存或基于模型的协同过滤适用于POI推荐。因此，这些方法遭受数据稀缺问题。

第二，在POI建议中，可以获得不同类型的上下文信息，例如，POI的地理坐标、签入时间戳、用户的友谊、POI的类别等。利用上下文信息提高推荐精度是非常重要的。例如，地理坐标，作为传统中不可用的重要类型的上下文信息推荐任务被用于POI推荐[22,15,3]，因为用户访问附近POI的概率要高得多[22,15]。以前的工作[22,3,12,11]开发了不同的方法来利用不同类型的上下文信息。 然而，这些方法通常是针对特定类型的上下文开发的，并且很难将它们概括为处理另一种类型的上下文信息。

在本文中，我们提出了一种新的基于排序的POI推荐问题的因子分解方法。 我们考虑以排名的方式获得因子分解，原因有两个：一方面它对隐式反馈数据更有效[16]，另一方面，稀疏性问题可以得到缓解，因为访问（正例）和未访问的POI都将有助于学习排名功能。具体地，我们假设登记频率越高，用户优选的POI越多; 未访问的POI不如访问过的POI更受欢迎。我们不是像传统的基于矩阵分解的方法那样拟合登记频率，而是根据用户对POI的偏好排名来学习用户和POI的潜在因素。在我们提出的方法中，未访问的POI也有助于学习，这将有助于缓解稀疏性问题。

除了解决数据稀缺问题，我们的分解模型可以轻松地包含不同类型的上下文信息。在所提出的方法中，我们通过**它与所考虑的每个上下文变量之间的成对交互分数的总和来计算POI的推荐分数**。这使我们能够通过简单地包括更多交互分数来轻松地合并不同类型的上下文信息。成对预测还统一了处理矩阵和张量数据的方式。也就是说，可以使用一个统一框架来处理POI推荐和上下文感知POI推荐。

我们在这份文件中的贡献可归纳如下：

* 我们针对POI推荐提出了一种新的基于排名的地理因子分解方法（称为Rank-GeoFM），该方法解决了数据稀缺性问题，并结合了地理影响，这是一种重要的上下文信息。
* 为了优化Rank-GeoFM的目标函数，我们使用随机梯度下降法。 然而，计算随机梯度下降存在两个困难。首先，目标函数是无差别的。为了解决这个问题，我们提出了对目标函数的连续近似。其次，为了计算随机梯度，我们需要计算所有POI的预测分数，这需要大量计算。为了解决这个问题，我们开发了一种基于抽样的快速学习方案来计算梯度。
* 当其他类型的上下文信息可用时，该方法易于推广。我们以时间信息为例，演示如何使用我们的模型来合并用于时间感知的POI推荐。
* 对两个真实数据集的广泛实验Foursquare和Gowalla证明，就准确性而言，Rank-GeoFM在POI推荐和时间感知POI推荐方面都优于最先进的方法。

本文的其余部分安排如下：在第二节中，我们简要回顾了相关的工作。在第三节中，我们介绍了所提出的方法。第四节给出了实验结果。最后，我们在第五节中总结了这篇论文。

**2. RELATED WORK**

**2.1 POI Recommendation**

**POI Recommendation.** 基于存储器的协同过滤（CF）技术，例如基于用户的CF和基于迭代的CF，被用于POI推荐。Ye等[22]采用线性插值将社会和地理影响纳入基于用户的CF框架中，用于POI推荐。他们的实验结果表明，基于用户的CF优于基于项目的CF用于POI推荐，将地理影响结合到基于用户的CF模型中可以显着提高推荐准确性，并且社会影响对性能的影响很小。 Levandoski等人通过将行程距离视为惩罚来扩展基于项目的CF方法[9]。

基于存储器的CF方法容易遭受数据稀疏性问题，因为用户-用户或项-项相似性需要基于公共检查来计算。当登记人稀少时，有两个用户或项目将共享很少的共同签入，因此，由此产生的相似性是不可靠的，以作出有效的建议。

基于模型的CF技术也被用于POI推荐。Noulas等[14]发现矩阵因子分解（MF）比基于用户的CF和基于迭代的CF用于POI推荐表现更差。在他们的工作中，用于显式反馈数据的传统MF方法被应用于POI推荐，因此不适合并且提供不良性能。基于个人登记位置通常围绕几个中心的观察，Cheng等[3]引入了一个多中心高斯模型来计算地理影响，然后启发式地将其与MF相结合以用于POI推荐。在这种方法中，MF仅通过拟合非零检查来执行，因此容易受到数据稀疏性问题的影响。 Liu等[12]提出了一种地理概率因子分析框架，即GTBNMF，它结合了基于贝叶斯非负矩阵分解（BNMF）的地理影响和文本影响。但是，BNMF是通过拟合零和非零签入来执行的，这可能不合理，因为零签到可能是缺失值而不应直接拟合。 所有这些因式分解方法都没有利用POI推荐的隐式反馈属性。

最近，考虑到检查是隐式反馈，Lian等[11]基于加权矩阵分解（WMF）开发了一个模型，即GeoMF，并将地理影响纳入WMF。该方法通过使用较大权重和使用较小权重的零检查来适合非零签入。虽然分配大权重可以突出非零签到，但直接适合零签到可能不是很合理，因为零签到可能是缺失值。而且，由于WMF的限制，将该方法概括为其他类型的上下文信息并不容易。

**Context-aware POI recommendation.** 上述针对POI推荐的方法中的大部分利用了在提出建议中的地理影响。然而，他们处理地理影响的方法不能处理其他类型的上下文信息。时间是另一种重要的上下文类型，时间感知POI推荐旨在为用户在给定时间推荐POI。 Yuan等人提出了一种名为UTE + SE的方法，该方法扩展了基于用户的CF，将时间和地理效应与线性组合框架相结合[24]。Yuan等进一步提出了一种基于图形的方法，称为BPP，用于时间感知POI推荐[25]，它通过偏好传播在由登记数据构建的图形上进行推荐。BPP [25]的表现优于UTE + SE [24]。此外，Gao等[7]研究了对POI推荐的时间效应，但没有研究时间感知的POI推荐。 他们开发了一种正则化的非负矩阵分解方法，但他们没有考虑地理影响。

在POI推荐中使用的其他类型的上下文包括POI的类别[13]，POI的文本描述[23,12,10,8,26]和当前访问的POI [4,6]。

**2.2 Ranking based Learning Criteria(基于排序的学习准则)**

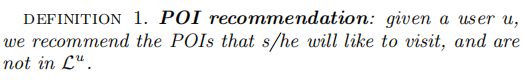
贝叶斯个性化排序（BPR）[16]是一种着名的基于排序的客观标准，它可以结合矩阵分解模型为隐式反馈问题产生有希望的性能。BPR基于项目的成对比较来学习排名模型，使得ROC曲线下面积（AUC）可以最大化。它给予每个项目对相等的权重[19,21]。有序加权成对分类（OWPC）[20]是另一个最近提出的排名损失度量。该方法将排名视为一组成对分类问题，并通过分配较高权重来强调前N位置的分类。OWPC已成功应用于文本检索[20]和图像标注[21]。

本文研究了基于OWPC准则的POI推荐方案。我们提出的方法在两个方面不同于现有方法[20,21]。**首先**，现有的OWPC被开发用于对具有二进制值的问题进行排序，即相关性或不相关性，而在本文中，我们扩展目标函数以对具有不同访问频率的POI进行排序，并提供随机梯度下降优化的解决方案。**其次**，我们开发了一种POI推荐的通用分解方法，它能够利用不同类型的上下文信息。

**3. PROPOSED METHOD(推荐方法)**

在这一部分中，我们提出了一种基于排序的POI推荐的因式分解方法。我们首先用一个排序目标函数来表示POI推荐问题，然后介绍了POI推荐问题。如何优化它。最后，对时间感知POI推荐模型进行了推广。

我们首先在表1中总结了本文中使用的符号。然后，POI推荐问题定义如下：



**3.1 Ranking based Geographical Factorization**

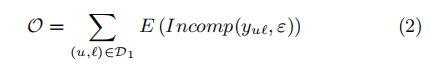
**3.1.1 Preference Ranking Objective Function**

在本小节中，我们制定了POI推荐的目标函数。 由于登记数据的稀疏性，我们通过拟合用户对POI的偏好排名来设计目标函数，而不是像传统的分解方法那样拟合他/她的登记频率。

首先，我们需要根据他/她的登记数据推断用户对POI的偏好排名。 直观地，我们假设登记频率越高，用户更喜欢POI越多; 未访问的POI不如访问过的POI更受欢迎。换句话说，对于给定的用户u，POIl的排名应高于POI l' xuℓ>xuℓ'，其中xuℓ表示用户你访问POIl的频率。

基于直觉，我们提出了一种度量推理排名与因式分解模型所产生的排名之间不相容性的方法。特别是对于给定的用户u和POIℓ，不兼容性可以通过以下方法来衡量：

其中I(·)是指示符函数，I(a)＝1，当A为真时，0否则为正数；yuℓ表示用户u的POI l推荐得分，将通过本文的因子分解模型计算得出。我们可以看到Eq（1）根据登记数据计算用户u应该排在低于l的POI数，但是通过分解模型排名高于l。注意，ε-余量用于计算分解模型的排名，即，只有当yuℓ<yuℓ'+ε时，我们认为l'对于用户u的排名高于l。Incomp（yuℓ，ε）测量用户u的错误排名高于ℓ的POI数量，我们称之为“排名不兼容”。

接下来，我们设计了我们的偏好排名目标函数，用于学习分解模型。 具体而言，一个好的分解方法应该尽可能地减少排名不兼容性，因此我们提出以下目标函数来最小化：

其中E(·)是一个函数，用于将排名不兼容的incomp(Yuℓ，ε)转换为亏损：



我们定义了E(0)=0。

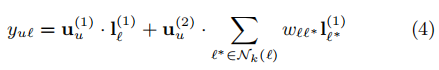
在方程（2）中，我们汇总了D1中所有用户POI对的损失，以计算总损失。我们注意到，根据方程（1），Incomp（yuℓ，ε）总是等于零 用户-POI对（u，l）̸∈D1。在OWPC [20,21]之后，我们采用平滑加权方案将Incomp（yuℓ，ε）转换为损耗。具体地，等式（3）中的函数E（r）体现了这种转换。可以看出，E（r）计算错误排序的POI的每个等级位置（从1到r）的损失总和，其中每个位置i被分配有损失1 / i。例如，假设我们有Incomp（yuℓ，ε）= 3，即三个POI错误地排名高于用户u的POI l。 因此，该对（u，l）的损失由E（3）= 1 + 1/2 +1/3给出。

我们目标函数的一个优点是它能够克服数据稀疏性问题。 根据方程（1），POI l对用户的排名不相容性由所有其他POI l'∈ L确定，这些POI大多是未访问的POI（因为用户经常访问非常少的POI）。因此，未访问的POI也有助于学习模型，而在传统MF中它们被忽略了。因此，利用目标函数可以解决CHE的稀疏性问题。录入数据，不直接拟合零检入。

**3.1.2 Geographical Factorization Method**

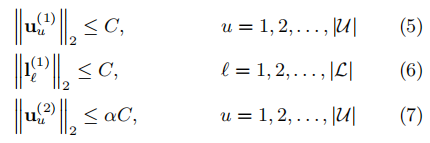
在这一小节中，我们提出了一种计算推荐得分的**地理因式分解方法**。我们的分解模型能够表征用户对POI的偏好。 此外，它还结合了地理环境对POI建议的影响。一方面，我们将用户和POI的潜在因子参数化为矩阵的K维空间；它们用于模拟用户自己的偏好，就像传统的矩阵分解方法那样。另一方面，我们为用户引入一个额外的潜在因子矩阵，并使用U（2）来模拟用户和POI之间的交互以结合地理影响。为此，我们进一步构造| L |×| L| 地理影响矩阵W，其中wlℓ'是POIl被访问的概率，因为已经访问了POI l'。按照先前的研究[22,3,12]，我们设定，否则为0。这里我们只考虑每个POIl的k-最近邻Nk（l）。 公式背后的直觉是用户通常倾向于访问附近的POI。我们对矩阵W的每一行进行归一化，使得，因为它代表了影响概率。

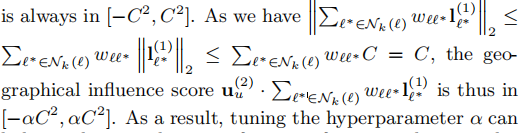
让表示我们的地理分解模型的参数。我们将在3.2节介绍学习这些参数的方法。假设已经学习了这些参数。给定用户u和POI l，我们计算推荐得分yuℓ如下：



其中符号·表示内积，表示矩阵U（1）的第u行。 类似的符号用于其他矩阵。在等式（4）中，第一项模拟**用户偏好分数**，而第二项模拟用户因其邻居而喜欢POI的**地理影响分数**。

为了避免过拟合问题，我们将模型的潜在因素约束为一个球，它作为正则化器[21]。具体来说，我们有以下限制：



其中C> 0且0≤α≤1是超参数。我们将潜在因子从U（1）和L（1）约束为半径为C的小球，并将潜在因子从U（2）约束为半径为αC的较小球。在这里，我们引入了超参数α来平衡用户偏好和地理影响分数的贡献。利用基本代数，我们得到了任意两个向量a和b的。因此，我们知道用户偏好评分总是在中。

因此，调整超参数α可以平衡用户偏好和地理影响分数对最终推荐得分的贡献。

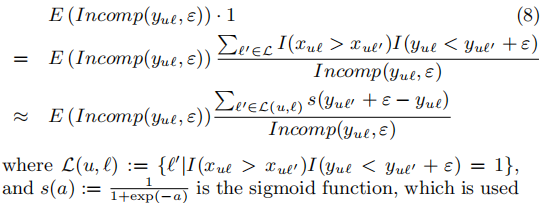
**3.2 Optimization and Learning Algorithm**

接下来，我们提出了所提出的学习模型参数Θ的方法，使得方程（2）中的O最小化。 由于O是通过对每个用户-POI对的损失求和来计算的，因此我们采用随机梯度下降（SGD）方法进行优化。也就是说，我们的目标是最小化E(incomp(Yuℓ，ε)，给出每个训练实例

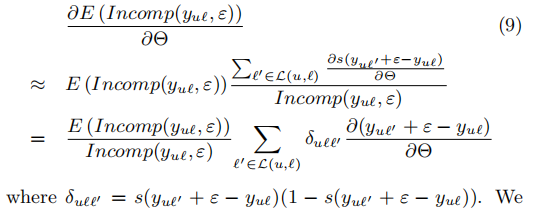
(u，ℓ)∈D1)。然而，有两个困难：(1)E(incomp(Yuℓ，ε)不连续、不可微，难以优化；(2)优化需要了解incomp(Yuℓ，ε)，但计算时间较长. 接下来，我们将介绍如何解决这两个问题。

**3.2.1 Continuous Approximation（连续逼近）**

为了使E(incomp(Yuℓ，ε)在参数Θ上连续，我们将其重写如下：



用于近似指示函数。在重写的基础上，可以计算Θ更新的随机梯度。具体来说，我们

我们注意到Eq（9）不是标准的梯度计算，因为E（Incomp（yuℓ，ε））和Incomp（yuℓ，ε）也与Θ有关，但我们不考虑它们的导数。我们的类比梯度计算遵循[21]中的思想。虽然随机梯度可以用方程（9）计算，但在实践中是不可行的。 这是因为方程（9）中的求和和Incomp（yuℓ，ε）都要求将所有POI的推荐得分计算为Eq（4），这需要花费O（K | L | k）运算并且是耗时的。

在下一小节中，我们将引入一个快速学习方案来解决这个问题。

**3.2.2 Fast Learning Scheme**

我们快速学习的关键思想是消除求和并用采样方法估计Incomp（yuℓ，ε）。

让我们首先重温等式(8)。我们可以从方程（8）中的第一个等式看出，只有错误排序的POI，即满足I(xuℓ>xuℓ') I(yuℓ<yuℓ'+ε) = 1的POI l，有助于损失E(Incomp（yuℓ，ε）)。因此，方程（8）可以被重新解释为对一组错误排序的POI样本所引起的损失的预期，其中每个POI样本l'引起损失：



并且每个POI样本的概率为1。这促使我们通过对一个不正确排序的POI进行抽样来近似计算随机梯度.在这种情况下，我们



这是方程（9）的近似值。显然，总和被消除了。

要通过方程（11）计算梯度，我们仍然需要知道Incomp（yuℓ，ε）。我们通过对一个不正确排序的POI进行采样来计算它的近似值。具体来说，给定一个用户-POI对（u，l），我们从L重复采样一个POI，直到得到错误排序的POI l'，使得I(xuℓ>xuℓ')I(yuℓ<yuℓ'+ε)=1。设n表示在获得这样的POI之前的抽样试验次数。显然，n遵循参数p = 的几何分布。显然，n服从参数p= 的几何分布。由于我们知道参数p的几何分布的期望为1/p，所以我们有n≈[1/p]= 。因此，我们可以估计Incomp（yuℓ，ε）≈。我们在这里的想法类似于[21]中用于另一个问题的观点。通过使用估计，我们通过以下方式重写方程（11）：



作为梯度的自适应缩放因子。当n很小时，梯度将具有大的幅度，这是合理的，因为小的n意味着Incomp（yuℓ，ε）很大，并且在这种情况下，参数应该用大步骤更新。类似地，当n很大时，我们将获得具有小幅度的合理梯度。 使用Eq（12），基于SGD的优化执行如下：



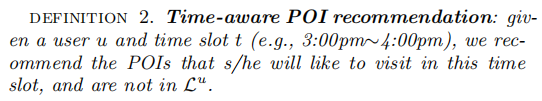
其中γ是学习率。

使用方程（12）中的梯度计算可以获得显着的加速。 方程（9）的复杂度为O（K | L | k），而方程（12）的复杂度为O（Knk）。一般来说，我们有n«| L | 在训练开始时，n <| L | 当训练达到稳定阶段时。开始时，模型没有经过良好的训练，因此Incomp（yuℓ，ε）通常很大，这导致非常小的n，即n«| L |; 当训练达到稳定阶段时，预计更多被访问的POI被正确排序，因此Incomp（yuℓ，ε）变小，n变得稍大。但是，每个访问过的POI都不太可能正确排名，因此通常我们仍然有n <| L |。 我们发现在我们的实验中加速是数量级的。

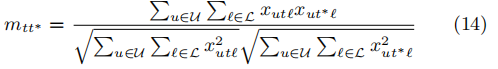
总结了算法1中提出的基于排序的地理因式分解方法(Rank-GeoFM).在该算法中，我们在d1中遍历所有的用户-poi签入对，并更新潜在的直到过程收敛为止(第3行∼16)。在每次迭代中，给定一个用户-PEI对，首先进行采样过程以估计Incomp（yuℓ，ε）并获得一个POI样本（第6~8行）。基于Incomp（yuℓ，ε）的估计 和采样的POI'，我们用SGD方法（第9~15行）更新相关的潜在因子。 检查更新的潜在因子的范数约束，并且预测违反约束的那些（第16行）。

3.3 Time-aware POI Recommendation

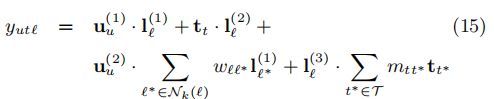
在本小节中，我们使用时态信息作为示例上下文来说明如何轻松推广我们的方法以合并其他类型的上下文。 我们考虑时间感知POI建议。



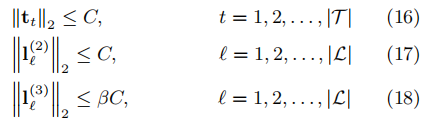
方程（2）中的目标函数可以很容易地扩展到包含时间上下文。为了计算推荐得分，我们扩展了方程（4），其中两个附加项捕获时间因子。一个术语是时**间流行度得分**，其指示该POI在该时隙中是否流行。另一个术语被称为**时间影响分数**，它基于在先前工作中进行的以下观察[24,25]：POI在一个时隙的流行度总是受到某些接近或类似时隙的影响，即，流行度在近或相似的时段之间相关。除了之外，我们还引入了三个附加项的潜在因子矩阵。具体而言，我们将时隙的潜在因子参数化为| T | ×K矩阵T; 并且针对POI参数化另外两个潜在因子矩阵L（2）和L（3），其中L（2）用于模拟与时间点的时间流行度得分的交互，并且L（3）用于模拟具有接近或相似时间的交互。 时间影响评分的时段。我们进一步构造一个| T | ×| T | 矩阵M，其中mtt \*是时隙t中POI的流行度得分受时隙t \*中的POI影响的概率。 通过以前的研究[24,7]，我们计算mtt \*为：



并且我们将M归一化为矩阵，使得每一行是概率向量。 给定用户u和时隙t，POI l的推荐得分计算如下：

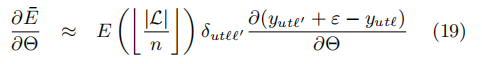


其中，四个术语分别表示用户偏好评分、时效性评分、地理影响评分和时间影响评分。我们用Θ={U(1)，L(1)，U(2)，T，L(2)，L(3)}表示因子分解模型的参数。与POI-用户设置类似，还施加了规范约束，以防止过度安装：



其中0≤β≤1是为了控制时间影响在计算推荐分数中的重要性。

算法1可以很容易地适应时间感知的POI推荐。具体来说，我们遍历D2中的所有用户时间POI元组以更新潜在因子。给定每个（u，t，l）∈D2，我们继续对POI进行采样，直到得到一个满足I（xutℓ>xutℓ'）I（yutℓ<yutℓ'+ε）= 1的POI l'，再次得到Incomp（yutℓ，ε） 估计为 [|L| /n]。随机梯度也可以通过以下方法计算：



。 在此基础上，相应的潜在因素得到相应更新。

**4. EXPERIMENTS**

我们对POI推荐和时间感知POI推荐进行了综合实验，评价了该方法的性能。

**4.1 Experimental Setup**

**4.1.1 Datasets**

我们在实验中使用了两个真实数据集[24] 1。 一个是2010年8月至2011年7月在新加坡制作的Foursquare登记数据，另一个是2009年2月至2010年10月期间在加利福尼亚州和内华达州制作的Gowalla登记入住数据。Foursquare数据包括由2,321个用户在5,596个POI进行的194,108次登记，Gowalla数据包括由10,162个用户在24,250个POI进行的456,988次登记。 每次登记都与时间戳相关联。

对于每个用户，我们将他/她最近签入的20％作为测试集来掩盖，以评估不同算法的性能。最早的70％签到用作训练集，剩余的10％签到用作验证集。基于训练集，我们构建了一个用户-POI矩阵X，分别为Foursquare和Gowalla提供了73,011和210,894个非零条目，这些条目将用于POI建议。对于Foursquare和Gowalla矩阵，X的密度分别为0.56％和0.085％。对于时间感知POI建议，我们将数据分成24小时（时间段），然后分别为Foursquare和Gowalla获得具有91,228和244,580非零条目的张量X. 两个数据的X密度分别为0.029％和0.0041％。 我们可以看到矩阵X和张量X都非常稀疏。

**4.1.2 Metrics**

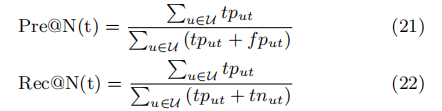
我们使用两个广泛使用的度量来评估不同推荐算法的性能，即精度@N和召回@N（由Pre @ N和Rec @ N表示），其中N是推荐POI的数量。

对于POI建议，给定用户u，我们计算PRE@N和Rec@N如下[22]：



其中tpu是算法产生的基础事实和前N个结果中包含的POI数量; fpu是算法中前N个结果中的POI数量，但不是基础事实; 和tnu是地面实况中包含的POI数量，但算法中的前N个结果中没有。报告的Pre @ N（Rec @ N）是所有用户的精确（召回）值的平均值[22]。

Pre @ N（Rec @ N）报告的是对于时间感知POI推荐的平均值，给定用户u和时隙t，我们让tput，f put和tnut成为tpu的时间特定扩展，f pu 和tnu，分别。 然后，在时间t，Pre @ N（t）和Rec @ N（t）计算如下[24]：所有用户的精确（召回）值[22]。



如[24,25]所示，然后通过平均所有时隙的精度（召回）值来报告平均Pre @ N（Rec @ N）。 对于这两个指标，我们分别在我们的实验中考虑N = 5,10和20（默认N = 5），因为最重要的建议更重要。

**4.1.3 Baseline methods**

对于POI推荐，我们将我们的模型与下面的基线方法进行比较。

* **UCF：**这是基于用户的CF方法，其中用户-用户相似度是根据签入数据计算的。
* **UCF+G：**该方法以线性插值的方式将地理影响纳入基于用户的CF[22]，这是POI推荐的一种有代表性的方法。
* **PMF：**概率矩阵分解[18]是为推荐系统开发的一种著名的因子分解.
* **BPR-MF**：由于我们的方法是基于排序的因子分解，我们还将BPR-MF视为基线，这是基于贝叶斯个性化排序标准的最流行的基于排名的矩阵分解[16]。注意此方法尚未评估在之前关于POI推荐的工作中。
* **GTBNMF**：如第2节所述，GTBNMF是最近的POI推荐方法[12]，它将贝叶斯矩阵分解与主题模型相结合。 在我们的数据中，没有可用的文本信息，因此我们仅使用此模型的分解部分进行比较。
* **GeoMF**：GeoMF是POI推荐[11]中最先进的方法.

对于时间感知的POI建议，使用以下方法作为基线。

* **UCF（+ G）：**这些是上面描述的UCF和UCF + G. 它们都没有利用时间信息，因此为所有时隙产生相同的建议。
* **UTF：**UTF是一种基于用户的时间协同过滤方法[5]，它通过使用时间衰减函数对签到进行加权来计算用户之间的相似性。
* **UCLAF**：UCLAF是一种基于PARAFAC的张量分解模型，最初建议使用用户位置活动张量数据来推荐位置和活动[27]，其中对分解施加Laplician正则化项以包含额外信息。我们将该模型应用于张量X，并分别用矩阵W和M构造了包含地理影响和时间影响的Laplian正则化项。
* **PITF：**这是一种基于等级的张量分解方法[17]，其中使用了贝叶斯个性化排序准则。
* **LRT：**这是最近开发的具有时间信息的POI推荐的矩阵分解方法[7]。 LRT通过将用户的潜在因素约束为在两个连续时隙中相似来结合时间影响。
* **UTE + SE**：如第2节所述，该方法利用时间感知POI推荐的地理影响和时间影响[24]。当仅考虑时间影响时，该模型由UTE表示，否则由UTE + SE表示。
* **BPP：**这是最新的时间感知POI建议[25]，它包含了地理和时间的影响。

在这些方法中，UTF、PITF和LRT没有利用地理影响。

**4.2 Experimental Results**

**4.2.1 Parameter Tuning**

在实验中，我们为所有数据集设置超参数ε= 0.3和C = 1.0。 对于学习率γ，我们在实验中设置小值0.0001以确保泛化精度。 对于其他参数，我们根据验证集调整它们以找到最佳值，然后在测试集中使用它们。

图2（a）显示了在两个数据集的两个设置下Rank-GeoFM的性能，因为我们改变了地理影响的参数α。 我们发现Rank-GeoFM对于两个数据的POI推荐在α= 0.2时表现最佳，并且在α= 0.1时对两个数据的时间感知POI推荐表现最佳。图2（b）显示了Rank-GeoFM的性能，因为我们改变了在地理影响矩阵构造中使用的参数k。 我们可以看到，对于所有情况，在k = 300时实现了最佳性能。图2（c）展示了参数β的影响，该参数β用于在时间感知POI推荐下的时间影响。 我们可以看到，在Foursquare和Gowalla上分别以β= 0.1和β= 0.2产生最佳性能。最后，我们展示了潜在因素的维度K对图2（d）中的性能的影响。 我们发现Rank-GeoFM的性能对尺寸K不敏感，我们在实验中使用K = 100。

**4.2.2 Results on POI Recommendation**

图3（a）-（d）显示了POI推荐的两个数据集上所有方法的性能。首先，我们可以看到基于存储器的方法UCF和UCF + G的性能比除了PMF之外的其他方法都差，后者都是基于因子分解的方法，并且UCF + G由于考虑了地理影响而改进了UCF。PMF表现最差，因为它是为明确的反馈数据而开发的，例如用户电影评级。它不适合POI推荐，其中签到是隐式反馈。这与[14]中的发现一致。

在其他基于因子分解的方法中，BPR-MF的性能非常有前景，尽管这种方法不能利用地理影响，并且在以前的工作中尚未用于POI推荐。该原因是BPR-MF作为基于排序的因子分解方法更适合处理隐式反馈数据。我们观察到GeoMF比GTBNMF表现更好。 这是因为GTBNMF通过在X中相等地拟合零和非零条目来进行分解，而GeoMF通过为非零条目分配更高权重来解决分解，这更适合于隐式数据。

我们可以看到，提出的Rank-GeoFM始终优于最先进的GeoMF方法。 就Pre @ 5而言，Foursquare和Gowalla数据集的改进分别超过41.6％和10％。 原因是GeoMF通过拟合具有不同权重的非零和零登记来解决数据稀疏性问题，这比我们的排名方法更不合理，因为零签到可能是缺失值并且不应该直接拟合。Rank-GeoFM / G表示我们的模型，不考虑地理影响。 可以看出，Rank-GeoFM / G的性能优于BPR-MF，这是一种基于排名的MF方法，不考虑地理影响。 这可能归因于我们的排名目标函数。 此外，由于地理影响的结合，我们观察到Rank-GeoFM分别在两个数据集上将Rank-GeoFM / G提高了30.3％和10.5％的Pre @ 5。

最后，我们注意到，我们的方法相对于基线的所有改进在配对t检验方面具有统计学意义，p值<0.01。

4.2.3 Results on Time-aware POI Recommendation

图4（a）-（d）显示了在未利用地理影响时，所有时间感知POI推荐方法的表现。 在所有基于内存的方法中，UCF比UTF和UTE表现更差，因为它不捕获时隙之间的相关性。虽然UTF和UTE都考虑时隙之间的相关性，但UTF仍然比UTE表现更差，因为UTE平滑了登记数据以处理数据稀疏性问题。在分解方法UCLAF，PITF和LRT中，LRT执行最差，因为它通过拟合张量X中的非零条目来获得分解，其中显着地受到稀疏性问题的影响。UCLAF的表现也不好，因为该方法适合X中的零和非零条目，这不是很合理。 PITF在三者中表现最佳，与其他方法相比，其性能也非常有前途。原因是PITF是一种基于排名的分解方法，可以缓解稀疏性问题。 此外，我们观察到基于图的方法BPP是最先进的方法，优于所有基于记忆和基于因子分解的基线方法。

Rank-GeoFM / G表示我们的模型具有时间影响但没有地理影响。 我们观察到我们的模型始终优于最先进的方法BPP，在Foursquare和Gowalla上分别以Pre @ 5表示比BPP好13.7％和16.2％。Rank-GeoFM / T / G表示我们的模型，不考虑时间和地理影响。 我们观察到Rank-GeoFM / T / G总是优于另一种基于排名的方法PITF。 此外，由于时间影响的结合，我们观察到Rank-GeoFM / G分别在两个数据上以Pre @ 5的方式将Rank-GeoFM / T / G提高了10.6％和13.1％。

图5（a）-（d）显示了进一步包含地理影响的方法的性能。 请注意，我们仅报告可利用地理影响的方法的结果。 结果表明，Rank-GeoFM在Foursquare和Gowalla上的表现分别优于其他方法的13.1％和15.3％，分别为Pre @ 5。

我们注意到，我们的方法相对于基线的所有改进在配对t检验方面具有统计学意义，p值<0.01。

**4.2.4 Comparison of Rank-GeoFM with and without Fast Learning Scheme**

我们比较了Rank-GeoFM的学习率，有和没有3.2.2节中介绍的快速学习方案。图6显示了结果，其中Foursquare数据作为比较的一个例子。没有快速学习方案的Rank-GeoFM意味着我们使用Eq（9）来计算梯度并在算法1中执行更新。我们观察到Rank-GeoFM（具有快速学习方案）在19小时内完成1000次迭代更新，而没有快速学习方案的对应物仅完成5次迭代。Rank-GeoFM在8小时内获得训练有素的模型，从而获得最佳性能，而对应者在经过19小时的训练后无法建立可接受的模型。该结果证明了使用快速学习方案的效率和必要性。此外，我们发现Rank-GeoFM的前500次迭代需要5.5小时，而接下来的500次迭代需要13.5小时。原因是在训练开始时，我们的模型没有对POI进行良好排名，因此抽取错误排名的POI所需的时间更少; 然而，随着培训过程的进行，排名变得更好，我们需要更多时间来抽样不正确的。 另一个有趣的观察结果是，我们发现Rank-GeoFM在大约100次迭代后精度有所下降。观察可能是因为我们的学习过程从粗略搜索转变为微调，即

E([| L |/n])（或梯度的大小）从大值减小到小值。

5. CONCLUSIONS

在本文中，我们提出了一种基于排序的因子分解方法Rank-GeoFM，用于POI推荐。 在所提出的模型中，我们通过拟合用户对POI的偏好排名来学习分解，这减轻了数据稀疏性问题。POI推荐和时间感知POI推荐的广泛实验结果表明，Rank-GeoFM显着优于最先进的方法。

Rank-GeoFM非常灵活地包含上下文信息。 将来，使用Rank-GeoFM调查其他上下文信息如何影响POI建议的性能将会很有趣。例如，如果我们知道每个POI的类别是cat（ℓ），那么方程（4）中的推荐得分可以修改如下以包含此信息：



其中ci是类别i的潜在因子，该等式中的最后一项表示类别cat（l）和用户u之间的相互作用得分。我们将把这作为我们未来的工作。