

第五章、上下文信息分析

讲师：武永亮



教学目标

- 理解地理位置定位的作用。
- 理解地点上下文信息

目录

1

时间上下文信息

2

地点上下文信息

利用上下文信息

- 之前提到的推荐系统算法主要集中研究了如何联系用户兴趣和物品，将最符合用户兴趣的物品推荐给用户，但这些算法都忽略了一点，就是用户所处的**上下文**（ context ）。这些上下文包括用户访问推荐系统的时间、地点、心情等，对于提高推荐系统的推荐系统是非常重要的。

利用上下文信息

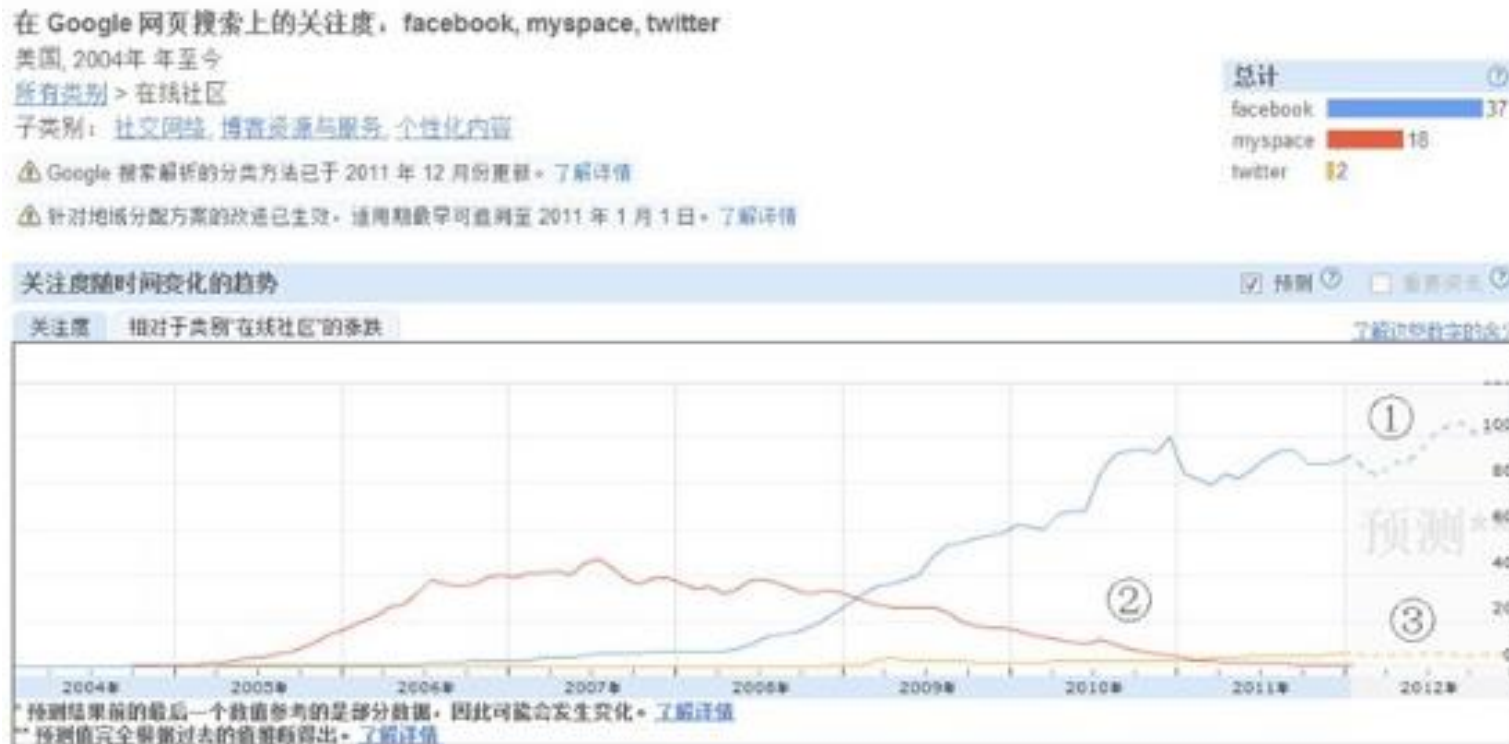


利用上下文信息

- 影响推荐算法的上下文很多种：
 - 时间上下文
 - 地点上下文
- 本章仍然研究TopN推荐，即如何给用户生成一个长度为 N 的推荐列表，而该列表包含了用户在某一时刻或者某个地方最可能喜欢的物品。

时间上下文信息

- 时间是一种重要的上下文信息，对用户兴趣有着深入而广泛的影响。时间信息对用户兴趣的影响表现在以下几个方面。
 - 用户兴趣是变化的
 - 物品也是有生命周期的
 - 季节效应



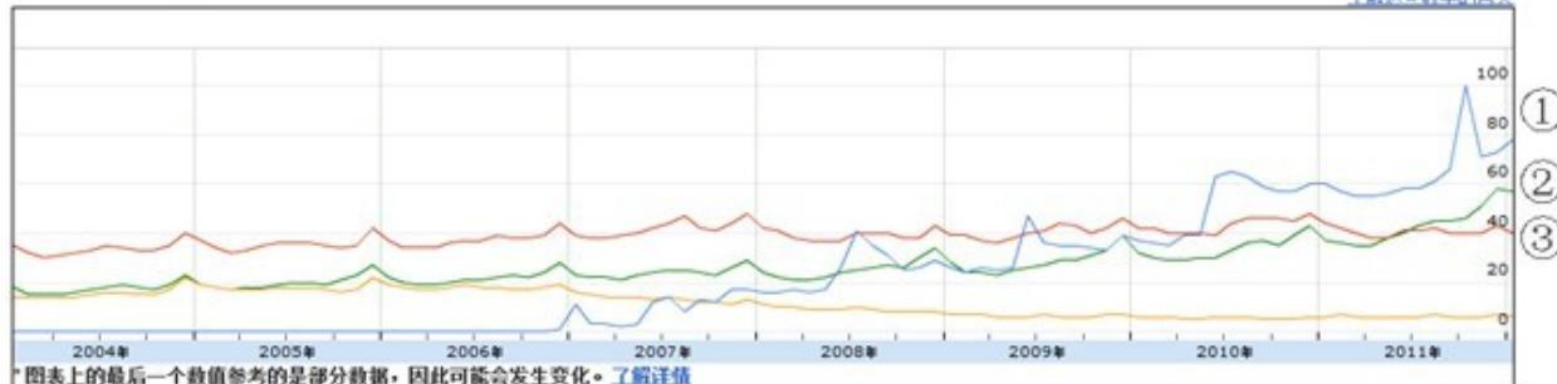
时间上下文信息

- 时间是一种重要的上下文信息，对用户兴趣有着深入而广泛的影响。时间信息对用户兴趣的影响表现在以下几个方面。
 - 用户兴趣是变化的
 - 物品也是有生命周期的
 - 季节效应

在 Google 网页搜索上的关注度：iphone, nokia, motorola, samsung
全球, 2004年 年至今
类别：互联网与电信, 计算机与电子产品, 购物, 艺术与娱乐, 汽车与车辆, 游戏
⚠ Google 搜索解析的分类方法已于 2011 年 12 月份更新。了解详情
⚠ 针对地域分配方案的改进已生效，适用期最早可追溯至 2011 年 1 月 1 日。了解详情

总计	
iphone	29
nokia	39
motorola	11
samsung	29

关注度随时间变化的趋势



时间上下文信息

- 时间是一种重要的上下文信息，对用户兴趣有着深入而广泛的影响。时间信息对用户兴趣的影响表现在以下几个方面。
 - 用户兴趣是变化的
 - 物品也是有生命周期的
 - 季节效应

在 Google 网页搜索上的关注度, ice cream, soup, chocolate, coffee
美国, 2004年 年至今
类别: [餐饮](#), [艺术与娱乐](#), [工商业](#), [家庭与园艺](#), [购物](#), [健康](#)
[Google 搜索解析的分类方法已于 2011 年 12 月份更新。了解详情](#)
[针对地域分配方案的改进已生效, 适用期最早可追溯至 2011 年 1 月 1 日。了解详情](#)

总计		
ice cream	20	
soup	39	
chocolate	55	
coffee	57	

关注度随时间变化的趋势



时间上下文信息

- 在给定时间信息后，推荐系统从一个静态系统变成了一个时变的系统，而用户行为数据也变成了时间序列。在给定数据集后，可以通过统计如下信息研究系统的时间特性。
 - 数据集每天独立用户数的增长情况
 - 系统的物品变化情况
 - 用户访问情况

时间上下文信息

- Delicious数据集包含950 000个用户在2003年9月到2007年12月间对网页打标签的行为。该数据集中包含132 000 000个标签和420 000 000条标签行为记录。该数据集每行是一条标签行为记录，由4部分组成——用户ID、日期、网页URL和标签，代表了一个用户在某一天对某个网页打上了某个标签的行为。

表5-1 离线实验数据集的基本统计信息

数 据 集	用 户 数	物 品 数	稀 疏 度
nytimes	4947	7856	99.65%
youtube	4551	7526	99.72%
wikipedia	7163	14770	99.86%
sourceforge	8547	5638	99.65%
blogspot	8703	10107	99.82%

时间上下文信息

- 不同类型网站的物品具有不同的生命周期，我们可以用如下指标度量网站中物品的生命周期。
 - 物品平均在线天数
 - 相隔 T 天系统物品流行度向量的平均相似度

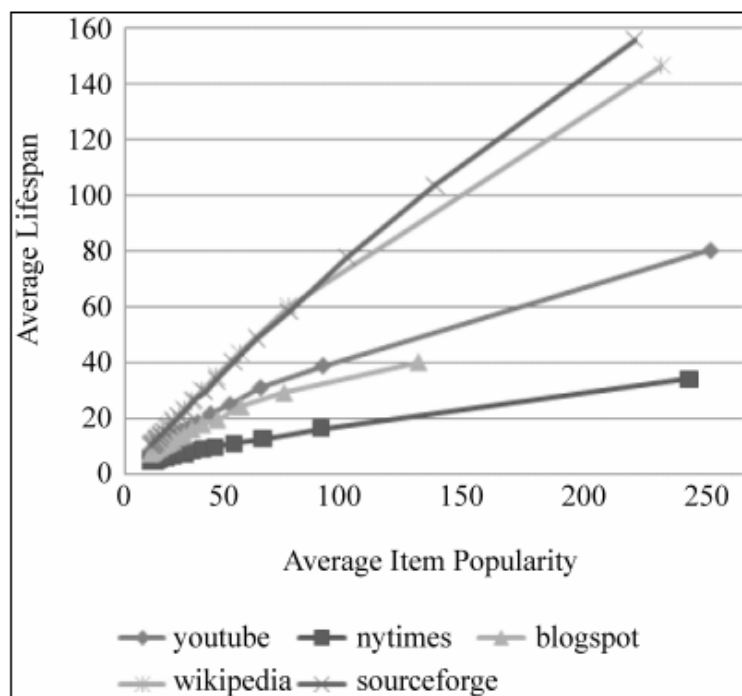


图5-5 不同数据集中物品流行度和物品平均在线时间的关系曲线

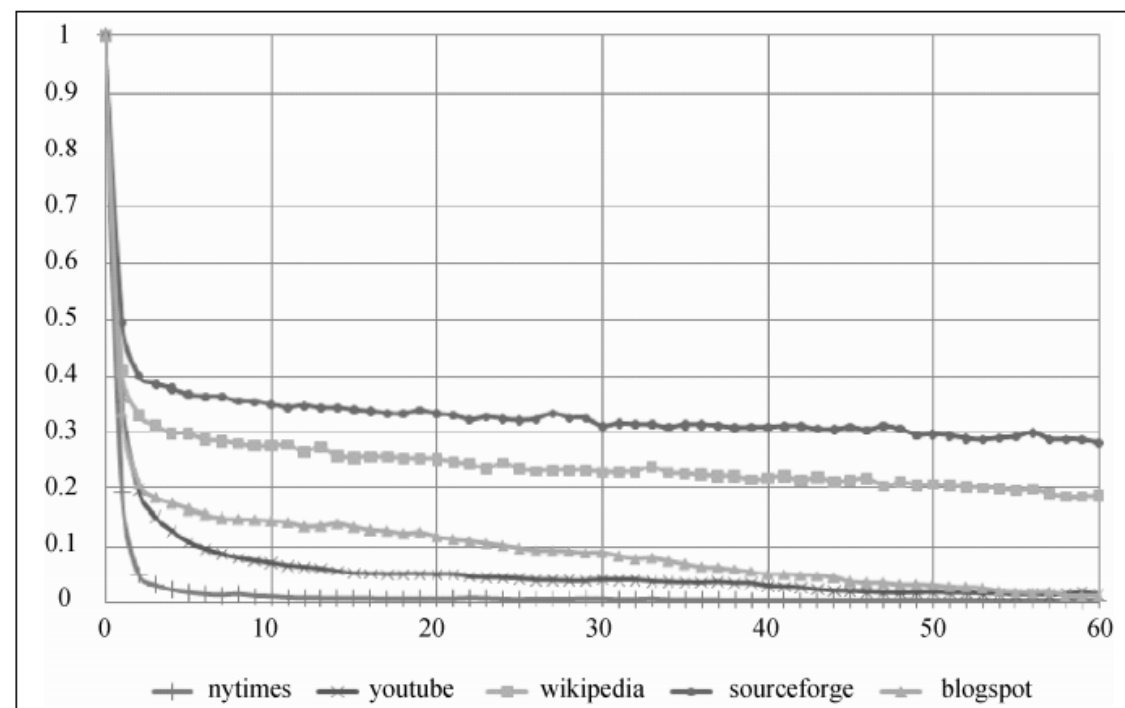


图5-6 相隔 T 天系统物品流行度向量的平均相似度

时间上下文信息


- 用户兴趣是不断变化的，其变化体现在用户不断增加的新行为中。一个实时的推荐系统需要能够实时响应用户新的行为，让推荐列表不断变化，从而满足用户不断变化的兴趣。
- 实现推荐系统的实时性除了对用户行为的存取有实时性要求，还要求推荐算法本身具有实时性，而推荐算法本身的实时性意味着：
 - 要求在每个用户访问推荐系统时，都根据用户这个时间点前的行为实时计算推荐列表。
 - 推荐算法需要平衡考虑用户的近期行为和长期行为，要保证推荐列表对用户兴趣预测的延续性。

时间上下文信息


Today's Recommendations For You

Here's a daily sample of items recommended for you. Click here to [see all recommendations](#).


Page 1 of 32




The Art of R Programming, 2nd Edition
by Norman Matloff
★★★★★ (14) \$24.29
[Fix this recommendation](#)




Introduction to Informatics
by Christopher D. Manning
★★★★★ (14) \$11.57
[Fix this recommendation](#)



Data Mining with Kettle and ETL
by Graham Williams
★★★★★ (13) \$12.25
[Fix this recommendation](#)



Data Mining: Practical Techniques
by Ian H. Witten
★★★★★ (17) \$38.48
[Fix this recommendation](#)



Mining the Social Web, 2nd Edition
by Matthew A. Russell
★★★★★ (13) \$29.39
[Fix this recommendation](#)

↓


MongoDB: The Definitive Guide (Paperback)
by Kristina Chodorow (Author), Michael Dirolf (Author)
★★★★★ (16 customer reviews) | [Like](#) (24)

↓


Today's Recommendations For You

Here's a daily sample of items recommended for you. Click here to [see all recommendations](#).


Page 1 of 32




The Art of R Programming, 2nd Edition
by Norman Matloff
★★★★★ (14) \$24.29
[Fix this recommendation](#)




Mining the Social Web, 2nd Edition
by Matthew A. Russell
★★★★★ (13) \$29.39
[Fix this recommendation](#)



Introduction to Informatics
by Christopher D. Manning
★★★★★ (14) \$11.57
[Fix this recommendation](#)




Data Mining with Kettle and ETL
by Graham Williams
★★★★★ (13) \$12.25
[Fix this recommendation](#)



Mahout in Action (Paperback)
by Sean Owen
★★★★★ (4) \$29.39
[Fix this recommendation](#)

Recommended for You



Mahout in Action
by Sean Owen (Author), et al.
Our Price: \$29.39
Used & new from \$25.00
[Add to Cart](#) [Add to Wish List](#)


Rate this item

★★★★★

☐ I own it

☐ Not interested

Because you liked...



MongoDB: The Definitive Guide
(Paperback)
by Kristina Chodorow (Author), Michael Dirolf (Author)
[Like](#) [Unlike](#)
☐ Don't use for recommendations

时间上下文信息

- 很多推荐系统的研究人员经常遇到一个问题，就是每天给用户的推荐结果都差不多，没有什么变化。推荐系统每天推荐结果的变化程度被定义为推荐系统的**时间多样性**。时间多样性高的推荐系统中用户会经常看到不同的推荐结果。
- 时间多样性高是否就能提高用户的满意度？
 - 给用户推荐最热门的10部电影。
 - 从最热门的100部电影中推荐10部给用户，但保证了时间多样性，每周都有7部电影推荐结果不在上周的推荐列表中。
 - 每次都从所有电影中随机挑选10部推荐给用户。

时间上下文信息

- 然后，研究人员进行了用户调查实验，发现了如下现象。
 - A、B算法的平均分明显高于C算法。这说明纯粹的随机推荐虽然具有最高的时间多样性，但不能保证推荐的精度。
 - A算法的平均分随时间逐渐下降，而B算法的平均分随时间基本保持不变。这说明A算法因为没有时间多样性，从而造成用户满意度不断下降，从而也说明了保证时间多样性的重要性。

时间上下文信息

- 提高推荐结果的时间多样性需要分两步解决：
 - 首先，需要保证推荐系统能够在用户有了新的行为后及时调整推荐结果，使推荐结果满足用户最近的兴趣；
 - 其次，需要保证推荐系统在用户没有新的行为时也能够经常变化一下结果，具有一定的时间多样性。

时间上下文信息

- 如果用户没有行为，如何保证给用户的推荐结果具有一定的时间多样性呢？一般的思路有以下几种。
 - 在生成推荐结果时加入一定的随机性。
 - 记录用户每天看到的推荐结果，然后在每天给用户进行推荐时，对他前几天看到过很多次的推荐结果进行适当地降权。
 - 每天给用户使用不同的推荐算法。可以设计很多推荐算法，然后在每天用户访问推荐系统时随机挑选一种算法给他进行推荐。
- 当然，时间多样性也不是绝对的。推荐系统需要首先保证推荐的精度，在此基础上适当地考虑时间多样性。在实际应用中需要通过多次的实验才能知道什么程度的时间多样性对系统是最好的。

时间上下文信息

- 在没有时间信息的数据集中，我们可以给用户推荐历史上最热门的物品。那么在获得用户行为的时间信息后，最简单的非个性化推荐算法就是给用户推荐最近最热门的物品了。

给定时间 T ，物品 i 最近的流行度 $n_i(T)$ 可以定义为：

$$n_i(T) = \sum_{(u,i,t) \in \text{Train}, t < T} \frac{1}{1 + \alpha(T - t)}$$

```
def RecentPopularity(records, alpha, T):  
    ret = dict()  
    for user,item,tm in records:  
        if tm >= T:  
            continue  
        addToDict(ret, item, 1 / (1.0 + alpha * (T - tm)))  
    return ret
```

时间上下文信息

- 时间上下文相关的ItemCF算法：基于物品（ item-based ）的个性化推荐算法是商用推荐系统中应用最广泛的，该算法由两个核心部分构成：
 - 利用用户行为离线计算物品之间的相似度；
 - 根据用户的历史行为和物品相似度矩阵，给用户做在线个性化推荐。
- 时间信息在上面两个核心部分中都有重要的应用：
 - 物品相似度
 - 在线推荐

时间上下文信息

- 回顾一下基于物品的协同过滤算法，它通过如下公式计算物品的相似度：

$$\text{sim}(i, j) = \frac{\sum_{u \in N(i) \cap N(j)} 1}{\sqrt{|N(i)| |N(j)|}}$$

- 在给用户 u 做推荐时，用户 u 对物品 i 的兴趣 $p(u, i)$ 通过如下公式计算：

$$p(u, i) = \sum_{j \in N(u)} \text{sim}(i, j)$$

- 在得到时间信息（用户对物品产生行为的时间）后，我们可以通过如下公式改进相似度计算：

$$\text{sim}(i, j) = \frac{\sum_{u \in N(i) \cap N(j)} f(|t_{ui} - t_{uj}|)}{\sqrt{|N(i)| |N(j)|}}$$

时间上下文信息

- UserCF算法同样可以利用时间信息提高预测的准确率。基本思想：给用户推荐和他兴趣相似的其他用户喜欢的物品。从这个基本思想出发，我们可以在以下两个方面利用时间信息改进UserCF算法。
 - **用户兴趣相似度** 两个用户兴趣相似是因为他们喜欢相同的物品，或者对相同的物品产生过行为。但是，如果两个用户同时喜欢相同的物品，那么这两个用户应该有更大的兴趣相似度。
 - **相似兴趣用户的最近行为** 在找到和当前用户 u 兴趣相似的一组用户后，这组用户最近的兴趣显然相比这组用户很久之前的兴趣更加接近用户 u 今天的兴趣。

时间上下文信息

- 假设我们今天要给一个NBA篮球迷推荐新闻。首先，我们需要找到一批和他一样的NBA迷，然后找到这批人在当前时刻最近阅读最多的新闻推荐给当前用户，而不是把这批人去年阅读的新闻推荐给当前用户，因为他们去年阅读最多的新闻在现在看显然过期了。
- UserCF通过如下公式计算用户u和用户v的兴趣相似度：

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)| \cup |N(v)|}}$$

- 其中 $N(u)$ 是用户u喜欢的物品集合， $N(v)$ 是用户v喜欢的物品集合。可以利用如下方式考虑时间信息：

$$w_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \frac{1}{1 + \alpha |t_{ui} - t_{vi}|}}{\sqrt{|N(u)| \cup |N(v)|}}$$

时间上下文信息

- 在得到用户相似度后，UserCF通过如下公式预测用户对物品的兴趣：

$$p(u,i) = \sum_{v \in S(u,K)} w_{uv} r_{vi}$$

- 如果考虑和用户u兴趣相似用户的最近兴趣，我们可以设计如下公式：

$$p(u,i) = \sum_{v \in S(u,K)} w_{uv} r_{vi} \frac{1}{1 + \alpha(t_0 - t_{vi})}$$

时间上下文信息

时间段图模型 $G(U, S_U, I, S_I, E, w, \sigma)$ 也是一个二分图。 U 是用户节点集合， S_U 是用户时间段节点集合。一个用户时间段节点 $v_{ut} \in S_U$ 会和用户 u 在时刻 t 喜欢的物品通过边相连。 I 是物品节点集合， S_I 是物品时间段节点集合。一个物品时间段节点 $v_{it} \in S_I$ 会和所有在时刻 t 喜欢物品 i 的用户通过边相连。 E 是边集合，它包含了3种边：(1)如果用户 u 对物品 i 有行为，那么存在边 $e(v_u, v_i) \in E$ ；(2)如果用户 u 在 t 时刻对物品 i 有行为，那么就存在两条边 $e(v_{ut}, v_i), e(v_u, v_{it}) \in E$ 。 $w(e)$ 定义了边的权重， $\sigma(e)$ 定义了顶点的权重。

时间上下文信息

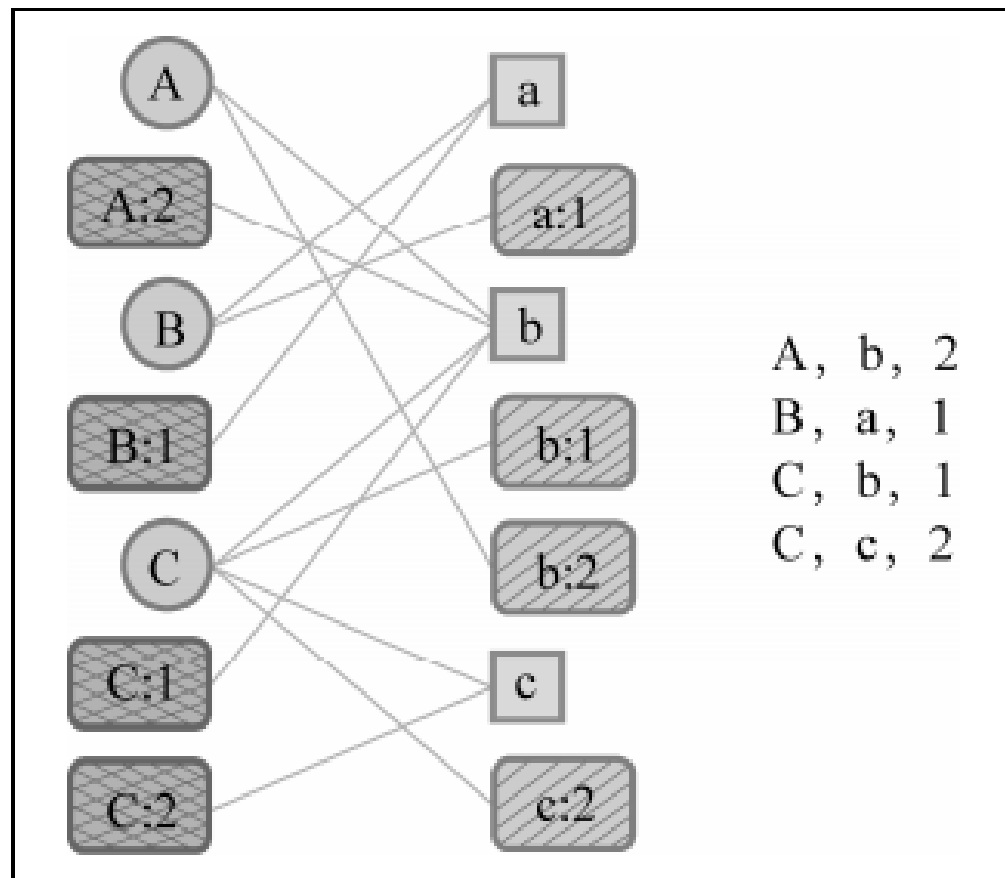


图5-8 时间段图模型示例

时间上下文信息

- 我们提出了一种称为路径融合算法的方法，通过该算法来度量图上两个顶点的相关性。两个相关性比较高的顶点一般具有如下特征：
 - 两个顶点之间有很多路径相连；
 - 两个顶点之间的路径比较短；
 - 两个顶点之间的路径不经过出度比较大的顶点。

时间上下文信息

假设 $P = \{v_1, v_2, \dots, v_n\}$ 是连接顶点 v_1 和 v_n 的一条路径，这条路径的权重 $\Gamma(P)$ 取决于这条路径经过的所有顶点和边：

$$\Gamma(P) = \sigma(v_n) \prod_{i=1}^{n-1} \frac{\sigma(v_i) \cdot w(v_i, v_{i+1})}{|\text{out}(v_i)|^\rho}$$

这里 $\text{out}(v)$ 是顶点 v 指向的顶点集合， $|\text{out}(v)|$ 是顶点 v 的出度， $\sigma(v_i) \in (0, 1]$ 定义了顶点的权重， $w(v_i, v_{i+1}) \in (0, 1]$ 定义了边 $e(v_i, v_{i+1})$ 的权重。上面的定义符合上面3条原则的后两条。首先，因为 $\frac{\sigma(v_i) \cdot w(v_i, v_{i+1})}{|\text{out}(v_i)|^\rho} \in (0, 1)$ ，所以路径越长 n 越大， $\Gamma(P)$ 就越小。同时，如果路径经过了出度大的顶点 v' ，那么因为 $|\text{out}(v')|$ 比较大，所以 $\Gamma(P)$ 也会比较小。

在定义了一条路径的权重后，就可以定义顶点之间的相关度。对于顶点 v 和 v' ，令 $p(v, v', K)$ 为这两个顶点间距离小于 K 的所有路径，那么这两个顶点之间的相关度可以定义为：

$$d(v, v') = \sum_{P \in p(v, v', K)} \Gamma(P)$$

对于时间段图模型，所有边的权重都定义为1，而顶点的权重 $\sigma(v)$ 定义如下：

$$\sigma(v) = \begin{cases} 1 - \alpha & (v \in U) \\ \alpha & (v \in S_U) \\ 1 - \beta & (v \in I) \\ \beta & (v \in S_I) \end{cases}$$

这里， $\alpha, \beta \in [0, 1]$ 是两个参数，控制了不同顶点的权重。

时间上下文信息

- 对每一个用户，将物品按照该用户对物品的行为时间从早到晚排序，然后将用户最后一个产生行为的物品作为测试集，并将这之前的用户对物品的行为记录作为训练集。推荐算法将根据训练集学习用户兴趣模型，给每个用户推荐 N 个物品，并且利用准确率和召回率评测推荐算法的精度。本节将选取不同的 $N(10, 20, \dots, 100)$ 进行10次实验，并画出最终的准确率和召回率曲线，通过该曲线来比较不同算法的性能。

$$\text{Recall}@N = \frac{\sum_u |R(u, N) \cap T(u)|}{\sum_u |T(u)|}$$

$$\text{Precision}@N = \frac{\sum_u |R(u, N) \cap T(u)|}{\sum_u |R(u, N)|}$$

时间上下文信息

- 本节的离线实验将同时对比如下算法，将它们的召回率和准确率曲线画在一张图上。

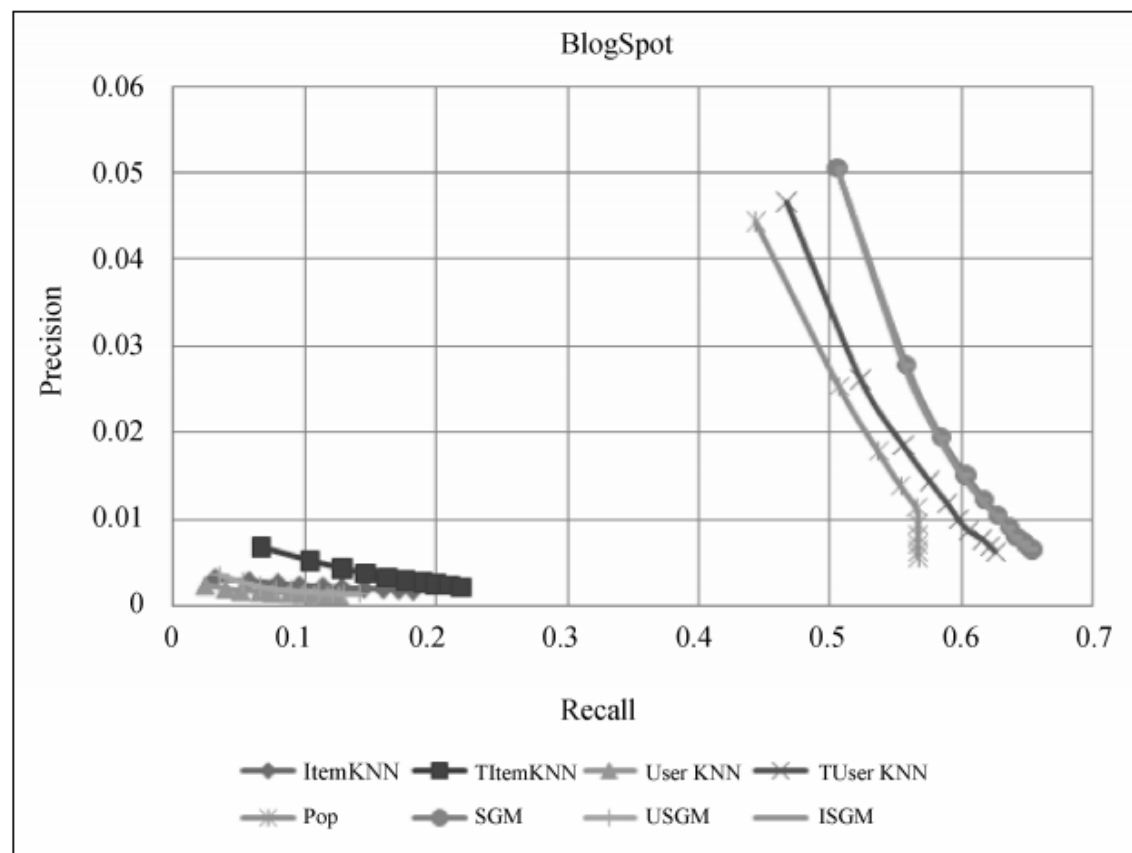


图5-9 BlogSpot数据集的召回率和准确率曲线

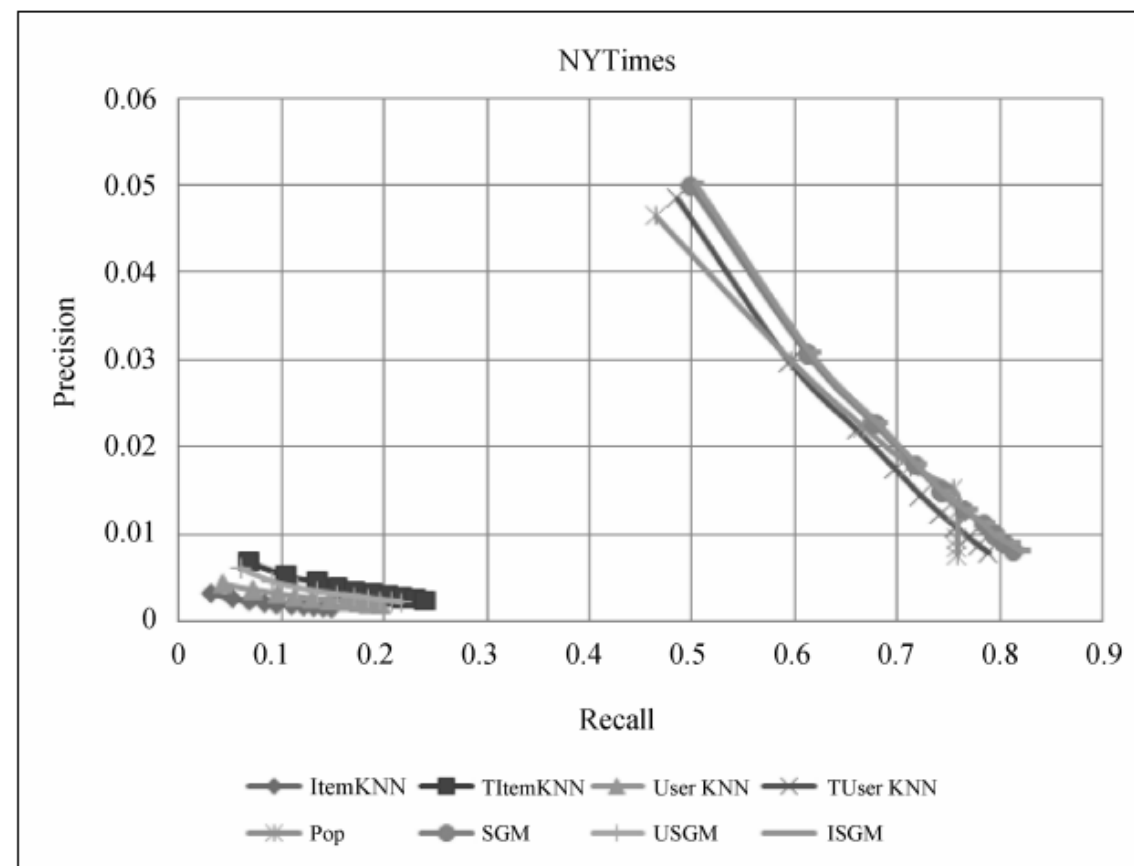


图5-10 NYTimes数据集的召回率和准确率曲线

时间上下文信息

- 本节的离线实验将同时对比如下算法，将它们的召回率和准确率曲线画在一张图上。

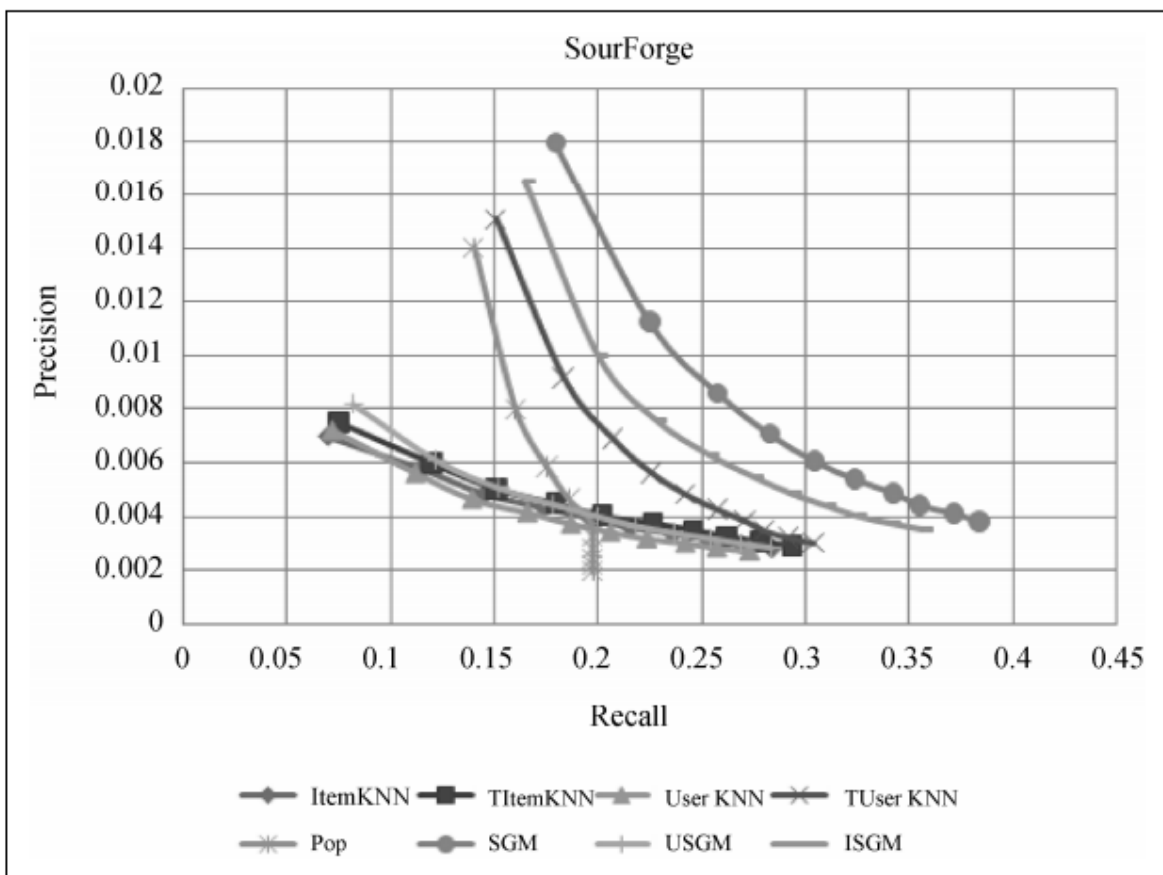


图5-11 SourceForge数据集的召回率和准确率曲线

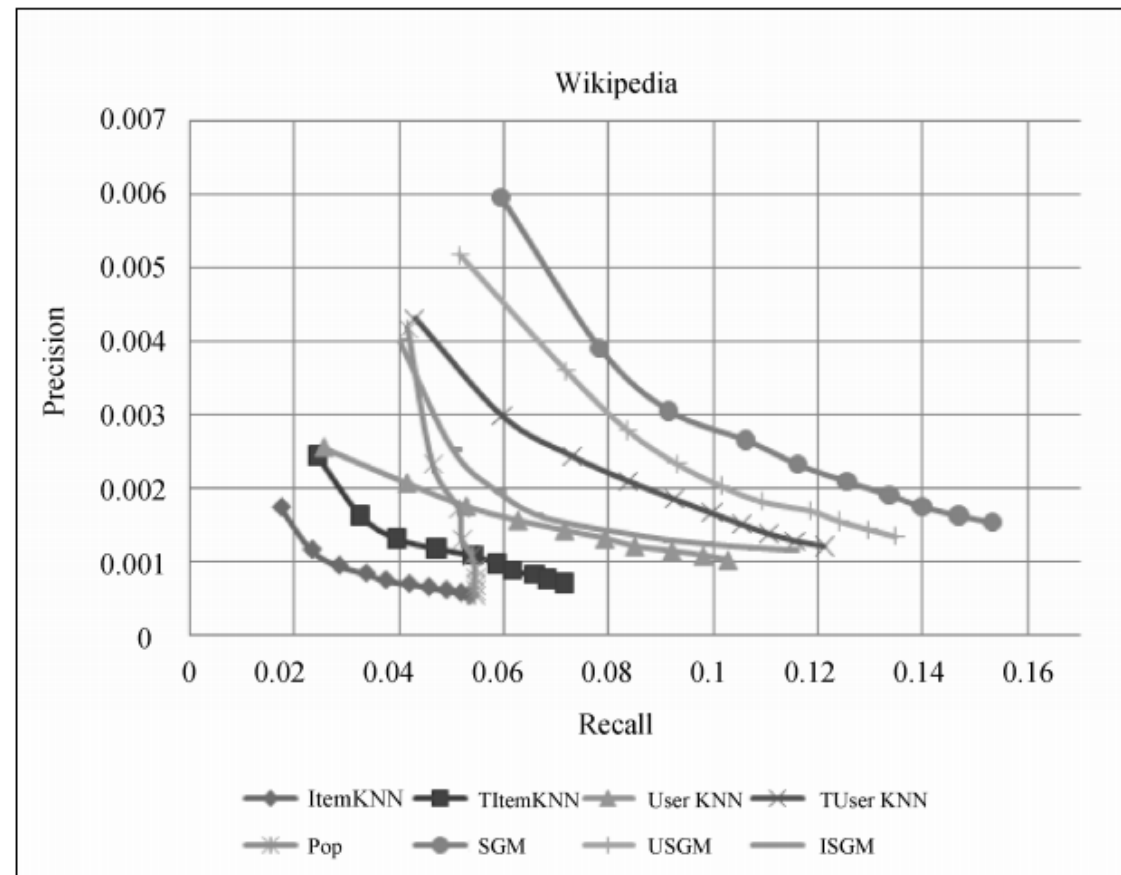


图5-12 Wikipedia数据集的召回率和准确率曲线

时间上下文信息

- 这些曲线的形状将数据集分成了两类。一类是BlogSpot、YouTube、NYTimes，另一类是Wikipedia和SourceForge。这主要是因为第一类数据集的时效性很强，因此用户兴趣的个性化不是特别明显，每天最热门的物品已经吸引了绝大多数用户的眼球，而长尾中的物品很少得到用户的关注。

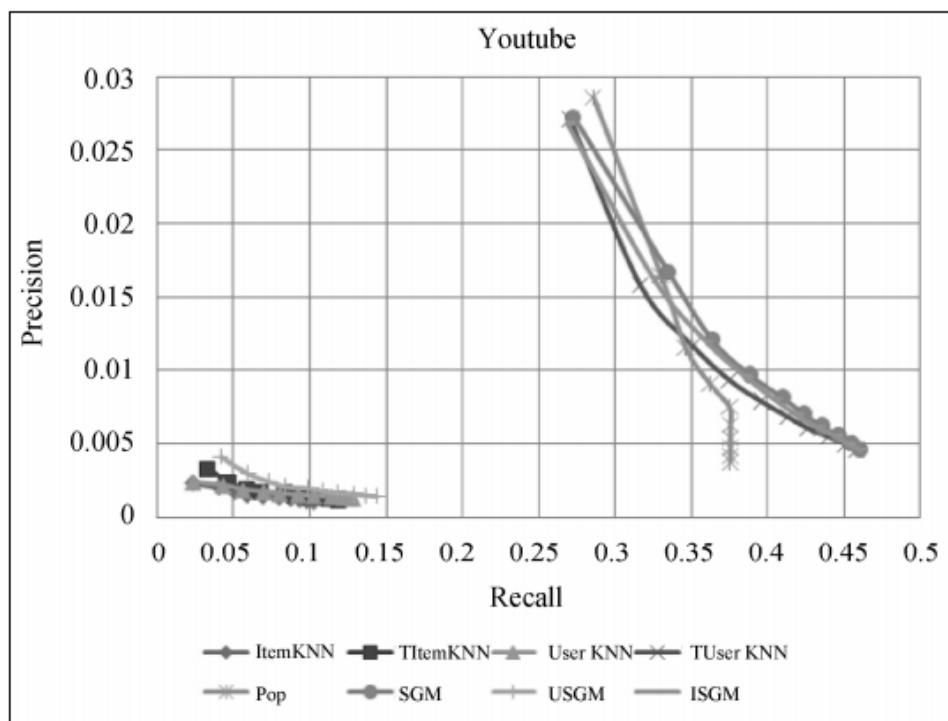


图5-13 YouTube数据集的召回率和准确率曲线

目录

1 时间上下文信息

2 地点上下文信息

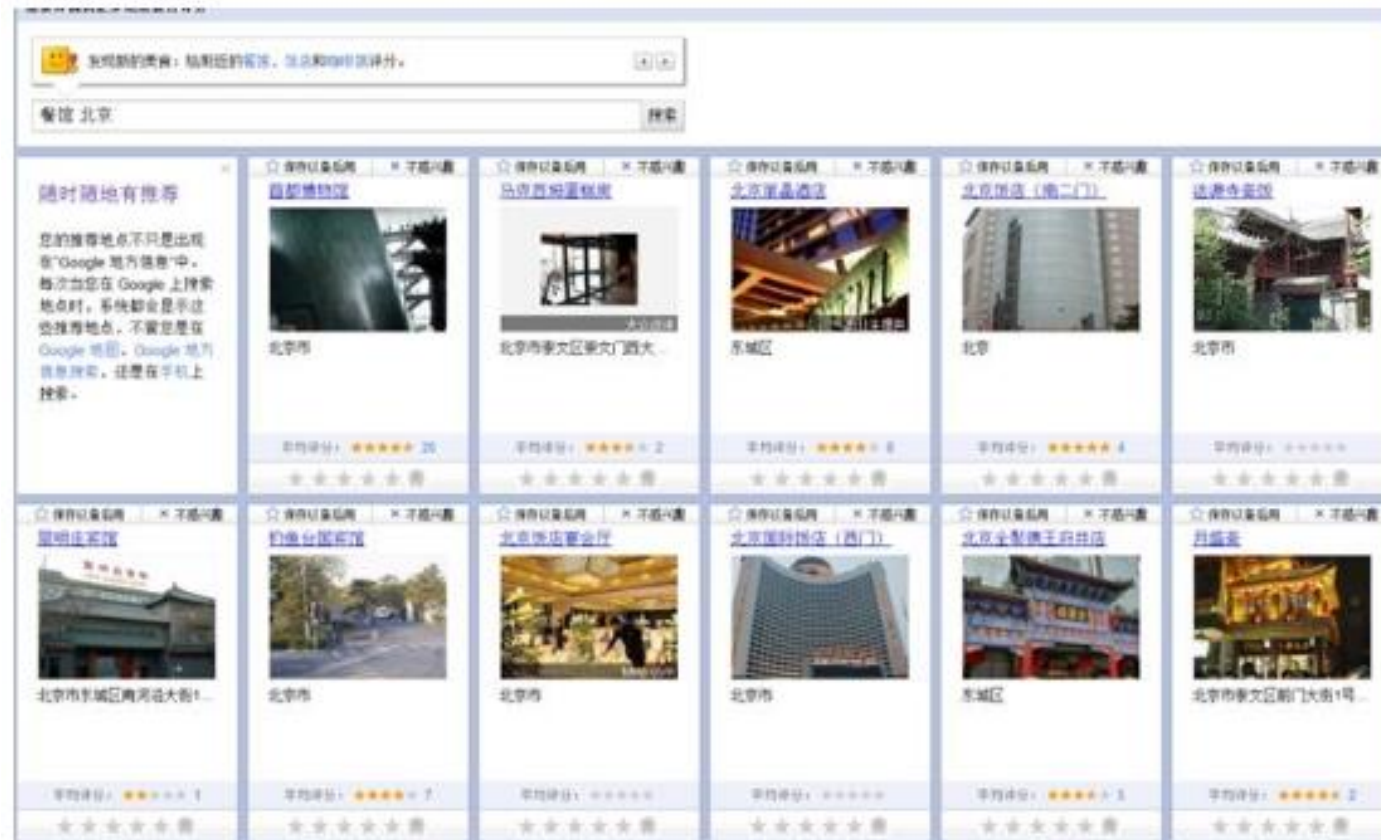
地点上下文信息

- 地点作为一种重要的空间特征，也是一种重要的上下文信息。不同地区的用户兴趣有所不同，用户到了不同的地方，兴趣也会有所不同。很多基于位置的服务（LBS）软件都提供了推荐附近餐馆和商店的功能。



地点上下文信息

- 地点作为一种重要的空间特征，也是一种重要的上下文信息。不同地区的用户兴趣有所不同，用户到了不同的地方，兴趣也会有所不同。很多基于位置的服务（LBS）软件都提供了推荐附近餐馆和商店的功能。



地点上下文信息

- 研究人员提出过一个称为LARS（Location Aware Recommender System, 位置感知推荐系统）的和用户地点相关的推荐系统。
 - 该系统首先将物品分成两类，一类是有空间属性的，另一类是无空间属性的物品。
 - 同时，它将用户也分成两类，一类是有空间属性的，另一类用户并没有相关的空间属性信息。
- 它使用的数据集有3种不同的形式。
 - （用户，用户位置，物品，评分）
 - （用户，物品，物品位置，评分）
 - （用户，用户位置，物品，物品位置，评分）

地点上下文信息

- LARS通过研究前两种数据集，发现了用户兴趣和地点相关的两种特征。
 - 兴趣本地化：不同地方的用户兴趣存在着很大的差别。
 - 活动本地化：一个用户往往在附近的地区活动。

表5-2 美国、英国、德国用户兴趣度最高的歌手

德 国	美 国	英 国
Die Ärzte (德国柏林著名的摇滚乐队)	girl talk (美国音乐人)	Biffy Clyro (苏格兰摇滚乐队)
Clueso (德国歌手)	They Might Be Giants (成立于1982年的美国替代摇滚乐队)	Feeder (威尔士的替代摇滚乐队)
Peter Fox (德国柏林的Hip hop音乐人)	Guster (美国波士顿的替代摇滚乐队)	Idlewild (苏格兰摇滚乐队)
Deichkind (德国汉堡的乐队)	Saves the Day (美国普林斯顿的摇滚乐队)	Elbow (英格兰摇滚乐队)
K.I.Z. (德国柏林的hip hop乐队)	Spoon (美国奥斯汀的摇滚乐队)	Girls Aloud (英格兰和爱尔兰的流行女子乐团)

地点上下文信息

- 对于第一种数据集，LARS的基本思想是将数据集根据用户的位置划分成很多子集。因此，数据集也会划分成一个树状结构。然后，给定每一个用户的位置，我们可以将他分配到某一个叶子节点中，而该叶子节点包含了所有和他同一个位置的用户的行为数据集。然后，LARS就利用这个叶子节点上的用户行为数据，通过ItemCF给用户进行推荐。
- 不过这样做的缺点是，每个叶子节点上的用户数量可能很少，因此他们的行为数据可能过于稀疏，从而无法训练出一个好的推荐算法。为此，我们可以从根节点出发，在到叶子节点的过程中，利用每个中间节点上的数据训练出一个推荐模型，然后给用户生成推荐列表。

地点上下文信息

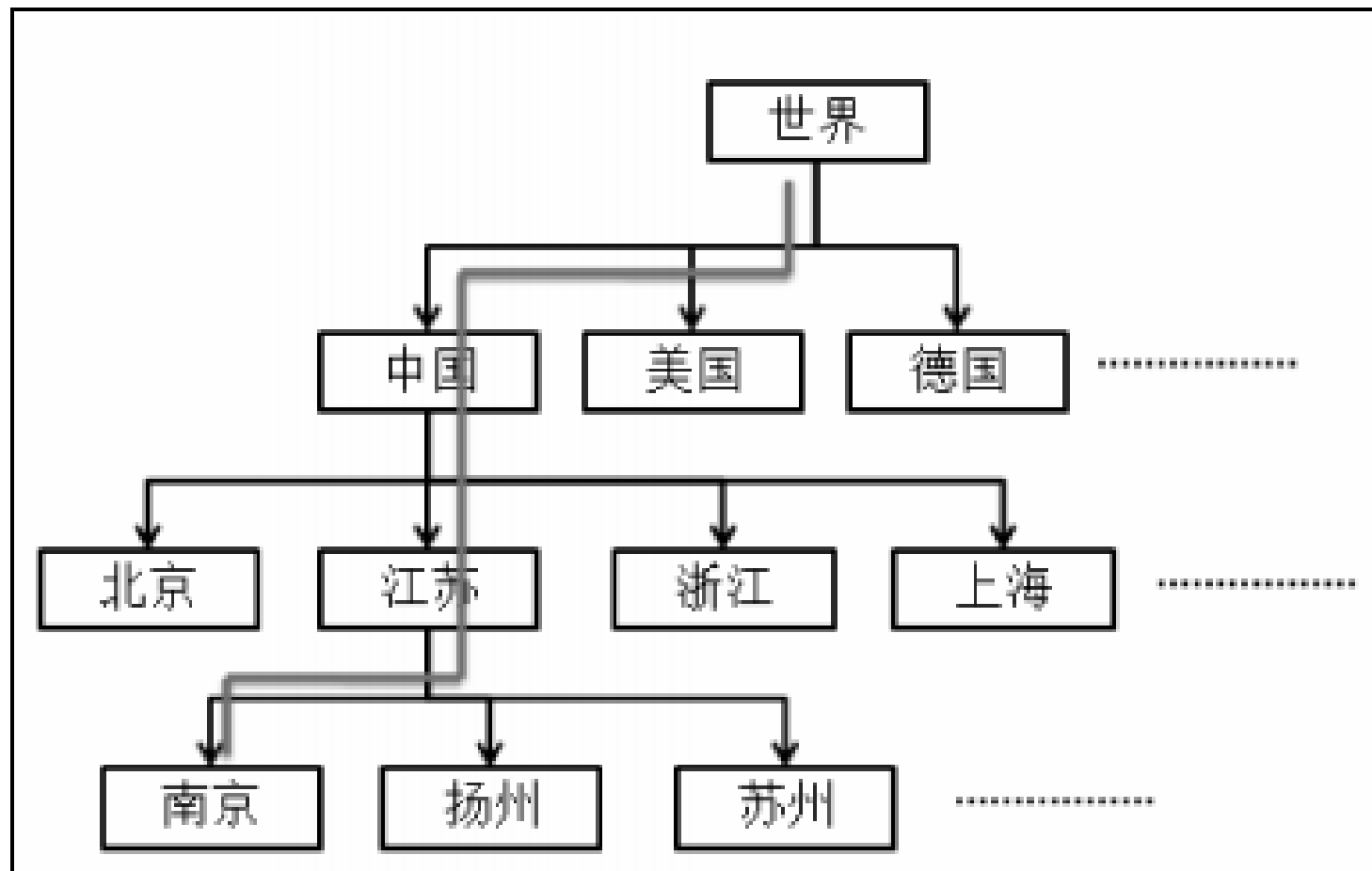


图5-16 一个简单的利用用户位置信息进行推荐的例子

地点上下文信息

- 对于第二种数据集，每条用户行为表示为四元组（用户、物品、物品位置、评分），表示了用户对某个位置的物品给了某种评分。对于这种数据集，LARS会首先忽略物品的位置信息，利用ItemCF算法计算用户 u 对物品 i 的兴趣 $P(u, i)$ ，但最终物品 i 在用户 u 的推荐列表中的权重定义为：

$$\text{RecScore}(u, i) = P(u, i) - \text{TravelPenalty}(u, i)$$

地点上下文信息

- 对于第三种数据集，LARS一文没有做深入讨论。不过，从第三种数据集的定义可以看到，它相对于第二种数据集增加了用户当前位置这一信息。而在给定了这一信息后，我们应该保证推荐的物品应该距离用户当前位置比较近，在此基础上再通过用户的历史行为给用户推荐离他近且他会感兴趣的物品。

内容回顾

1

时间上下文信息

2

地点上下文信息

Thank you !