

基于混合推荐和隐马尔科夫模型的服务推荐方法

马建威^{1,2}, 陈洪辉¹, STEPHAN Reiff-Marganiec³

- (1. 国防科学技术大学 信息系统工程重点实验室, 湖南 长沙, 410073;
2. 第三军医大学卫勤教研室, 重庆, 400038;
3. 莱斯特大学 计算机科学与技术系, 英国 莱斯特)

摘要: 针对现阶段越来越多的服务开始部署于云环境, 服务数量呈几何级增长, 必须获取并推荐最优服务, 而传统的基于内容的过滤或协同过滤方法缺乏对新用户和冗余服务的有效处理方法, 提出一种在云环境下对最优服务进行有效推荐的方法。首先, 分析 2 种协同过滤方法的优缺点, 并提出改进的混合推荐算法; 其次, 针对常常被忽略的新用户学习策略, 提出新用户偏好的确定方法; 针对服务的动态变化情况, 基于隐马尔科夫模型(hidden Markov model)提出一种冗余服务消解策略。最后, 基于真实数据集和通过公开 API 获取的公共服务集进行实验。研究结果表明: 所提出的算法与其他方法相比具有更高的准确度和更好的服务质量, 能更有效地提高系统性能。

关键词: 协同过滤; 服务选择; 新用户学习; 隐马尔科夫模型; 冗余检测

中图分类号: TP393

文献标志码: A

文章编号: 1672-7207(2016)01-0082-09

Recommending services via hybrid recommendation algorithms and hidden Markov model in cloud

MA Jianwei^{1,2}, CHEN Honghui¹, STEPHAN Reiff-Marganiec³

- (1. Science and Technology of Information System Engineering Laboratory, National University of Defense Technology, Changsha 410073, China;
2. Department of Health Service, Third Military Medical University, Chongqing 400038, China;
3. Department of Computer Science, University of Leicester, Leicester, UK)

Abstract: With the increase of the number of users using web services for online activities through thousands of services, proper services must be obtained; however, the existing methods such as content-based approaches or collaborative filtering do not consider new users and redundant services. An effective approach was proposed to recommend the most appropriate services in a cloud computing environment. Firstly, a hybrid collaborative filtering method was proposed to recommend services. The method greatly enhances the prediction of the current QoS value which may differ from that of the service publication phase. Secondly, a strategy was proposed to obtain the preferences of the new users who are neglected in other research. Thirdly, a HMM (hidden Markov model)-based approach was proposed to identify redundant services in a dynamic situation. Finally, several experiments were set up based on real data set and publicly published web services data set. The results show that the proposed algorithm has better performance than other methods.

Key words: collaborative filtering; service selection; new user learning; hidden Markov model; redundancy detection

收稿日期: 2015-01-22; 修回日期: 2015-03-25

基金项目(Foundation item): 国家自然科学基金资助项目(71071160); 全军后勤科研重点计划项目(BWS14J032); 湖南省优秀研究生创新项目(CX2011B024); 国防科技大学优秀研究生创新项目(B110502); 第三军医大学人文社科基金资助项目(2015XRW10) (Projects(61070216, 71071160) supported by the National Natural Science Foundation of China; Project(BWS14J032) supported by the Logistics Research Plan of Chinese PLA; Project(CX2011B024) supported by the Outstanding Graduate Student Innovation Fund of Hunan Province; Project(B110502) supported by the Outstanding Graduate Student Innovation Fund of National University of Defense Technology; Project(2015XRW10) supported by Social Science Foundation of Third Military Medical University)

通信作者: 马建威, 博士, 讲师, 从事社会网络分析、服务计算技术及卫勤信息化技术研究; E-mail: majianwei@nudt.edu.cn

近年来,大量的分布式系统部署环境不断向云环境迁移,越来越多的服务开始以云服务的形式展现,服务的用户也呈现出几何量级式快速增长,并且这种趋势仍在加剧。然而,随着大量云服务的出现,诸如服务提供者的虚假恶意行为、服务使用高峰时所出现的瓶颈以及云环境的动态性等都会造成信息质量发生动态变化,甚至对用户变得不可知,当前摆在面向服务应用面前的最大挑战已经转变为如何最大化地利用服务资源,并且充分保证用户的满意度。对服务质量进行预测并推荐适当的服务成为解决问题的关键。推荐系统在解决以上问题方面正发挥着越来越重要的作用^[1]。协同过滤方法(collaborative filtering)是推荐系统的一种信息推荐方法,在学习用户偏好以及预测用户兴趣等方面得到广泛应用,并且取得了非常好的应用效果。然而,传统的协同过滤方法通常通过记录用户交易历史的方法计算用户之间的相似性,从而进行过滤推荐,却无法有效地对新用户进行推荐,这就出现所谓的“冷启动”(cold start)问题。冷启动问题在云环境条件下更加容易发生,原因在于每天都有数以百万级的新用户在开始使用互联网寻找合适的服务资源。此外,由于拷贝与转载操作简单,因此,互联网上经常会出现大量的冗余服务资源,而传统的协同过滤方法无法有效区分冗余资源,这也使得用户常常受限于大量冗余的资源列表,而无法从中找出合适的资源,进而极大地降低了用户的使用满意度。为此,本文作者提出一种基于混合推荐算法和隐马尔科夫模型的信息推荐方法。首先提出一种混合协同过滤方法,利用随机行走(random walk)策略进行云环境下的信息推荐;然后,基于受欢迎度和信息熵的方法提出新用户兴趣判定的方法;最后,基于隐马尔科夫模型(hidden Markov model)提出服务冗余的判定方法,并进行实验验证。

1 问题描述

由于网络服务数据急剧上升,海量的资源使得用户无法去尝试使用每一个服务,因此,对服务质量进行预测并推荐服务资源成为了一个急需解决的关键问题。近年来,人们对服务选择和服务推荐的研究较多,如自适应资源管理^[2]、历史信息分析^[3]以及各种统计方法^[4]已在服务选择领域得到广泛应用并取得了较好的研究成果。用户一般服务矩阵见表1。

这里所提出的服务质量预测和推荐方法主要是基于对服务使用者与服务提供者和服务推荐者之间进行

表1 用户-服务矩阵

Table 1 User-service matrix

User	Service 1	Service 2	...	Service m
User 1	$r_{1,1}$	$r_{1,2}$...	$r_{1,m}$
User 2	$r_{2,1}$	$r_{2,2}$...	$r_{2,m}$
\vdots	\vdots	\vdots	\vdots	\vdots
User n	$r_{n,1}$	$r_{n,2}$...	*
Active user u	$r_{u,1}$	*	*	$r_{u,m}$

注:“*”表示用户尚未对该服务进行评价。

相似度计算的思想提出的。原因在于:在 QoS 信息相对稳定的前提下,服务的历史数据统计能够对服务的选择和推荐产生巨大而有意义的影响。在当前研究中,有很多关于用户相似度计算的技术如基于内容的过滤^[5]和协同过滤^[6-8]。但对云环境下的超大规模用户数量而言,基于内容的过滤方法将面临巨大的计算复杂度问题,无法有效地建立用户配置文件,因此,基于协同过滤的思想是该种情况下服务推荐的有效手段。协同过滤主要有2类:基于用户的协同过滤(user-based collaborative filtering)和基于项目的协同过滤(item-based collaborative filtering)。

1) 基于用户的协同过滤(user-based collaborative filtering)的基本思想是计算用户与用户之间的相似度 $S_{im}(a,u)$,典型的方法包括最近 K 邻居算法(K-nearest neighbor algorithm)以及皮尔森相关系数算法(Pearson correlation coefficient algorithm)^[9]。

$$S_{im}(a,u) = \frac{\sum_{i \in I} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2}} \quad (1)$$

其中: $r_{a,i}$ 表示用户 a 对服务 i 的评价,如表1所示;而所有用户对服务 i 的平均评价用 \bar{r}_a 表示, $\bar{r}_a = \sum_{i \in I} r_{a,i} / |I_i|$; \bar{r}_u 表示用户 u 对所有服务的平均评价。

该种方法的优点在于易于实现且结果容易理解。但缺点也显而易见,其中最重要的是其性能会随着数据规模的增大而急剧下降,即该方法的扩展性较差,因此,不适宜应用于云环境背景下。为解决以上问题,DAS等^[10]提出了一些解决方法,但这些方法在面对新用户和新服务时显得效果不明显(新用户和新服务没有任何历史信息)。

2) 基于项目的协同过滤(item-based collaborative filtering)与基于用户的方法不同,其主要集中于服务集,通过计算服务之间的相似度来完成预测和推荐。由于服务之间的关系相对来说较固定,因此,该方法可以进行预计算(pre-computing),从而可以有效地缓

解大规模数据条件下的计算复杂度问题^[11],但仍缺乏满足新用户需求的有效手段。任意 2 个服务 i 与服务 j 之间的相似度($S_{im}(i,j)$)可以通过下式进行计算:

$$S_{im}(i,j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,i} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_j)^2}} \quad (2)$$

2 基于受欢迎度和互信息熵理论的信息推荐方法

2.1 混合协同过滤方法(hybrid collaborative filtering method)

通过上述内容分析可以得出:在云环境下,虽然基于用户的协同过滤和基于项目的协同过滤都有着各自的优点,但任意一种单独实施都无法有效地解决问题。为更好地利用以上 2 种方法的优点,首先提出一种简单的混合 2 种推荐的协同过滤方法,以计算用户 a 对服务 i 的评价预测值 $P_c(r_{a,i})$:

$$P_c(r_{a,i}) = \omega \left[r_a + \frac{\sum_{u \in U} S_{im}(a,u)(r_{u,i} - \bar{r}_a)}{\sum_{u \in U} S_{im}(a,u)} \right] + (1-\omega) \left[\bar{r}_i + \frac{\sum_{j \in I} S_{im}(i,j)(r_{a,j} - \bar{r}_i)}{\sum_{j \in I} S_{im}(i,j)} \right] \quad (3)$$

其中: ω 为可信(confidence)权重, $\omega \in [0,1]$,通常在数据规模较小时设置为较大值,当数据规模很大时设置为较小值,主要用于调节这 2 种方法在混合推荐中所占的权重,从而充分发挥这 2 种推荐方法各自的优势。

Algorithm 1: Greedy search for Top- N services

Input: Service set R

Output: Top- N similarity services set X

1. for $j \rightarrow 1$ to m
2. for $i \rightarrow 1$ to m .
3. Calculate $S_{im}(i,j)$
4. $x \leftarrow S_{im}(i,j)$
5. if x among the top- n largest values
6. $x \leftarrow 0$
7. else
8. add x to X
9. end if
10. end for
11. end for

12. Return X

在通常情况下,使用 Top- N 推荐算法确定与目标用户最相似的 N 个用户,并将这 N 个用户的相关服务推荐给目标用户。算法 1 给出了 Top- N 算法的贪婪搜索思想,当数据规模较小时,该种算法具有较好性能。

但在云环境下当服务规模及数量巨大时,上述方法的计算复杂度较大。因此,在这种情况下首先使用概率的潜在语义索引 PLSI(probabilistic latent semantic indexing)对用户进行社团划分,而后使用最小优先独立排列哈希 MinHash(minwise independent permutation hashing)^[10]建立用户索引,从而保证可扩展性的需要,见图 1。然后,使用随机行走策略(算法 2)在社团中随机行走,从而避免遍历所有的服务集,大大降低计算复杂度,见图 2。

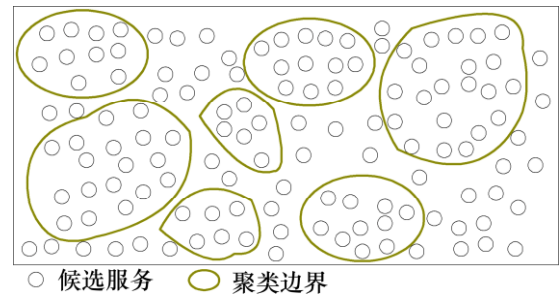


图 1 聚类分析

Fig. 1 Cluster classification

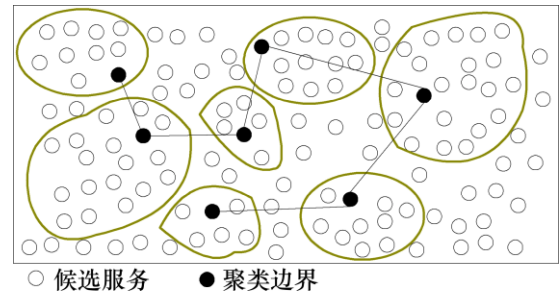


图 2 类间的随机行走策略

Fig. 2 Random walk strategy in clusters

2.2 新用户推荐策略

对于新用户,推荐系统中并不具备任何关于目标用户的信息,因此,传统的协同过滤算法无法有效地为新用户进行服务推荐,即出现“冷启动”(cold start)问题。为了有效地解决此类问题,推荐系统必须设法获得用户的一些信息,最有效的办法就是直接向用户提问题,并提供服务给用户进行评价^[12]。但在现实生活中应用这种方法存在着较大局限性:若向用户提出太多的问题,则会极大地降低系统的可用性,导致用户难以忍受复杂的使用流程;而若缺乏足够的问题,

则又无法有效地获得用户信息。

Algorithm 2: Service set selection algorithm

Input: Available services in the cloud, services number limit N

Output: Service set R

1. $R = \{ \}$
2. Pick a random service with required functionality on the cloud, add it to R
3. While ($n < N$)
4. Pick a random service on the cloud similar to selected service
5. $n++$
6. add it to R
7. Return R

基于欢迎度进行推荐的方法是一种对新用户推荐较有效的方法,但该方法存在的主要问题是:若仅仅基于该策略进行信息推荐,则将使推荐系统无法实现个性化推荐,因为最受欢迎的信息往往也会受到大家的欢迎,而这根本无法体现出个性之间的差异;另外,基于信息熵的推荐方法能够从每次评价值中获取大量的信息,从而能够对新用户推荐起到积极作用^[13],但在海量数据规模下,被评价的信息数量所占的比例很低,常常出现数据稀疏性问题(sparsity problem),单独使用基于信息熵的方法将导致没有信息熵可以利用,从而无法进行推荐。可根据以下步骤解决这些问题:

- 1) 将服务集 R 中的所有备选服务按照受欢迎度的降序排列。
- 2) 基于文献[14]提出的关系划分方法将用户按照不同关系划分成若干社区。
- 3) 当有新的待推荐用户时,判断该用户是否与现有用户存在关系。若存在关系,则进行用户相似度计算;若不存在关系,则从服务集 R 中随机抽选不同受欢迎度的服务提供给用户进行评价。

采用前面提出的混合推荐方法进行新用户推荐。

2.3 基于隐马尔科夫模型的冗余信息消解策略

前面讨论了静态条件下的信息推荐。然而,在实际生活中,服务的 QoS 数据随时可能因环境的变化而发生改变;此外,服务的提供商也很可能对其他服务提供商所提供的服务进行简单复制、粘贴就发布出来,这导致大量的冗余服务出现,而这些冗余服务的质量无法得到保障,且可能存在大量的恶意行为。用户被大量的冗余信息所掩盖,无法区分到底哪一个才是真正需要的服务,隐藏在迷雾之后的就是用户对系统的极大不信任。为了解决这些问题,提出一种在动态环

境下预测和获取服务真实 QoS 值的方法。该方法主要从冗余检测(redundancy detection)和 QoS 控制(QoS control)进行研究。文献[15]讨论了 QoS 控制问题,为此,这里研究冗余检测的解决方法。

解决问题的关键就是确定冗余信息源,并且找出能够符合用户需求的数据信息。为此,收集可能会影响服务未来状态的信息,并周期性地更新服务信息以获取最新数据。

将数据源划分为两类:原始数据源(original data)和冗余数据源(redundant data)。

定义 1:原始数据源。原始数据源指能够自由改变和自主更新的数据源,不受其他服务的影响。

定义 2:冗余数据源。冗余数据源指从原始数据源复制信息(全部或部分)的数据源,有 2 种情况:一是冗余数据源从原始数据源复制信息后,自主进行更新,变成新的原始数据源;另一种是从原始数据源复制信息后停止更新^[16],或者只在原始数据源更新信息后的部分时段进行更新。

由于第 1 种情况的冗余数据源在复制原始数据后变成了 1 个新的原始数据源,对它的判别主要是基于可信度判别,传统的推荐系统可以解决此问题;而第 2 种情况的冗余数据源在复制完成后依然是冗余数据,且随时可能进行变化和修改,传统的推荐系统无法有效确定其冗余属性,该结果依然会出现在推荐列表中,为此,这里主要研究第 2 种情况的冗余数据源并解决相关问题。

在通常情况下,当 2 个服务之间存在很多共同的错误时可以断定存在冗余数据源,而一般的解决方法是记录服务的使用和更新信息,通过记录服务更新的频率来判断冗余数据源。这种方法简单易行,但当冗余数据源存在恶意攻击意图时,它可以随机更新,纯粹依靠记录更新频率无法有效鉴别冗余数据源,为此,引入隐马尔科夫模型(hidden Markov model)^[17]来解决此类问题。隐马尔科夫模型可以解决 3 类基本问题:evaluation problems, decoding problems 和 learning problems。与马尔科夫模型不同是,隐马尔科夫模型中的状态对于用户是不可见的,且每种状态可以随机生成 1 种观察值,为此,首先给出隐马尔科夫模型。

定义 3:隐马尔科夫模型(HMM)。1 个隐马尔科夫模型可以定义为 1 个三元组 $M = (A, B, \pi)$ 。其中: $A = (a_{ij})$ 为转移概率矩阵,定义为从一个状态转移到另一个状态的概率; $B = (b_{ij})$ 为观察值概率矩阵,定义为任意给定状态下观察值的概率分布; $\pi = (\pi_i)$ 为初始概率分布,定义为每个状态作为初始状态的概率分布。设 $O = (O_1, O_2, \dots, O_n)$ 为观察值序列,而

$S = (S_1, S_2, \dots, S_n)$ 为状态序列。

本文中用于确定冗余服务的 HMM 模型实例如下。

隐藏状态 $S = (S_1, S_2)$ 。

观察值 $O = (O_1, O_2)$ 。

转移概率 $a_{ij} = P(s_i | s_j)$, 表示服务在 t 时刻从一种状态 s_i 转移到另一种状态 s_j 的概率。

观察概率 $b_i = P(o_j | s_i)$, 表示服务在状态 s_i 观察值发生变化的概率。

初始概率 $\pi_i = P(s_i)$ 表示服务处在初始状态 s_i 的概率。

至此得到 HMM 模型求解所需的所有必备要素。在 HMM 模型的求解过程中, 由于可以得到服务的若干观察值, 如 $O = (O_1, O_2, \dots, O_n)$, 这样, 实际 HMM 的求解过程就可以转化为 HMM 的经典问题 learning problem 的求解过程, 能够通过一系列方法求解出 HMM 的所有参数 $M = (A, B, \pi)$, 并得出能够最佳匹配观察值序列的 π 。以下简要介绍求解该问题的步骤 (算法 3), 相关推导见文献[17]。

第 1 步, 定义前向变量 $\alpha_t(i)$ 为给定模型 M 时 t 时刻以前的观察值序列和在 t 时刻服务处在状态 s_i 的条件概率, 如下式所示:

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, q_t = s_i | M); 1 \leq t \leq T \quad (4)$$

然后, 通过使用归纳法求解 $\alpha_t(i)$ 。

1) 在初始状态下,

$$\alpha_1(i) = \pi_i b_i(O_1); 1 \leq t \leq T \quad (5)$$

2) 在归纳过程中,

$$\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) a_{ij}] b_j(O_{t+1}); 1 \leq t \leq T-1; 1 \leq j \leq N \quad (6)$$

3) 终止状态为

$$P(O/M) = \sum_{i=1}^N \alpha_n(i) \quad (7)$$

第 2 步, 定义后向变量 $\beta_t(i)$ 为给定模型 M 和 t 时刻服务处在概率 s_i 时从 $t+1$ 时刻到终止时刻的部分观察值序列的条件概率:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_n | q_t = s_i, M); 1 \leq t \leq T-1 \quad (8)$$

同样, 采用归纳法求解 $\beta_t(i)$ 。

1) 在初始状态下,

$$\beta_n(i) = 1; 1 \leq i \leq N \quad (9)$$

2) 在归纳过程中,

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j); \\ t = n-1, n-2, \dots, 1; 1 \leq i \leq N \quad (10)$$

3) 在终止状态下,

$$P(O/M) = \sum_{i=1}^N \beta_1(i) \quad (11)$$

第 3 步, 定义 $\theta_t(i, j)$ 为给定观察值序列和模型 M 时服务状态从 t 时刻的 s_i 转移为 $t+1$ 时刻的 s_j 的条件概率:

$$\theta_t(i, j) = P(s_t = i, s_{t+1} = j | O, M) \quad (12)$$

综合利用式(5)~(12), 可以得到 $\theta_t(i, j)$ 的表达式如下:

$$\theta_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (13)$$

第 4 步, 定义 $\gamma_t(i)$ 为给定观察值序列和模型 M 时服务在 t 时刻处在状态 s_i 的条件概率, 则可以将 $\gamma_t(i)$ 与 $\theta_t(i, j)$ 对应起来, 如下式所示:

$$\gamma_t(i) = \sum_{j=1}^N \theta_t(i, j) \quad (14)$$

然后, 依据式(15)~(17)获得 M 的各个参数值:

$$\hat{\pi}_i = \gamma_1(i) \quad (15)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{n-1} \theta_t(i, j)}{\sum_{t=1}^{n-1} \gamma_t(i)} \quad (16)$$

$$\hat{b}_j(k) = \frac{\sum_{t: O_t=k} \gamma_t(j)}{\sum_t \gamma_t(j)} \quad (17)$$

最后, 利用 Baum-Welch 算法实现冗余服务的判断。

Algorithm 3: RDM (redundancy detection method)

Input: Observable sequence $O = (O_1, O_2, \dots, O_n)$

Output: $M = (A, B, \pi)$

1. Sample the service set as the candidate service S^t
2. Initial M_0
3. Estimation
4. If $\lg P(O|M) - \lg P(O|M_0) < \sigma$
5. $M = M_0$
6. Else

7. $M_0=M$
8. got step 4, re-estimation
9. End if
10. Output

图3所示为本文所提算法的系统实现模型。图3主要表现了系统的各个组成构件、构件之间的关系以及各种输入输出数据等。首先用 PLSI 对候选服务进行聚类, 而后采用混合推荐算法和基于 random walk 策略的 Top-N 贪婪搜索方法获取最符合需求的服务。为提高服务质量, 还综合考虑新用户推荐、QoS 控制和冗余检测等多项工作。

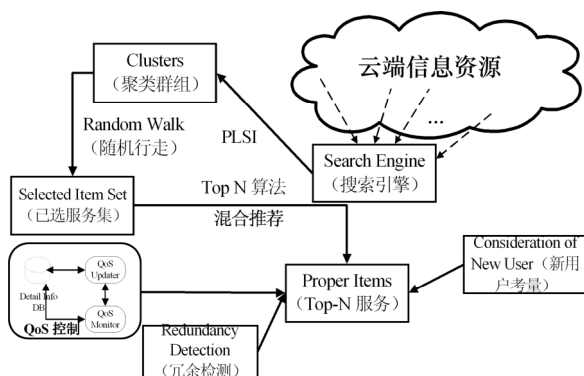


图3 服务搜索的系统模型框架

Fig. 3 System model for services search

3 实验及性能分析

3.1 实验设计

为验证本文所提算法的性能, 设计一组实验, 实验采用 Matlab R2009a 进行编程, 相关服务部署在 Axis1.4.2 上。实验数据为 MovieLens 数据集^[18]和通过公开 API 收集到的 10 000 条 Web 服务数据。其中 MovieLens 数据集包含 6 040 个用户和 3 900 部电影, 以及每个用户对电影进行 1~5 星的评分, 该数据集被广泛用于验证推荐系统性能。而对于实验使用的 Web 服务数据, 只计算其中响应时间及可用性这 2 项 QoS 指标。并且在实验过程中, 采用文献[19]中描述的方法对每个服务的 QoS 数据进行归一化处理。

实验的评价指标是在信息推荐领域广泛使用的指标 E_{MA} , 其表达式如下:

$$E_{MA} = \frac{\sum_{i,j} |r_{i,j} - P(r_{u,i})|}{N} \quad (18)$$

其中: $r_{i,j}$ 为服务的实际 QoS 值; $P(r_{u,i})$ 为预测值。

E_{MA} 越小, 表示预测精度越高。

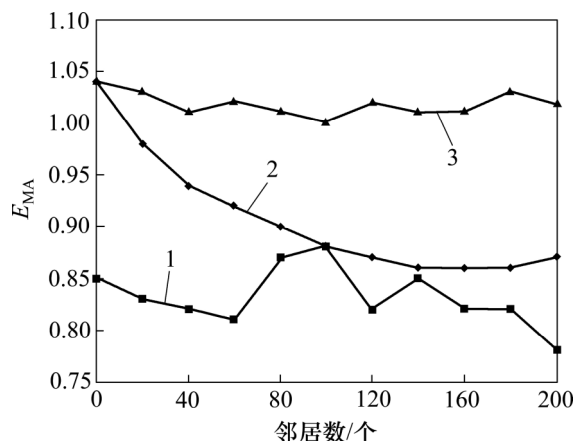
3.2 实验结果及分析

实验的对比方法包括基于用户的协同过滤方法 (user-based CF)、平均预测方法 (average prediction)、贪婪搜索算法 (greedy) 和随机预测方法 (random) 等。用这些方法进行准确率对比分析实验、新用户影响分析实验、冗余消减策略应用分析实验和延迟分析实验。

3.2.1 准确率对比分析

主要基于 MovieLens 数据集进行实验, 以用户邻居数为横坐标, 以 E_{MA} 为纵坐标验证算法性能。实验结果如图4所示。

从图4可见: 1) 当服务数量增加时, 可供评价的服务数量也增加, 而这将同时提供混合推荐算法和基于用户方法的推荐准确性, 但对平均预测方法不造成影响; 2) 混合推荐算法很好地平衡了基于用户的协同过滤方法和基于项目的协同过滤方法的优缺点, 因此, 对于性能的改善有积极作用。



1—Hybnd CF; 2—User based CF; 3—Average prediction.

图4 不同方法的准确率对比分析

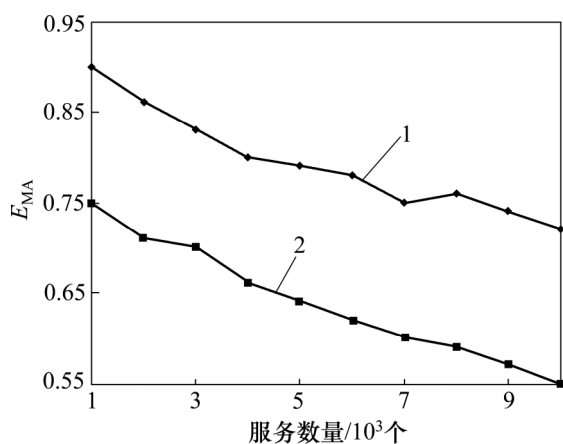
Fig. 4 Accuracy comparison of different methods

3.2.2 新用户影响分析

该实验建立在公共服务数据集上, 结果如图5所示。实验结果表明: 考虑了新用户偏好的混合推荐算法性能明显优于未考虑新用户偏好的算法性能。而由于新用户没有历史信息, 因此, 基于熵的方法无法提供有效的推荐。

3.2.3 冗余消减策略应用分析

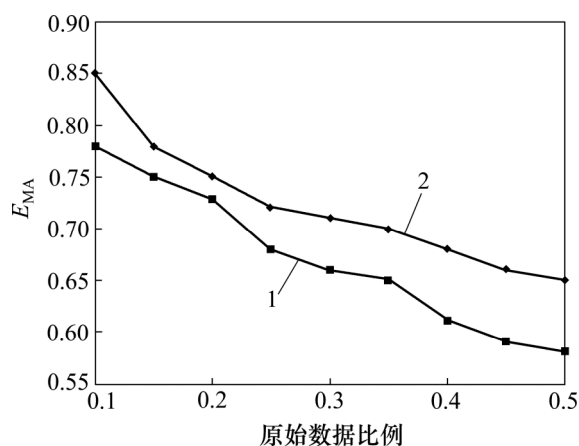
在该实验中, 使用 HMM 模型检测冗余服务。实验结果见图6。从图6可见: 使用了冗余检测方法的算法性能明显优于未使用冗余检测方法的算法性能, 而当冗余服务比例较大时该算法的性能优势更加突出。



1—未考虑新用户偏好的混合推荐算法；
2—考虑新用户偏好的算法。

图 5 新用户影响分析

Fig. 5 Impact of new user



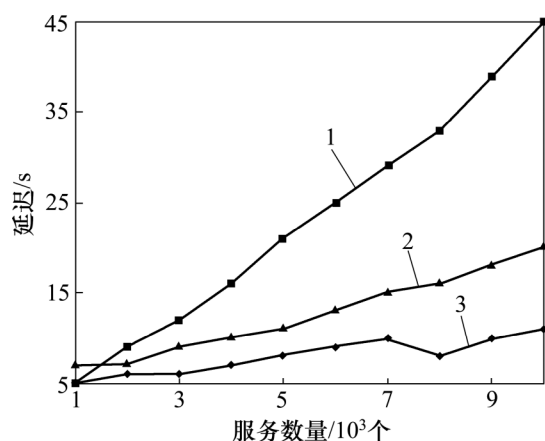
1—使用了冗余检测方法的算法；
2—没使用冗余检测方法的算法。

图 6 冗余消解策略应用分析

Fig. 6 Impact of redundancy detection

3.2.4 延迟分析

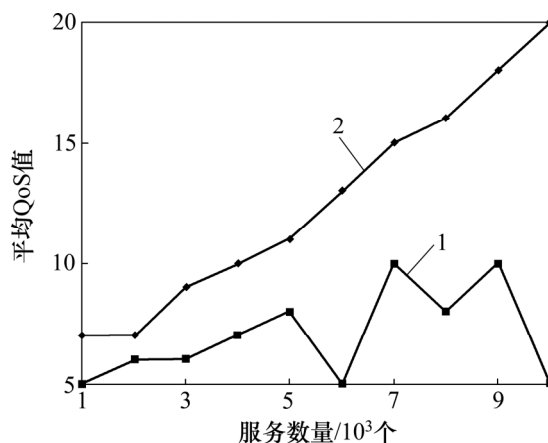
进一步地进行延迟分析(图 7)以及平均 QoS 值比较,结果分别见图 7 和图 8。对比方法为贪婪搜索算法(greedy)、随机预测方法(random)和冗余检测算法(RDM)。由于服务数量增加,搜索最优服务的时间必然会延长,因此,贪婪搜索算法在延迟上呈指数级增加是用户无法接受的,而 RDM 算法的延迟仅仅稍高于 Random 算法的延迟,平均 QoS 值却得到极大提升,这表明 RDM 对于最优服务的选择和推荐具有很重要的所用。而其原因在于 RDM 使用了有效的聚类方法和随机行走策略。



1—贪婪搜索算法；2—随机预测方法；3—冗余检测算法。

图 7 不同算法的延迟分析

Fig. 7 Delay comparison of different algorithms



1—随机预测方法；2—冗余检测算法。

图 8 不同算法的平均 QoS 指标对比分析

Fig. 8 Average QoS comparison of different algorithms

4 相关工作

与本文工作相关的研究主要包括:基于 QoS 的服务选择(QoS-based service selection)、协同过滤推荐(collaborative filtering based recommendation)、新用户偏好识别(learning of new user needs)和冗余检测(redundancy detection)。

近年来,基于 QoS 服务选择的研究很多^[2, 4, 5, 20-21]。BALKE 等^[20]综合利用了服务使用模式和用户需求偏好来扩大服务发现的范围,并借此提高候选服务的质量;文献[22]提出了一种基于预测的 QoS 管理算法,通过使用在线配置文件以及抽样方法来评估动态服务数据的代价并借此调整服务 QoS,以满足不同需求。

协同过滤推荐已被广泛用于发现相似用户或相似产品^[6-8, 22]。主要的协同过滤方法包括基于用户的协同过滤方法和基于项目的协同过滤方法,但这2种方法各自存在无法克服的缺点。为此,文献[23]提出了一种混合推荐算法以给出准确的推荐,但该方法对新用户偏好欠考量;文献[24]利用了贝叶斯网络(Bayesian network inference)进行信息推荐;而文献[25]利用社会信誉度模型来进行在线社会网络中的信息推荐。

对于新用户偏好的缺点,绝大多数的研究集中于信息的受欢迎度分析,但由于将最受欢迎的信息推荐给用户无法有效地发现用户的个性偏好,因此,这种方法并不可靠。为此,文献[12]提出一种综合利用信息熵(entropy)和受欢迎度的方法;而文献[10]则利用“访问了这件商品的用户,又访问了另一件商品”的模式(co-visitation)进行了新用户的推荐。

目前人们对信息冗余检测的研究很少,部分相关研究包括:文献[1]提出了一种通过构建用户相关度来过滤无关信息的方法,而文献[14]则提出了一种用户社会关系划分的方法,并通过这种方法有效地发现用户的兴趣,以过滤无关信息。

5 结论

1) 为有效发现云环境下的最优服务,提出了一系列信息推荐方法。首先对协同过滤方法进行分析,并基于历史信息提出混合推荐方法;而后通过对新用户特点进行分析,提出了新用户的偏好获取方法;由于冗余信息会对最优服务的选择造成巨大影响,因此,基于隐马尔科夫模型进一步提出冗余信息的消解方法。最后,依托真实数据集和通过公共API获取的服务进行实验,实验结果表明上述算法在推荐的准确性和时效性等方面都取得了较大提升。

2) 基于混合推荐的方法仅仅考虑了基于内存的协同过滤方法(memory-based collaborative filtering),还没有考虑网络的本身的结构拓扑,在下一步工作中,需综合考虑网络拓扑,并预测动态服务和推荐上以准确捕捉用户偏好的动态变化。

参考文献:

- [1] LV Y H, MOON T, KOLARI P, et al. Learning to model relatedness for news recommendation[C]// Proceedings of the 20th International Conference on World Wide Web. Hyderabad, India: ACM, 2011: 57-66.
- [2] HUH E, WELCH L R, TJADEN B, et al. Accommodating QoS prediction in an adaptive resource management framework[C]// Parallel and Distributed Processing-15 IPDPS 2000 Workshops. Cancun, Mexico: IEEE, 2000: 792-799.
- [3] SMITH W, FOSTER I, TAYLOR V. Predicting application run times using historical information[J]. Journal of Parallel and Distributed Computing, 2004, 64(9): 1007-1016.
- [4] BLAKE M B, NOWLAN M F. Predicting service mashup candidates using enhanced syntactical message management[C]// 2008 IEEE International Conference on Services Computing. Honolulu, USA: IEEE, 2008: 229-236.
- [5] MOONEY R J, ROY L. Content-based book recommendation using learning for text categorization[C]// Proceedings of the ACM International Conference on Digital Libraries. Kyoto, Japan: ACM, 2000: 195-204.
- [6] SHAO L S, ZHANG J, WEI Y, et al. Personalized QoS prediction for web services via collaborative filtering[C]// IEEE International Conference on Web Services. Salt Lake City, USA: IEEE, 2007: 1-8.
- [7] SU X Y, KHOSHGOFTAAR T M. A survey of collaborative filtering techniques[J]. Advances in Artificial Intelligence Archive, 2009.doi:10.1155/2009/421425.
- [8] MA H, KING I, LYU M R. Effective missing data prediction for collaborative filtering[C]// Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Amsterdam, Netherlands: ACM, 2007: 39-46.
- [9] RESNICK P, LAKOVOU N, SUSHAK M, et al. GroupLens: An open architecture for collaborative filtering of net news[C]// International Conference on Computer Supported Cooperative Work. Chapel Hill, USA: ACM, 1994: 175-186.
- [10] DAS A, DATAR M, GARG A. Google news personalization: scalable online collaborative filtering[C]// Proceedings of the 16th International Conference on World Wide Web. Banff, Canada: ACM, 2007: 271-280.
- [11] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms[C]// Proceedings of the 10th International Conference on World Wide Web. Hong Kong: ACM, 2001: 285-295.
- [12] RASHID A M, ALBERT I, COSLEY D, et al. Getting to know you: Learning new user preference in recommender systems[C]// International Conference on Intelligent User Interfaces. San Francisco, USA: ACM, 2002: 127-134.
- [13] KOHRS A, MERIALDO B. Improving collaborative filtering for new users by smart object selection[C]// Proceedings of International Conference on Media Feature. Florence, Italy: IEEE, 2001.
- [14] TANG S J, YUAN J, MAO X F, et al. Relationship classification

- in large scale online social networks and its impact on information propagation[C]// IEEE International Conference on Computer Communications. Shanghai, China: IEEE, 2011: 2291–2299.
- [15] CHEN H H, MA J W, HUANGFU X P, et al. Process reservation for service oriented application[C]// 6th World Congress on Services. Miami, USA: IEEE, 2010: 162–163.
- [16] DONG X L, BERTI-ENQULE L, SRIVASTAVA D. Truth discovery and copying detection in a dynamic world[J]. Proceedings of the Vldb Endowment, 2009, 2: 562–573.
- [17] RABINAR L R. A tutorial on hidden markov models and selected applications in speech recognition[J]. Proceedings of the IEEE, 1989, 77(2): 257–286.
- [18] Grouplens Research. MovieLens data sets[EB/OL]. 2010. <http://grouplens.org/datasets/movielens/>
- [19] LIU Y T, NGU A, ZENG L Z. QoS computation and policing in dynamic web service selection[C]// Proceedings of the 13th International Conference on World Wide Web. New York, USA: ACM, 2004: 66–73.
- [20] BALKE W, MATTHIAS W. Towards personalized selection of web services[C]// Proceedings of the 12th International Conference on World Wide Web. Budapest, Hungary: ACM, 2003: 20–24.
- [21] LI M, HUAI J P, GUO H P. An adaptive web services selection method based on the QoS prediction mechanism[C]// IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops. Milan, Italy: IEEE, 2009: 395–402.
- [22] WEI Y, PRASAD V, SANG H, et al. Prediction-based QoS management for real-time data streams[C]// Proceedings of the 27th IEEE International Real-Time Systems Symposium. Rio de Janeiro, Brazil: IEEE, 2006: 344–355.
- [23] ZHENG Z B, MA H, LYU M R, et al. WSRec: A collaborative filtering based web service recommender system[C]// 2009 IEEE International Conference on Web Services. Los Angeles, USA: IEEE, 2009: 437–444.
- [24] YANG X W, GUO Y, LIU Y. Bayesian-inference based recommendation in online social networks[C]// IEEE International Conference on Computer Communications. Shanghai, China: IEEE, 2011: 551–555.
- [25] CHEN R C, LUA E K, CAI Z H. Bring order to online social networks[C]// IEEE International Conference on Computer Communications. Shanghai, China: IEEE, 2011: 541–545.

(编辑 陈灿华)