

DOI: 10.7652/xjtuxb201612014

混合因子矩阵分解推荐算法

赵长伟^{1,2}, 彭勤科¹, 张志勇²

(1. 西安交通大学电子与信息工程学院, 710049, 西安; 2. 河南科技大学信息工程学院, 471023, 河南洛阳)

摘要: 针对矩阵分解推荐算法在潜在属性与已知属性之间不能建立对应关系的问题,提出了一种混合显式属性与隐式属性的矩阵分解算法。该算法使用显式属性的相关性对因子矩阵进行约束,能够抑制稀疏数据矩阵分解中过拟合的问题,提高推荐精度,由于因子矩阵中包含显式属性,所以混合因子矩阵分解算法可以实现对新用户和新产品推荐,部分地解决了冷启动问题,实现了从评分数据到显式属性的映射,并对推荐结果给出一定的解释。在 MovieLens 数据集上的实验结果表明:相同因子数下,混合因子矩阵分解算法的推荐精度均优于偏置概率矩阵分解算法,并能够基于显式属性实现对新产品的推荐。

关键词: 推荐算法;矩阵分解;混合因子;推荐解释;冷启动

中图分类号: TP393;G558 **文献标志码:** A **文章编号:** 0253-987X(2016)12-0087-05

A Matrix Factorization Algorithm with Hybrid Implicit and Explicit Attributes for Recommender Systems

ZHAO Changwei^{1,2}, PENG Qinke¹, ZHANG Zhiyong²

(1. School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;

2. School of Information and Engineering, Henan University of Science and Technology, Luoyang, Henan 471023, China)

Abstract: A novel hybrid matrix factorization algorithm (HMF) is proposed to solve the problem that the correlation between latent factors and explicit attributes can not be established in traditional matrix factorization methods. The algorithm combines implicit and explicit attributes and uses correlations among explicit attributes to constrain factor matrixes, and to relieve the over fitting in sparse data matrix decomposition. Since factor matrixes include explicit attributes, HMF is used to solve the problem of cold start and to recommend new items. HMF realizes mapping from rating matrix to weights of explicit attributes and offers an interpretation for recommender items. Experiment on MovieLens datasets shows that the accuracy of HMF is superior to that of BPMF for same number of factors, and HMF can be used to recommend new items based on explicit attributes.

Keywords: recommender algorithm; matrix factorization; hybrid factor; recommended interpretation; cold start

在应用驱动下,个性化推荐系统得到了学术界和商业界共同的重视^[1-3],推荐精度不断提高。矩阵分解(matrix factorization,MF)算法是个性化推荐中的经典算法。基本矩阵分解算法^[4-5]将用户产品

打分矩阵分解为用户潜在因子矩阵和产品潜在因子矩阵,用户对产品的打分预测值由用户潜在因子矩阵和产品潜在因子矩阵的内积求得。概率矩阵分解(probability matrix factorization, PMF)算法是应用较多的矩阵分解算法,在数据稀疏的情况下,PMF算法容易出现过拟合现象。为了防止数据过拟合,Netflix 竞赛获胜者采用了正则化的矩阵分解算法(RSVD)^[1],正则化能够提高整体打分预测的精度,在矩阵分解算法中被广泛采用。考虑到不同用户和产品具有不同打分偏置,Rendle 等使用了偏置概率矩阵分解算法(BPMF)^[6],该算法能够适应不同用户和产品的打分偏置,并有效地提高了打分预测的精度。PMF 和 BPMF 算法仅考虑用户对产品的打分,并假定用户和产品的潜在属性是符合独立同分布的。含用户社交关系和产品属性相关性的矩阵分解算法^[7-10]将用户社交关系和产品已知属性的相关性作为矩阵分解的约束,从而提高了整体的推荐精度和推荐项的覆盖率。在推荐结果解释方面,非负矩阵分解(NMF)算法^[11]限定属性取值为非负,与属性的有无和权重值相对应,其结果具有一定的解释意义,但 NMF 算法没有给出潜在因子与显式属性的对应关系。针对矩阵分解算法的效率问题,Ortega 等使用了群矩阵分解算法^[12],但群矩阵分解算法本质上属于正则化算法。针对推荐数据稀疏的问题,Zhao 等使用迁移学习的方法^[13-14],融合不同领域的的数据,解决了推荐系统数据稀疏的问题。

在传统的矩阵分解算法中,由于使用潜在属性因子,所以不能对推荐结果给出解释,也不能确定对特定用户推荐结果是否有改善。基于此,本文提出了一种结合显式属性和隐式属性的矩阵分解算法(hybrid matrix factorization, HMF)。HMF 算法的创新点主要有 3 个方面:①在矩阵分解过程中,使用用户或产品之间显式属性的相关性对因子矩阵进行约束,防止稀疏数据矩阵分解中的过拟合问题;②利用 HMF 算法包含的用户和产品的显式属性,可以实现对新用户和新产品推荐,部分地解决了冷启动问题;③实现了打分矩阵到显式属性的映射,能够对推荐结果给出一定的解释。

1 混合矩阵分解算法

1.1 矩阵分解算法

对于 m 行 n 列的用户产品打分矩阵 $\mathbf{R}$, m 为用户数, n 为产品数, r_{ij} 是矩阵 $\mathbf{R}$ 中第 i 行第 j 列的元素,表示第 i 个用户对第 j 个产品的打分。由于打

分数据十分稀疏, $\mathbf{R}$ 中存在大量的缺失数据。基本的矩阵分解算法在已知打分数据项上将打分矩阵 $\mathbf{R}$ 分解为 m 行 k 列的用户潜在属性因子矩阵 $\mathbf{U}$ 和 n 行 k 列的产品潜在属性因子矩阵 $\mathbf{V}$, 其中 k 为潜在属性因子数($k \ll m, n$), 并使 $\mathbf{UV}^T$ 尽可能接近 $\mathbf{R}$ 。在基本矩阵分解算法中,由于使用了潜在属性因子,所以基本的矩阵分解算法不能给出推荐的解释。此外,该方法假定属性因子之间满足独立同分布条件,没有考虑用户和产品之间的关联关系对矩阵分解的影响。

结合用户和产品属性相关性的矩阵分解算法,对分解的因子矩阵进行相关性约束,使代价函数尽可能小,代价函数
$$\sum_{r_{ij} \in \mathbf{R} \& r_{ij} \neq \Phi} (r_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \alpha \sum_{i,j \in c_1} \mathbf{U}_i \mathbf{U}_j^T + \beta \sum_{i,j \in c_2} \mathbf{V}_i \mathbf{V}_j^T$$
 中, c_1 、 c_2 分别表示用户和产品之间的关联关系, α 、 β 分别表示用户和产品之间的关联关系的权重。使用相关关系对潜在矩阵分解进行约束,能够有效解决矩阵分解算法中打分数据稀疏的问题,防止数据过拟合,提高整体的打分预测精度。

1.2 混合属性因子矩阵分解算法

HMF 算法中,用户和产品因子矩阵 $\mathbf{U}$ 、 $\mathbf{V}$ 由显式属性因子和隐式属性因子组成,隐式因子也被称为潜在因子。显式属性能够表示用户或产品之间已知属性的相关性,隐式属性表示不能明确表示的推荐因子。HMF 算法中,用户属性矩阵 $\mathbf{U} = [\mathbf{U}^{\text{ex}} \quad \mathbf{U}^{\text{im}}]$, 其中, m 行 k_1 列的矩阵块 $\mathbf{U}^{\text{ex}}$ 为显式属性矩阵块, m 行 k_2 列的矩阵块 $\mathbf{U}^{\text{im}}$ 为隐式属性矩阵块, k_1 为显式属性数, k_2 为隐式属性数, $k_1 + k_2 = k$ 为用户属性因子数。对于产品属性矩阵,同样有 $\mathbf{V} = [\mathbf{V}^{\text{ex}} \quad \mathbf{V}^{\text{im}}]$ 。

当产品因子矩阵中包含显式属性,用户因子矩阵采用隐式属性因子表示时,混合因子矩阵分解算法的代价函数定义为

$$L(\mathbf{U}, \mathbf{V}) = \sum_{r_{ij} \in \mathbf{R} \& r_{ij} \neq \Phi} (r_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 \quad (1)$$

混合因子矩阵分解算法通过优化如下目标函数实现

$$\min \sum_{r_{ij} \in \mathbf{R} \& r_{ij} \neq \Phi} (r_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 \quad (2)$$

为了防止数据过拟合,对 $\mathbf{U}$ 、 $\mathbf{V}$ 进行正则化,相应的目标函数为

$$\min \sum_{r_{ij} \in \mathbf{R} \& r_{ij} \neq \Phi} (r_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \lambda (\|\mathbf{U}_i\|^2 + \|\mathbf{V}_j\|^2) \quad (3)$$

式中: λ 为正则化系数。为了使矩阵分解结果具有解释性,限制产品因子属性值为非负值,目标函数为

$$\min \sum_{r_{ij} \in R^0, r_{ij} \neq \phi} (r_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \lambda (\|\mathbf{U}_i\|^2 + \|\mathbf{V}_j\|^2) \quad \text{s. t. } v \geq 0 \quad (4)$$

式中: $v \in V$ 为产品因子属性的值。

1.3 问题求解

随机梯度下降算法 (stochastic gradient descent, SGD) 和交替最小二乘法 (alternating least squares, ALS) 是矩阵分解中最常用的算法。本文使用改进的随机梯度下降算法对因子矩阵进行求解。首先使用代价函数对 $\mathbf{U}_i$ 和 $\mathbf{V}_j$ 求偏导,有

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{U}_i} &= -2\mathbf{V}_j(r_{ij} - \mathbf{U}_i \mathbf{V}_j^T) + 2\lambda \mathbf{U}_i \\ \frac{\partial L}{\partial \mathbf{V}_j} &= -2\mathbf{U}_i(r_{ij} - \mathbf{U}_i \mathbf{V}_j^T) + 2\lambda \mathbf{V}_j \end{aligned} \quad (5)$$

然后,在负梯度方向对 $\mathbf{U}_i$ 和 $\mathbf{V}_j$ 迭代更新, $\mathbf{V}_j$ 由隐式属性 $\mathbf{V}_j^{\text{im}}$ 和显式属性 $\mathbf{V}_j^{\text{ex}}$ 两部分组成,其显式属性 $\mathbf{V}_j^{\text{ex}}$ 保持不变。则有

$$\begin{aligned} \mathbf{U}_i &\leftarrow \mathbf{U}_i + \alpha(\mathbf{V}_j(r_{ij} - \mathbf{U}_i \mathbf{V}_j^T) - \lambda \mathbf{U}_i) \\ \mathbf{V}_j^{\text{im}} &\leftarrow \mathbf{V}_j^{\text{im}} + \alpha(\mathbf{U}_i(r_{ij} - \mathbf{U}_i \mathbf{V}_j^T) - \lambda \mathbf{V}_j) \end{aligned} \quad (6)$$

式中: α 为学习速率。为了保证属性值具有解释的意义,当 $\mathbf{V}_j^{\text{im}}$ 中出现 $v < 0$ 时,取 $v = 0$ 。

在该算法中,产品属性矩阵中包含产品的显式属性 $\mathbf{V}_j^{\text{ex}}$,在迭代过程中,显式属性 $\mathbf{V}_j^{\text{ex}}$ 的值保持不变。 $\mathbf{U}_i \mathbf{V}_j^T$ 中与显式属性 $\mathbf{V}_j^{\text{ex}}$ 对应的用户属性矩阵 $\mathbf{U}^{\text{ex}}$ 的因子值反映了用户 i 对产品 j 显式属性的偏好程度。

2 推荐解释与冷启动

在传统的矩阵分解算法中,由于分解的矩阵使用潜在属性因子,所以不能对推荐结果给出解释,也不能推荐新产品和新用户,即面临冷启动问题。在混合因子矩阵分解算法中,由于包含了用户或产品的显式属性,所以能够给出推荐解释并能一定程度上解决冷启动问题。

将用户潜在属性因子矩阵 $\mathbf{U}$ 分为与产品显式属性特征对应的块 $\mathbf{U}^{\text{ex}}$ 和与产品隐式属性特征对应的块 $\mathbf{U}^{\text{im}}$,则显式属性的打分权重为

$$w_{r_{ij}} = \frac{\mathbf{U}_i^{\text{ex}}(\mathbf{V}_j^{\text{ex}})^T}{\mathbf{U}_i^{\text{im}}(\mathbf{V}_j^{\text{im}})^T + \mathbf{U}_i^{\text{ex}}(\mathbf{V}_j^{\text{ex}})^T} \quad (7)$$

显式属性所对应的权值表示了用户对产品已知属性的偏好程度,进一步,可以使用该权值对推荐结果给出相应的解释。对于已知属性的新产品,当 $w_{r_{ij}}$ 大于一定阈值时,可以依据产品的已知属性向用户推

荐新产品。

针对冷启动问题,考虑对产品因子矩阵采用显式属性表示,在产品因子属性已知的基础上,使用式(8)计算用户因子矩阵的值。

$$\mathbf{U}_i \leftarrow \mathbf{U}_i + \alpha(\mathbf{V}_j^{\text{ex}}(r_{ij} - \mathbf{U}_i(\mathbf{V}_j^{\text{ex}})^T) - \lambda \mathbf{U}_i) \quad (8)$$

这种情况下,用户 i 对已知属性产品 j 的打分预测值为

$$\hat{r}_{ij} = \mathbf{U}_i(\mathbf{V}_j^{\text{ex}})^T \quad (9)$$

与产品矩阵表示相似,当用户矩阵使用混合属性因子表示时,可以实现向新用户推荐产品项。

3 实验结果及分析

3.1 数据集

实验使用了 MovieLens 数据集^[15],MovieLens 数据集由 3 个不同大小数据集构成,其中 100 kB 的数据集中包含 943 个用户对 1 682 部电影的 100 000 条评分数据,评分等级为 1 到 5,共 5 个等级,等级 1 到 5 反映了用户对该电影的偏好程度。每个用户至少对 20 部电影进行了评分。每一条评分数据包含用户、电影、评分级评分时间信息,该数据集还包含简单的用户人口学统计信息和电影的体裁类型信息。

3.2 打分预测精度

推荐系统有多种评价指标,打分预测精度常用的评价指标有绝对误差 E_m 和均方误差 E_r

$$E_m = \frac{\sum_{(u,i) \in T} |r_{ui} - \hat{r}_{ui}|}{|T|} \quad (10)$$

$$E_r = \left(\frac{\sum_{(u,i) \in T} (r_{ui} - \hat{r}_{ui})^2}{|T|} \right)^{1/2} \quad (11)$$

式中: T 为测试集, $|T|$ 表示测试集样本的数目。均方误差 E_r 对预测误差较大的值更敏感。

实验中对不同潜在属性因子数和混合因子数的推荐算法的预测精度进行了比较,实验结果采用 5 折交叉验证的方法。

图 1、图 2 给出了 BPMF 算法在不同因子数 k 下打分预测精度 E_m 和 E_r 变化的情况。实验中仅使用了用户对产品的打分信息,正则化系数 λ 设置为 0.02,学习率 α 设置为 0.1。用户和产品矩阵因子均为潜在属性因子。

由图 1、图 2 可以看出,在因子数 k 值小于确定值的情况下,潜在因子矩阵分解算法打分预测精度随 k 值的增加而增加。为了验证结合显式属性的矩阵分解 HMF 算法的有效性,分别对总因子数固定

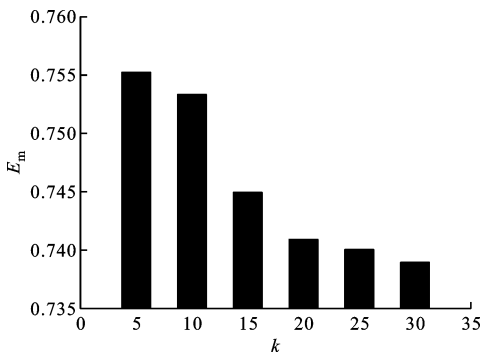


图 1 潜在因子数与绝对误差的关系

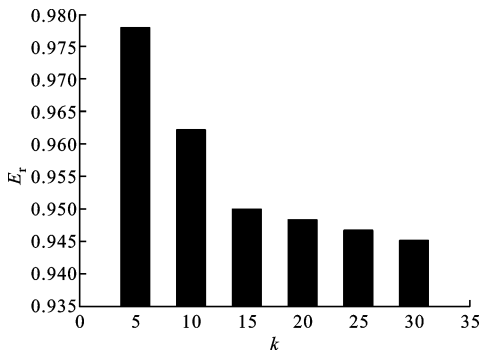


图 2 潜在因子数与均方误差的关系

和显式属性因子数固定的情况进行了实验。表 1 给出了总因子数固定为 5 的情况下,含不同数目显式因子的 HMF 算法的预测精度。实验中正则化系数 λ 设置为 0.02,学习率 α 设置为 0.1,显式属性因子值固定为 0.2。实验中显式因子来自电影体裁信息,在 MovieLens 数据集中,包含了 19 种不同类型的体裁信息,但属于某些体裁数的电影非常少,这些体裁属性在预测精度中没有普遍性。实验中按照不同体裁的电影数对体裁属性进行排序,得到一个体裁属性排序列表。在 HMF 算法中,显式属性数为 n 时,对应的为体裁属性排序列表的 top- n 项的属性。在表 1 中,当显式属性数为 5 时,电影矩阵 $\mathbf{V}$ 中仅包含显式属性。

表 1 不同显式属性因子数时 HMF 算法的预测精度

显式因子数	E_m	E_r
1	0.735 1	0.939 2
2	0.731 6	0.942 1
3	0.739 6	0.946 8
4	0.740 4	0.945 8
5	0.753 2	0.957 9

由表 1 可以看出,当包含两个显式因子时,绝对误差最小为 0.731 6。与 BPMF 算法相比,其预测精度有较大的提高,这是因为 HMF 算法中结合了

显式因子,增加了显式属性的相关约束。同时,该方法中包含了隐式属性,隐式属性能够表示用户偏好和产品属性中难以显式表示的偏好信息。随着显式因子数的增加,显式属性相关性约束增加,隐式属性表示的推荐相关的潜在因子被削弱,导致推荐精度降低。

表 2 中给出了显式因子数固定为 5,总因子数变化范围为 5 到 30 时,HMF 算法的打分预测精度。

表 2 固定显式因子数时 HMF 算法的预测精度

总因子数	E_m	E_r
5	0.753 2	0.957 9
10	0.745 2	0.952 3
15	0.738 8	0.941 6
20	0.743 9	0.953 1
25	0.737 4	0.943 1
30	0.737 2	0.941 1

由表 2 可以看出,在显式属性因子数固定的情况下,HMF 算法的打分预测精度与显式属性因子数占比有一定的关系。在因子数为 15 时,HMF 算法取得较好的结果,显式属性对因子矩阵的相关关系给出了较好的约束。随着因子数的增加,显式属性在全部因子数中的比例降低,相关性约束减小,与 BPMF 算法相比,HMF 算法对推荐精度提高十分有限。

3.3 冷启动和推荐解释

在个性化推荐系统中,传统的 MF 方法不能推荐新产品和新用户的原因是用户偏好和产品属性均采用潜在属性因子表示,用户偏好和产品属性不能明确表示。HMF 算法中,当产品项使用含有显式属性因子矩阵表示时,与其对应的用户因子项可看作用户对该属性的偏好。HMF 算法能够基于用户的偏好和产品显式属性推荐新的产品。

表 3 给出了 MovieLens 数据集中使用电影体裁类型作为显式属性时,HMF 算法和基于产品属性相似度算法在新产品项上的推荐精度。实验中 HMF 和基于产品属性相似度算法均使用了电影体裁项中前 5 个包含电影数较多的属性项。

由表 3 可以看出,在新产品推荐上,HMF 算法的精度远高于基于产品属性推荐方法的精度,这是因为 HMF 算法能够区分用户对产品不同属性项偏好的权重,而基于产品属性的推荐方法对属性的权重不做区分。

表 3 HMF 算法和 Item-based 算法新产品推荐精度

算法	E_m	E_r
HMF	0.753 2	0.805 1
Item-Based	0.957 9	1.050 1

由于 HMF 算法包含显式属性,该方法也能够对推荐结果给出一定的解释。具有推荐解释功能使推荐结果更容易被用户接受。与产品推荐类似,当用户矩阵含有显式属性时,HMF 算法可以实现对新用户的推荐和推荐解释。

4 结 论

本文提出了一种混合因子矩阵分解 HMF 算法,HMF 算法的因子矩阵中包含显式属性和隐式属性。与基本的矩阵分解算法相比,HMF 算法使用了显式属性之间的相关性,对稀疏数据矩阵分解给出了合理的约束,提高了推荐精度。同样,由于使用了显式属性因子,HMF 算法能够推荐新产品并对推荐结果给出一定的解释。本文中显式属性仅使用了电影体裁属性,没有考虑其他显式属性和显式属性选取的问题,下一步,将对显式属性选取进行研究,对用户偏好的深层次原因进行分析,进一步提高推荐系统的推荐精度和推荐解释的能力。

参考文献:

[1] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems [J]. Computer, 2009, 42(8): 30-37.

[2] MA W, FENG X, WANG S, et al. Personalized recommendation based on heat bidirectional transfer [J]. Physica: A Statistical Mechanics and Its Applications, 2016, 444: 713-721.

[3] RICCI F, ROKACH L, SHAPIRA B. Recommender systems handbook [M]. 3rd ed. Berlin, Germany: Springer, 2010: 1-35.

[4] SALAKHUTDINOV R, MNIH A. Probabilistic matrix factorization [C]// Proceedings of the 2015 Advances in Neural Information Processing Systems. Cambridge, MA, USA: MIT Press, 2015: 1257-1264.

[5] SALAKHUTDINOV R, MNIH A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo [C]// Proceedings of the International Conference on Machine Learning. New York, USA: ACM, 2008: 880-887.

[6] RENDLE S, SCHMIDT-THIEME L. Online-updating

regularized kernel matrix factorization models for large-scale recommender systems [C]// Proceedings of the 2008 ACM Conference on Recommender Systems. New York, USA: ACM, 2008: 251-258.

[7] 秦继伟, 郑庆华, 郑德立, 等. 结合评分和信任的协同推荐算法 [J]. 西安交通大学学报, 2013, 47(4): 100-104.

QIN Jiwei, ZHENG Qinghua, ZHENG Deli, et al. A collaborative recommendation algorithm based on ratings and trust [J]. Journal of Xi'an Jiaotong University, 2013, 47(4): 100-104.

[8] 郭磊, 马军, 陈竹敏, 等. 一种结合推荐对象间关联关系的社会化推荐算法 [J]. 计算机学报, 2014, 37(1): 219-228.

GUO Lei, MA Jun, CHEN Zhumin, et al. Incorporating item relations for social recommendation [J]. Chinese Journal of Computer, 2014, 37(1): 219-228.

[9] MA H, KING I, LYU M R. Learning to recommend with explicit and implicit social relations [J]. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3): 135-136.

[10] MA H, KING I, LYU M R, et al. SoRec: social recommendation using probabilistic matrix factorization [C]// Proceedings of the 2008 ACM Conference on Information and Knowledge Management. New York, USA: ACM, 2008: 931-940.

[11] LEE D D, SEUNG H S. Learning the parts of objects by non-negative matrix factorization [J]. Nature, 1999, 401(6755): 788-791.

[12] ORTEGA F, HERNANDO A, BOBADILLA J, et al. Recommending items to group of users using matrix factorization based collaborative filtering [J]. Information Sciences, 2016, 345: 313-324.

[13] ZHAO L, PAN S J, XIANG E W, et al. Active transfer learning for cross-system recommendation [C]// Proceedings of the 27th AAAI Conference on Artificial Intelligence. Palo Alto, CA, USA: AAAI Press, 2013: 1205-1211.

[14] JIANG M, CUI P, WANG F, et al. Social recommendation across multiple relational domains [C]// Proceedings of the ACM International Conference on Information and Knowledge Management. New York, USA: ACM, 2012: 1422-1431.

[15] HARPER F M, KONSTAN J A. The movielens datasets: history and context [J]. ACM Transactions on Interactive Intelligent Systems, 2015, 5(4): 1068-1074.

(编辑 武红江)