

结合评分时间和用户空间的协同过滤推荐算法

李炎 艾均 苏湛

(上海理工大学光电信息与计算机工程学院 上海 200093)

摘要 大多数科学研究的协同过滤推荐算法一般只利用用户的评分信息,而忽视了评分产生的时间以及用户所处的空间对推荐的影响。针对这一问题,从评分时间和用户空间角度对推荐算法的相似度计算、邻居用户选择以及评分预测过程进行改进。利用评分时间和用户空间两方面因素,提出结合评分时间和用户空间的协同过滤推荐算法。用户的兴趣随着时间的改变而发生改变,用户最近评分的影响力比以前评分的影响力更大。对没有空间信息的数据集,采用复杂网络拓扑布局算法把相关联的用户排布成用户空间网络布局。通过实验验证了该算法在准确性方面有显著的提高。

关键词 协同过滤 复杂网络 布局算法 评分时间 用户空间

中图分类号 TP3 文献标识码 A DOI:10.3969/j.issn.1000-386x.2018.12.046

COLLABORATIVE FILTERING RECOMMENDATION ALGORITHM COMBINING RATING TIME AND USER SPACE

Li Yan Ai Jun Su Zhan

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract In scientific research, most of the collaborative filtering recommendation algorithm generally only uses the user's rating information. It ignores the influence of time generated by rating and user space on recommendation. To solve this problem, we improved the similarity computing, neighbor's choice and rating prediction process of recommendation algorithm from the perspective of rating time and user space. Combining rating time and user space, we proposed a collaborative filtering recommendation algorithm. With time changing, users' interests change. Their recent ratings are more influential than previous ratings. For data sets without spatial information, the topology layout algorithm of complex network was adopted to arrange related users into the user-space network layout. Experiments verify that the algorithm has a significant improvement in accuracy.

Keywords Collaborative filtering Complex network Layout algorithm Rating time User space

0 引言

随着科学技术和计算机应用的高速发展,人类进入了信息爆炸的时代。面对海量的信息,用户需要根据其信息需求投入大量的时间进行信息过滤和信息选择,产生了信息过载问题^[1]。推荐系统是解决这个问题行之有效的方法,并且在这些年发展迅速。推荐系统根据用户的信息需求,为其提供个性化的推荐,在海

量的信息空间中,为用户推荐有用的信息^[2]。目前推荐系统主要包括基于协同过滤的推荐^[3-4],基于内容的推荐技术^[5],基于网络的推荐技术^[6]和混合推荐技术^[7-9]四类。目前应用广泛且颇受欢迎的是协同过滤推荐算法,在物品推荐的过程中只依赖推荐的社交性而不需要关注物品的信息就可以实现推荐。

协同过滤推荐算法通常根据用户行为计算每一对对象的相似度度量,并向相似的用户推荐一个类似的项目。例如,如果用户 A 与用户 B 相似,那么两个用

户在项目上做了相同的评价,而用户 B 喜欢项目 C,但是用户 A 还没有对项目 C 进行评价,那么此算法可以假定用户 A 喜欢项目 C,并将其推荐给用户 A。该过程的核心步骤包括计算对象相似度,根据相似度选择对象的邻居,并根据邻居的评级来预测评级。

传统上,网络科学一直专注于静态网络来探索链接预测。也就是说,在计算过程中,类似的方法会将节点之间的连接持久化。然而,越来越多的人认识到,大多数推荐系统最好被描述为时间网络,其中的链接只是间歇性的或有一定的重量衰减。另外,在推荐系统中节点的空间分布很少是科学家考虑的因素,因为拓扑结构不保持任何空间信息或位置约束,并影响相似计算的结果。很多学者针对时序方面提出了一系列的方案。文献[10]结合物品信息和情境上下文信息,发现情境上下文在提高推荐质量上有明显作用。文献[11]提出结合上下文信息得到的结果更容易获得用户信任。文献[12]提出时间物品提前过滤的方法,也就是说选择与目标时间相同的评分信息当作训练集去训练推荐模型的算法。文献[13]提出结合用户行为的时间信息对于提高推荐质量有明显作用。文献[14]提出的即时推荐系统能够提供量身定制的推荐列表,充分满足用户的需求,帮助他们获得有用的信息以及大数据空间里感兴趣的资源。但以上算法主要研究了用户行为时间特性,而忽略了用户所处的空间对于推荐的影响,所以本文利用用户行为的时间特性和用户空间因素,采用结合评分时间和用户空间的方法对推荐算法进行研究。

在现实生活中,用户的兴趣会随着时间的变化而改变,用户最近的评分相比用户以前的评分更能体现用户的兴趣点。所以在进行用户相似性计算时,对用户评分进行时间衰减,将原来单一评分改为相对时间的评分来降低历史评分对推荐的影响力,相应地增加最近评分对推荐的影响力。这样用户评分的时间特性对推荐的作用得以体现。

对于空间,假设用户在这样的推荐系统中构建一个隐藏的社交网络,每两个用户之间的连接就会推断出他们的品味或者友谊,以及用户在一个空间中的分布情况。在每个用户的空间约束下,可以通过使用用户之间的距离来选择邻居,而不是通过相似性。本文把用户抽象成网络节点,用户之间的相似度作为用户之间连接的边权重,利用复杂网络拓扑布局算法^[15]建立用户空间网络布局。通过用户空间布局计算所有用户之间的空间距离,根据用户空间距离选取一定数量的邻居用户,最终计算用户对未评分物品的预测评分值。

1 研究方法

1.1 协同过滤推荐算法

协同过滤推荐算法一般分为两种类型:一种是基于用户的协同推荐(UBCF),另一种是基于项目的协同推荐(IBCF)^[3-4]。

UBCF 算法从用户角度出发,计算用户相似度,从而确定目标用户的邻居用户,为目标用户进行项目推荐。

IBCF 算法从项目角度出发,计算项目相似度,从而确定该项目的相似项目,为用户进行项目推荐。

UBCF 算法主要通过以下 5 个步骤实现物品的推荐:

- (1) 初始化用户行为信息,建立用户评分矩阵;
- (2) 通过相似度公式计算用户相似度;
- (3) 根据用户相似度结果,选取目标用户的邻居用户;
- (4) 求出用户对物品的预测评分,生成评分预测矩阵;
- (5) 根据评分预测矩阵,选择前 N 个项目组成推荐物品集合进行 Top-N 推荐。

推荐系统中用户集合为 $U = \{u_i | 1 \leq i \leq m\}$,表示有 m 个用户,物品集合为 $O = \{o_j | 1 \leq j \leq n\}$,表示有 n 个物品。用户的评分矩阵集合为 $R = \{r_{ia}\} \in R^{m \times n}$,其中 r_{ia} 表示用户 i 对物品 α 的评分值,评分值的差异反映了用户对物品兴趣度。另外设置一个与评分矩阵集合对应的集合 $A = \{a_{ia}\} \in R^{m \times n}$,若用户 i 对物品 α 评过评分,则 a_{ia} 为 1,否则为 0。表 1 即为 UBCF 算法的用户-物品评分矩阵模型。虽然表 1 显示了用户对物品的行为信息,但用户行为的时间特性并没有表现出来。

表 1 用户-物品评分矩阵

用户-物品	o_1	...	o_j	...	o_n
u_1	r_{11}	...	r_{1j}	...	r_{1n}
\vdots	\vdots		\vdots		\vdots
u_i	u_{i1}	...	r_{ij}	...	r_{in}
\vdots	\vdots		\vdots		\vdots
u_m	r_{m1}	...	r_{mj}	...	r_{mn}

UBCF 算法通过用户相似度确定用户的邻居用户。本文采用基于用户的皮尔逊相似度计算公式计算用户相似度^[16],公式如下:

$$S(i,j) = \frac{\sum_{\alpha=1}^n (r_{i\alpha} - \bar{r}_i)(r_{j\alpha} - \bar{r}_j)}{\sqrt{\sum_{\alpha=1}^n (r_{i\alpha} - \bar{r}_i)^2} \sqrt{\sum_{\alpha=1}^n (r_{j\alpha} - \bar{r}_j)^2}} \quad (1)$$

式中: $r_{i\alpha}$ 为用户 i 对物品 α 的评分值; $r_{j\alpha}$ 为用户 j 对物品 α 的评分值; \bar{r}_i 为用户 i 对所有物品的选择性概率评分值的平均值; \bar{r}_j 为用户 j 对所有物品的选择性概率评分值的平均值; n 为用户的最近邻个数。

UBCF 算法选择相似度最高的前 m 个用户作为其邻居用户,构成邻居用户集合 T_i 。由于已知用户相似度以及邻居用户的评分信息,利用式(2)对目标用户 u_i 的所有未评分物品 o_j 进行评分预测。

$$r_{i\alpha}^U = \bar{r}_i + \frac{\sum_{j=1}^m [(r_{j\alpha} - \bar{r}_j) \times S_{ji}]}{\sum_{j=1}^m S_{ji}} \quad (2)$$

1.2 基于时间和空间的相似算法

考虑到一个包含 n 个节点和 m 条边的复杂网络 G ,我们认为网络 G 分布在一个空间区域内。其中 t 表示 G 的时间属性,而且节点代表推荐系统中的用户,边代表用户之间的连接,比如用户之间的相似性。

因此,对于任何节点,都有一个位置属性推断它们的空间位置,可以计算节点之间的距离。节点 n_{ij}^t 表示随着时间 t 的变化,节点 i 和 j 的位置以及距离也发生变化。 l_e 表示两个节点连接边的权重,类似地,权重也会随着时间的变化而变化。基于此,有一个包括项目和连接的复杂网络。

基于时空的推荐算法由以下规则组成:

规则 1 初始化用户之间的连接。

规则 2 初始化用户之间的位置。

规则 3 用户对物品的评分改变了它们的连接和位置。

规则 4 用户对物品的评分随着时间的变化而有一定程度的衰减。

规则 5 对于用户的评分预测,其对项目的评分取决于项目的平均评分和用户最近邻居的评分。

把用户集合 U 中所有用户抽象成网络节点,记用户节点数 n ,通过式(1)计算用户之间的相似度 $S(i,j)$,把用户之间的相似度作为用户节点之间连接的边权重,记用户之间的连接边为 m 。利用复杂网络拓扑布局算法建立用户之间的网络布局。由于数据量庞大,本文对用户网络布局中的边进行筛选,公式如下:

$$k = \frac{m}{n} \quad (3)$$

以 $k=1$ 为例,用户空间网络布局如图 1 所示。



图 1 $k=1$ 时用户网络布局

在用户空间网络布局中,对于每一个用户 u_i 都会生成一个坐标 $(x_{u_i}, y_{u_i}, z_{u_i})$,因此,利用欧几里得距离公式可以计算出相关联用户之间的空间距离,公式如下:

$$D(u_i, v_i) = \sqrt{(x_{u_i} - x_{v_i})^2 + (y_{u_i} - y_{v_i})^2 + (z_{u_i} - z_{v_i})^2} \quad (4)$$

对用户空间距离升序排列,用户之间距离越小,说明用户之间相关程度越高。取出与目标用户距离最近的前 m 个用户作为其邻居用户集合 TD_i 。

例如,有 10 个用户,用户编号为 1 到 10,其中用户 1 为目标用户,其余 9 个用户为用户 1 的相似用户,取用户 1 的 5 个邻居用户。用户相似度如表 2 所示。

表 2 用户 1 与其余用户相似度

用户	用户	相似度
1	1	0
1	2	1.076 664 050 186 200
1	3	1.419 665 518 254 770
1	4	1.763 413 844 038 070
1	5	2.843 772 616 195 180
1	6	2.017 048 150 166 410
1	7	2.981 001 083 719 920
1	8	2.690 164 437 127 400
1	9	0.148 083 160 198 254
1	10	0

本文对于邻居用户的选择是基于用户空间距离,有别于传统的基于用户相似度选取邻居用户的规则。用户 1 的空间布局如图 2 所示。

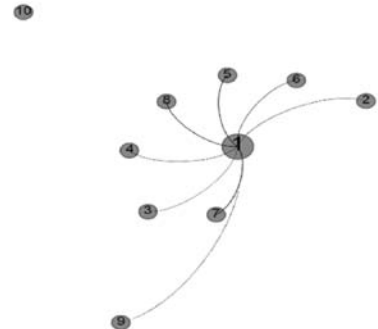


图 2 用户 1 的空间布局

图2清晰地展示了用户1与其余用户的拓扑结构图。用户之间的空间距离也能直观地显示在图中,并且距离越近的邻居,与用户1的连边越清晰。

本文使用拓扑布局算法,为空间中每个用户生成空间坐标。利用式(4),计算用户1与其他用户的空间距离。需要说明的是,对于没有连边的用户,即使也有相应的空间坐标,但是默认与用户1的空间距离无限大。用户1与其他用户空间距离如表3所示。对表中与用户1的空间距离进行升序排序,由于取用户1的前5个邻居,故取与用户1距离最小的前5个用户,分别为7,5,8,6,4。

表3 用户1与其余用户的空间距离

用户	用户	空间距离
1	1	0
1	2	49.980 232 934 944 1
1	3	43.153 685 997 527 5
1	4	38.644 758 286 472 4
1	5	31.912 296 304 753 7
1	6	36.124 104 960 663 4
1	7	31.333 672 822 257 9
1	8	32.312 529 698 549 1
1	9	88.800 554 265 455 0
1	10	∞

对于用户评分的时间效应,本文引入时间衰减函数,公式如下:

$$w(t, t(r_{i\alpha})) = \frac{1}{1 + \gamma |t - t(r_{i\alpha})|} \quad (5)$$

式中: t 为推荐时间; $t(r_{i\alpha})$ 为用户 i 对物品 α 的评分的时间信息; γ 为时间衰减因子,取值为 $[0, 1)$ 之间的小数。当衰减因子 γ 递增时,衰减函数 $w(t, t(r_{i\alpha}))$ 呈递减趋势,当 γ 越大时,说明时间的影响越大,可以动态的调整 γ 值来优化推荐结果。当衰减因子 $\gamma = 0$ 时,公式没有衰减,即不考虑时间因素,仅考虑用户空间因素对推荐结果的影响,这种情况需要与衰减后的情况进行对比。

式(5)按照历史评分产生的时间对评分的重要性进行了衰减,生成用户-物品的衰减评分值 $RT(u_i, o_j)$ 。

1.3 算法基本思想

根据以上描述,本文的算法思想如下:

输入: <用户(user), 物品(item), 用户行为(rating), 时间戳(TimeStamp) >。

输出: 预测评分值矩阵

步骤1 根据用户对物品的评分信息建立“用户-物品-评分-时间”模型。

步骤2 利用式(1)计算用户相似度 $S(i, j)$, 利用复杂网络拓扑布局算法建立用户空间网络布局, 并且为每一个用户生成空间坐标 (x_{ui}, y_{ui}, z_{ui}) 。

步骤3 利用式(4)计算所有用户之间的空间距离 $D(u_i, v_i)$, 然后选取前 m 个用户作为其邻居用户集合 TD_i 。

步骤4 在用户-物品-评分-时间模型中, 利用式(5), 计算用户-物品的衰减评分值 $RT(u_i, o_j)$, 再利用式(1)计算评分衰减之后的用户相似度 $ST(i, j)$ 。

步骤5 由以上步骤得出的 TD_i 和 $RT(u_i, o_j)$ 以及 $ST(i, j)$, 结合使用式(2)计算出用户对物品的预测评分值 $r_{i\alpha}^U$, 最终生成预测评分值矩阵。

2 实验结果及分析

2.1 实验数据

在数据集选取方面, MovieLens数据集不仅记录了943个用户对1682部电影的100000条评分^[17], 而且记录了用户行为的时间戳信息, 所以可以作为实验的数据集。

实验将数据集划分为训练数据集和测试数据集, 其中训练数据集占80%, 测试数据集占20%。这样有利于减少偶然因素带来的不利影响, 保证实验的准确性。为了检验结合评分时间和用户空间的协同过滤推荐算法(TS)的有效性, 本文用基于皮尔逊用户协同过滤算法(UCF-Pearson-Opt)和基于皮尔逊物品协同过滤算法(ICF-Pearson-Opt)作为对比^[18]。

在实验结果对比中, 邻居个数选择从10开始, 直到150, 每次增加10。需要说明的是, 邻居个数为5的对比结果也一并列出。

2.2 评价指标与结果及分析

推荐算法一般使用平均绝对误差MAE(Mean Absolute Error)和均方根误差RMSE(Root Mean Square Error)作为算法准确度的评价标准^[19]。假设 $r_{i\alpha}$ 表示用户 i 对物品 α 的真实评分, $r_{i\alpha}^*$ 表示融入时序和空间因素的推荐算法计算出的用户 i 对物品 α 的预测评分, T 表示测试集, 则平均绝对误差和均方根误差公式分别为:

$$MAE = \frac{\sum_{r_{i\alpha} \in T} |r_{i\alpha} - r_{i\alpha}^*|}{|T|} \quad (6)$$

$$RMSE = \sqrt{\frac{\sum_{r_{i\alpha} \in T} |r_{i\alpha} - r_{i\alpha}^*|^2}{|T|}} \quad (7)$$

实验使用MAE值和RMSE值度量推荐算法的准确度,

MAE 值和 RMSE 值越小,说明准确度越高。

本文提出的算法中,衰减系数 γ 是至关重要的参数。因此,首先通过实验研究了衰减系数 γ 对推荐结果的影响,选取最优的衰减系数。评价指标结果如图 3 和图 4 所示。

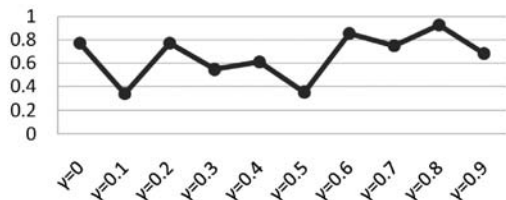


图 3 算法的 MAE 值随着的变化

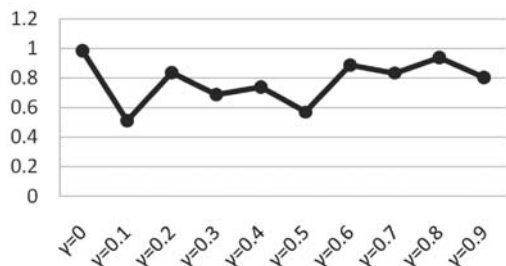


图 4 算法的 RMSE 值随着的变化

在图 3 和图 4 中,横坐标均代表衰减因子 γ ,纵坐标分别代表 MAE 值和 RMSE 值。可以看出,当衰减因子从 0 增加到 0.9 时,MAE 值和 RMSE 值波动比较大。当衰减系数 $\gamma = 0.1$ 时,MAE 值和 RMSE 值均取最小值。

为了比较不同算法的预测效果,计算 UCF-Pearson-Opt 和 ICF-Pearson-Opt 与本文提出的 TS 在不同邻居数目的平均绝对误差和均方根误差。需要说明的是,当衰减系数 $\gamma = 0$ 时,即没有考虑时间因素,仅考虑在用户网络布局中,空间距离因素对于推荐算法结果的影响。结果分别如图 5 和图 6 所示。

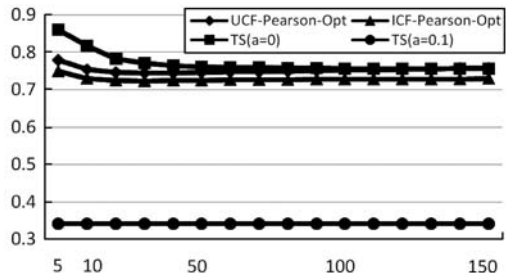


图 5 三种算法的 MAE 值,TS 为本文算法

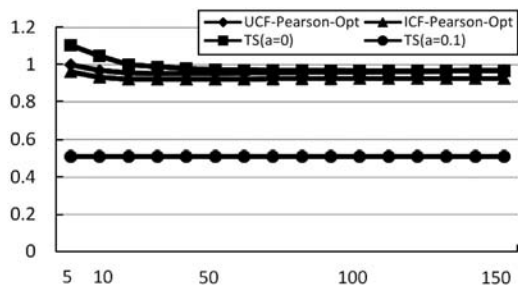


图 6 三种算法的 RMSE 值,TS 为本文算法

在图 5 和图 6 中,横坐标均代表目标用户的邻居用户数目,纵坐标分别代表 MAE 值和 RMSE 值。可以看出,本文所使用的 TS 算法在衰减因子 $\gamma = 0.1$ 时效果最好,在目标用户邻居数目为 5 时 MAE 值最低,为 0.340 102 736;当邻居数目为 80 时 RMSE 最低,为 0.508 920 893,并且随着用户相似邻居数目的增长,曲线呈水平增长,说明 TS 算法的稳定性较好。

TS 算法在衰减系数 $\gamma = 0.1$ 时的误差比 ICF-Pearson-Opt 算法更低,MAE 值提高了 38%,RMSE 值提高了 41%,说明了本文算法的有效性。当衰减因子 γ 时,随着邻居数目的增加,MAE 值和 RMSE 值由远远高于 UCF-Pearson-Opt 和 ICF-Pearson-Opt 算法逐渐归为持平状态,说明在仅考虑用户空间因素方面还有待改进和优化。

3 结 语

结合三种算法的结果可知,TS 算法在衰减系数 $\gamma = 0$ 时,推荐效果并不显著。当衰减系数 $\gamma = 0.1$ 时效果最好,相比 ICF-Pearson-Opt 算法,MAE 值提高了 38%,RMSE 值提高了 41%。综上所述,在 MovieLens 数据集上,通过实验证明,与传统的协同过滤算法比较,本文所用算法的准确率有显著提高,并且算法的时间复杂度较低。

参 考 文 献

- [1] Resnick P, Iacovou N, et al. GroupLens: an open architecture for collaborative filtering of netnews [C]//Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, Chapel Hill, North Carolina, United States, October 22-26.
- [2] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6): 734-749.
- [3] Linden G, Smith B, York J. Amazon.com recommendations: Item-to-Item collaborative filtering [J]. IEEE Internet Computing, 2003, 7(1): 76-80.
- [4] Huang Z, Zeng D, Chen H C. A comparison of collaborative-filtering recommendation algorithms for e-commerce [J]. IEEE Intelligent Systems, 2007, 22(5): 68-78.
- [5] Pazzani M J, Billsus D. 10 Content-based recommendation systems [M]//The Adaptive Web. Springer-Verlag, 2013:

325 - 341.

- [6] Zhang Y C, Medo M, Ren J, et al. Recommendation model based on opinion diffusion[J]. Epl, 2007, 80(6) :417 - 429.
- [7] 刘兆兴, 张宁, 李季明. 基于协同过滤和网络结构的个性化推荐算法[J]. 复杂系统和复杂性科学, 2011, 8(2) :29 - 33.
- [8] 赵超超. 基于用户和基于项目结合的个性化推荐算法[J]. 内蒙古农业大学学报(社会科学版), 2007, 6(9) :139 - 142.
- [9] Bogers T, Antal V D B. Recommending scientific articles using citeulike[C]//ACM Conference on Recommender Systems. ACM, 2008 :287 - 290.
- [10] Adomavicius G. Incorporating contextual information in recommender systems using a multidimensional approach[J]. ACM Transactions on Information Systems, 2005, 23(1) :103 - 145.
- [11] Gorgoglione M, Panniello M, Tuzhilin A. The effect of context_aware recommendations on customer purchasing behavior and trust[C]//ACM Conference on Recommender Systems, 2011 :85 - 92.
- [12] Baltrunas L, Amatriain X. Towards time-dependant recommendation based on implicit feedback[C]//Workshop on Context-aware Recommenders Systems (CARS'09), 2009 :1 - 5.
- [13] Panniello U, Tuzhilin A, Palmisano C, et al. Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems[C]//ACM Conference on Recommender Systems. ACM, 2009 :265 - 268.
- [14] Akermi I, Boughanem M, Faiz R. Just-In-Time recommendation approach within a mobile context[C]//IEEE/WIC/ACM International Conference on Web Intelligence. IEEE, 2017 :636 - 639.
- [15] Jacomy M, Venturini T, Heymann S, et al. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software[J]. Plos One, 2016, 9(6) :e98679.
- [16] 吴月萍, 郑建国. 改进相似性度量方法的协同过滤推荐算法[J]. 计算机应用与软件, 2011, 28(10) :7 - 8, 42.
- [17] 赵向宇. Top-N 协同过滤推荐技术研究[D]. 北京:北京理工大学, 2014.
- [18] 刘群, 陈阳, 易佳. 一种改进的基于物质扩散理论的Item-based 协同过滤算法[J]. 数字通信, 2013, 40(2) :11 - 14.
- [19] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[J]. Uncertainty in Artificial Intelligence, 2013, 98(7) :43 - 52.

(上接第 162 页)

5 结 语

本次针对移动智能终端,基于 Android 的互联网加密传输功能设计与实现,达到安全通信的目的,实现了限制非工作应用、节省通信流量的效果,解决了企事业单位移动平台终端的实际使用问题。

本次功能设计与实现是移动操作系统定制开发的实践,定制的移动智能操作系统功能,例如信息安全、权限管控、统一 UI、续航能力等,更符合业务实际使用需求,有广泛需求前景。同时,在系统体系中,平台“基础”决定上层应用“建筑”。对移动操作系统的深度优化,可以作为移动应用生态环境基础,例如规范应用 app 开发、移动安全通信协议等,形成企事业移动应用体系的基础规范,占得行业先机。

参 考 文 献

- [1] 董志超, 王柄璇, 倪光玉, 等. Linux 下 Ping 命令在内核中动态运行过程的跟踪与分析[J]. 计算机应用与软件, 2015, 32(7) :102 - 104, 154.
- [2] 王天伟. 一种 Linux 系统下的线程通信方法[J]. 计算机应用与软件, 2017, 34(10) :330 - 333.
- [3] 陈志锋, 李清宝, 张平, 等. 面向 Linux 的内核级代码复用攻击检测技术[J]. 软件学报, 2017, 28(7) :1732 - 1745.
- [4] 邵冬哲, 张庭, 刘斌. 软件定义网络中 TCP 伪拥塞问题探究[J]. 计算机研究与发展, 2017, 54(1) :192 - 201.
- [5] 李宇成, 梁宗希. Linux 摄像头驱动的设计优化及其对应的 Android 下 HAL 封装设计方法探究[J]. 计算机应用与软件, 2016, 33(9) :249 - 253.
- [6] 傅强. 浅析安卓开发的相关技术[J]. 计算机光盘软件与应用, 2014, 40(14) :292 - 292.
- [7] 杨潇亮. 基于安卓操作系统的应用软件开发[J]. 电子制作, 2014(19) :45 - 46.
- [8] 吴文思. 基于安卓的系统架构及程序开发探索[J]. 电子技术与软件工程制作, 2014(19) :76 - 76.
- [9] 张玉清, 王凯, 杨欢. Android 安全综述[J]. 电子技术与软件工程制作, 2014, 51(7) :1385 - 1396.
- [10] 罗升阳. Android 系统源代码情景分析[M]. 北京:电子工业出版社, 2012.
- [11] 郭霖霖. 第一行代码 Android[M]. 北京:人民邮电出版社, 2014.
- [12] 陈联柯, 邹仕洪. 基于 Android 平台的快速隐私清理的设计和实现[EB]. 中国科技论文在线.
- [13] 刘锐. Android 开发的性能优化[J]. 电脑编程技巧与维护, 2013(16) :115 - 116.