

基于层次化上下文因式分解机的推荐系统

秦大路, 李晓宇

(郑州大学 信息工程学院, 郑州 450001)

摘 要:在基于协同过滤的推荐系统中,因式分解机模型是基于矩阵分解的一般化模型,不需要特定支持向量,可直接应用于回归和分类中,并能更准确地处理稀疏矩阵.通过对其进行改进,在不提高时间复杂度的同时考虑上下文环境,并对上下文进行层次化处理.通过两组真实数据集,在不同的指标下进行实验.最后证实改进后的模型,在准确率和学习速率上优于原有模型.

关键词:推荐系统;协同过滤;矩阵分解;上下文环境;因式分解机

中图分类号:TP273

文献标志码:A

当今互联网,推荐系统(Recommender Systems)是在线媒体、电子商务等领域的研究热点之一.推荐系统主要分为基于内容(content-based)推荐系统、协同过滤(collaborative filtering)和混合(hybrid approaches)推荐系统三大类^[1].与基于内容推荐算法相比较,协同过滤算法不需要对内容或者用户本身进行解析,而是根据其历史行为数据提供个性化推荐,在过去十几年中取得了丰富的研究成果^[2].其中,基于矩阵分解(Matrix Factorization)的潜在因子模型(latent factor model)在预测的精准性和稳定性方面得到最为广泛的认可^[3].

在推荐算法中,假设评分矩阵 $R \in \mathbf{R}^{M \times N}$ (M 代表用户数, N 代表项目数) 通过潜在因子映射可降维映射表示为 $r_{ui} = q_i^T p_u$, 其中 $p_u, q_i \in \mathbf{R}^f$, 且 f 远远小于 $\min(M, N)$. 在此基础上, Koren 经过添加偏量, 考虑时间、空间影响因子对方程式进行了进一步扩展^[4]; Steffen 提出了因式分解机模型(Factorization Machines, FM). FM 模型的优点主要有: (1) FM 模型可更有效处理稀疏矩阵(SVM 难以处理之处); (2) FM 模型是线性复杂度, 可直接被优化不需要支持向量; (3) FM 模型是一般化模型, 通过对特征向量的定制可以满足其他模型^[5]. 基于以上优点, 将其应用在推荐系统中, 通过改进并提出基于层次化上下文的 FM 模型. 实验证明, 改进后的模型, 准确率有了一定提升.

1 背景陈述

1.1 传统的推荐系统

传统推荐系统可分为基于内容推荐、协同过滤、混合推荐, 其中协同过滤受到的认可最为广泛, 并且传统推荐可以表示为二元关系 $\langle u, i \rangle$. 传统推荐任务可以形式化定义为一个线性回归任务, 即 $U = \{\mu_1, \mu_2, \dots\}$, $I = \{i_1, i_2, \dots\}$ 预测函数 $y: U \times I \rightarrow \mathbf{R}$. 且 $y(u, i)$ 代表用户 U 对商品 I 的评分, 我们从数据集中选取训练集片段 S 且 $S \in U \times I$, 已知训练集中所有 $y(u, i)$ 数值. 此任务的最终目的是得到预测函数 y , 可表示任意一个用户与商品的函数关系.

1.2 上下文感知推荐

然而, 在很多个性化推荐中, 除考虑用户与商品之间的关系外, 上下文信息也是影响用户选择的重要因素. 例如, 旅游系统的推荐中, 系统对旅游景点的推荐在夏天和冬天明显不同; 同样, 对于在线内容提供商(电子商务、视频), 用户在何时、何地需求服务也是重要的考虑项.

收稿日期: 2014-10-17

基金项目: 国家自然科学基金(61073023)

作者简介: 秦大路(1990—), 男, 河南新乡人, 郑州大学硕士研究生, 研究方向为移动计算, E-mail: dalu610@163.com.

Tuzhilin 等人在上下文感知推荐的研究中已证明,融入上下文信息可以更好提高推荐精度,并且提出“上下文感知推荐系统(context-aware recommender systems, CARS)”的概念^[6]. Tuzhilin 将上下文环境定义为变量 c 且 $c \in C$, 其中 C 可代表时间、地点、伙伴, c 可代表具体的上下文环境,如周末、工作日、情侣、朋友等. 同理,此时预测函数 y 可以形式化表示为: $U \times I \times C_3 \times \dots \times C_m \rightarrow \mathbf{R}$.

1.3 矩阵分解

用户的评分数据可以表示为评分矩阵 R , 矩阵的每个数值代表用户 U 对商品 I 的评分,但是矩阵中有很多空的元素,我们称这些空的元素为缺失值(missing value). 因此,评分预测也可以表示为矩阵填空,最早的矩阵分解是 svd(奇异值分解),但这种方法,不仅需要补全庞大的稀疏评分矩阵、占据大量存储空间,而且时间复杂度也过高.

Funk-SVD 方法则将评分矩阵表示为 $r_{ui} = \sum q_i^T p_u$, 在均方根误差(Root-mean-square error, RMSE)的指标下,直接设置损失函数 $RLS-Opt = \sum_{(x,y) \in S} (y(x) - y)^2 + \sum_{\theta \in \Theta} \lambda_{(\theta)} \theta^2$ 添加正则参数防止过拟合,通过梯度下降方法直接最小化损失函数快速获得参数. 后续模型通常在其基础上通过添加偏置项和其他因子,例如 Koren 提出的 svb++ 模型, Steffen 通过对 SVM(支持向量机)和矩阵分解的改进,提出因式分解机模型,不仅能更好处理稀疏矩阵,其本身时间复杂度也是线性的.

2 上下文层次化 FM 模型

2.1 传统 FM 模型介绍

Koren 提出可以将评分矩阵分解地表示为 $r_{ui} = q_i^T p_u$, 并通过考虑时间空间和一些偏量因数,将评分矩阵进一步扩展表示为 $r_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T p_u(t)$. FM 模型在此基础上,通过一般化处理,将评分矩阵进一步分解. 存储所要计算的设计矩阵中,用 $\{1, 0\}$ 代表是否出现此变量. 特别是在稀疏矩阵中,存储压力得到减轻,推荐的精度也有所提升.

因此,FM 模型的方程式在 2 维的情况下可表示为:

$$\hat{y}(x) := w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j. \quad (1)$$

在公式(1)中, x_i 代表训练集中一个特征,其值只为 $\{0, 1\}$, 表示此条数据是否含有此特征; $w_0 \in \mathbf{R}$, $w \in \mathbf{R}^n$, $V \in \mathbf{R}^{n \times k}$ 是经过机器学习方法得到的参数,且 $\langle v_i, v_j \rangle := \sum_{f=1}^k v_{i,f} \cdot v_{j,f}$ ($k \ll n$), 一个行向量 v_i 则代表在 V 矩阵中第 i 个拥有 k 个特征的向量. k 是一个正整数的假设值,代表特征向量矩阵的维度. w_0 代表全局偏量, w_i 代表第 i 个特征变量的权值; 评分矩阵 $\hat{w}_{i,j} := \langle v_i, v_j \rangle$. 通过对评分矩阵的进一步分解,可以更适合处理稀疏矩阵下的推荐. 公式(1)通过一定的变换可以转化为:

$$\begin{aligned} \sum_{i=1}^n \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \langle v_i, v_j \rangle x_i x_j - \frac{1}{2} \sum_{i=1}^n \langle v_i, v_i \rangle x_i x_i = \\ &= \frac{1}{2} \sum_{f=1}^k \left(\left(\sum_{i=1}^n v_{i,f} x_i \right)^2 - \sum_{i=1}^n v_{i,f}^2 x_i^2 \right), \\ \hat{y}(x) &:= w_0 + \sum_{i=1}^n w_i x_i + \frac{1}{2} \sum_{f=1}^k \left(\left(\sum_{i=1}^n v_{i,f} x_i \right)^2 - \sum_{i=1}^n v_{i,f}^2 x_i^2 \right). \end{aligned} \quad (2)$$

经过转换后,公式(2)运算的复杂度变为线性复杂度,简单表示为 $O(k^n)$, 取决于对评分矩阵 $\hat{w}_{i,j}$ 的降维分解矩阵的维数 k , 以及每条评分数据所包含的变量. 由于各条评分数据往往只包含用户和商品两个变量,时间复杂度通常表示为 $O(k^2)$.

2.2 FM 模型改进

传统 FM 模型作为一种一般化模型,不需要特定的数据输入便可以实现所有分类和线性回归任务. 然而在实际应用中,对具体的任务进行相应修改和定制明显优于一般化模型. 在此,对特征向量进行相应设计,使其更恰当地应用于电影推荐中,在考虑上下文的同时,对上下文进行层次化处理,更精准地提供个性化推荐.

同时,排除不相关特征的干扰,使计算更高效,最后通过快速学习得到所需参数。

2.2.1 上下文层次化处理 上下文感知推荐系统对上下文的处理大致可分为 3 种情况:上下文预过滤(Contextual Pre-Filtering)、上下文后过滤^[7](Contextual Post-Filtering)和上下文建模(Contextual Modeling)。结合上下文预过滤和上下文建模两种方法,对上下文感知推荐系统进行处理。

在普通上下文环境下的推荐中,用户的评分数据可以转化为 $U \times I \times C_1 \times C_2 \times \dots \times C_n \rightarrow \mathbf{R}$,其中 U 代表用户, I 代表商品, C 代表各种上下文环境。本文以时间上下文信息例,传统处理上下文时间方法,一般采用天数为特征,或采用一天的几个时间段为分析。而在实际的应用中,数据集常常包括几个月、甚至几年的评分内容,简单地使用天数显然不合理。

在上下文信息处理方面,将上下文信息定义为一个集合 D ,且 $D_i = (D^1, D^2, \dots, D^q)$,集合 D 是一种层次化的结构,可以代表任意一种特殊上下文环境。如图 1,以时间、地点上下文信息为例表示 D 的层次关系。 D^q 代表层级更为精细,而 D^1 往往是较为粗糙层级。那么原有的上下文推荐 $U \times I \times C_3 \times \dots \times C_m \rightarrow \mathbf{R}$ 转换为: $U \times I \times (D_1^1 \times D_1^2 \times \dots \times D_1^q) \times (D_2^1 \times \dots) \rightarrow \mathbf{R}$,同时 HFM 的方程式可表示为:

$$\begin{aligned} \hat{y}(x) &:= w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j + g(x), \\ g(x) &= \sum_{m=1}^q w_m x_m + \sum_{i=1}^n \sum_{m=1}^q \langle v_i, v_m \rangle x_i x_m. \end{aligned} \tag{3}$$

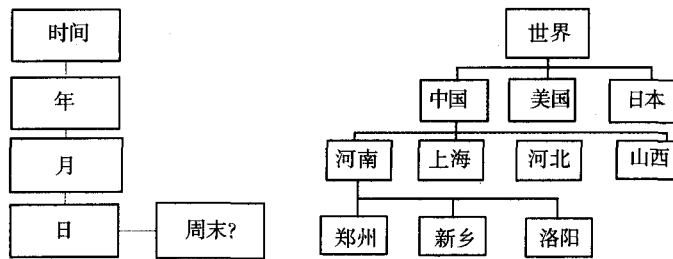


图1 上下文信息从层次化显示

在公式(3)中,添加新的权值 w_m 表示上下文环境的权值,添加 $\langle x_i, x_m \rangle$ 表示在上下文环境影响下新的相互关系。在考虑上下文环境同时,对上下文信息进行进一步分解,增强数据集各个评分数据的关系,更有利于对参数的学习,并且不影响方程式的时间复杂度^[8]。

2.2.2 模型快速学习 由于方程式的线性复杂度,在回归任务中,最广泛的损失函数是最小二乘函数,为防止数据的过拟合,通常添加一个规则变量通常为 L_2 ,即: $RLS - Opt = \sum_{(x,y) \in S} (\hat{y}(x) - y)^2 + \sum_{\theta \in \Theta} \lambda_{(\theta)} \theta^2$ 。

本文使用交替最小二乘法(Alternating Least Square, ALS)学习方法,通过训练集学习,多次迭代后得到合适的变量值。并且 ALS 方法优于 SGD 方法,不需要设置和调整学习速率,从而达到快速收敛^[9]。

由于方程式对于每一个变量 $\theta(w_0, w, V)$ 都是线性的,因此表示为: $\hat{y}(x | \theta) = g_{(\theta)}(x) + \theta h_{(\theta)}(x)$ 。又因: $\frac{\partial}{\partial \theta} RLS - OPT = \sum_{(x,y) \in S} (\hat{y}(x) - y) h_{(\theta)}(x) + \lambda_{(\theta)} \theta = 0$, 其中 $\hat{y}(x) - y = g_{(\theta)}(x) + \theta h_{(\theta)}(x) - y$, 即

$$(g_{(\theta)}(x) - y) h_{(\theta)}(x) + \theta h_{(\theta)}(x)^2 + \lambda_{(\theta)} \theta = 0, \text{ 从而可以计算出: } \theta = - \frac{\sum_{(x,y) \in S} (g_{(\theta)}(x) - y) h_{(\theta)}(x)}{\sum_{(x,y) \in S} h_{(\theta)}^2(x) + \lambda_{(\theta)}}$$

以 w_l 为例: $\hat{y}(x | w_l) = w_l x_l + g_{(w_l)}(x)$, 即 $h_{(\theta)}(x) = x_l$, 即 $(g_{(\theta)}(x) - y) h_{(\theta)}(x) = (\hat{y} - w_l x_l - y) x_l$,

$$\text{即 } w_l \leftarrow - \frac{\sum_{(x,y) \in S} (\hat{y} - y - w_l x_l) x_l}{\sum_{(x,y) \in S} x_l^2 + \lambda(w_l)} \text{。同理可求得 } (w_0, V) \text{。}$$

3 实验结果

3.1 实验平台,评价指标,数据集介绍

本文采用 PC(Intel(R),CPU 3.21 GHz, RAM 4 GB) 实验平台和 Windows XP SP3 操作系统,开发工具为 visual studio 2010,使用 C++ 语言编写,数据集的部分处理采用 python 处理.

根据文献[10]论证,采用均方根误差(Root-mean-square error, RMSE) 和平均绝对误差(mean absolute

$$\text{error, MAE}) \text{ 作为评价指标: } \text{RMSE} = \sqrt{\frac{\sum_{(x,y) \in S} (\hat{y} - y)^2}{n}}, \text{MAE} = \frac{|\hat{y} - y|}{n}.$$

本实验主要采用两个真实数据集:GroupLens 提供的 movielens 数据集(包含 943 用户对 1 682 部电影的 10^5 条评分信息)和 Netflix 大赛提供的数据集(此论文抽取 16 000 用户的随机 20 条评价、约 32 万条评分数据).取 80% 的数据为训练集,其余 20% 的数据为测试集.

3.2 实验效果

图 2 是 MovieLens 和 Netflix 两组数据集在 RMSE 指标(越低越好)评价下的比较(Iter 代表迭代次数), X 轴表示学习迭代次数, Y 轴表示当前 RMSE 值.从图 2 中可以看出, HFM 在准确率上优于原有的 FM 模型.图 3 则是两组数据集在 MAE 指标下, Y 轴表示 MAE 值,经过相同的迭代次数学习后,从图 3 中可以看出, HFM 明显优于 FM 模型.

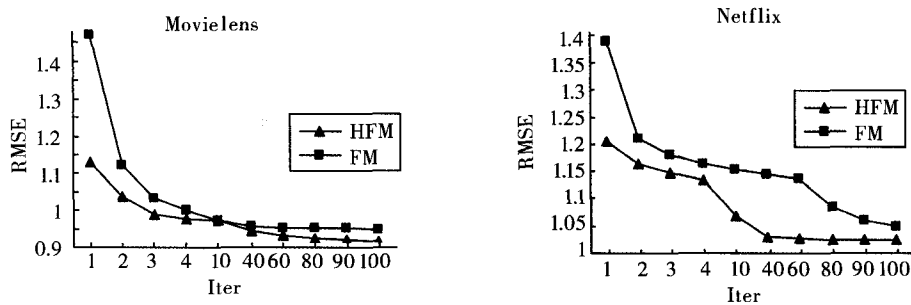


图2 RMSE比较

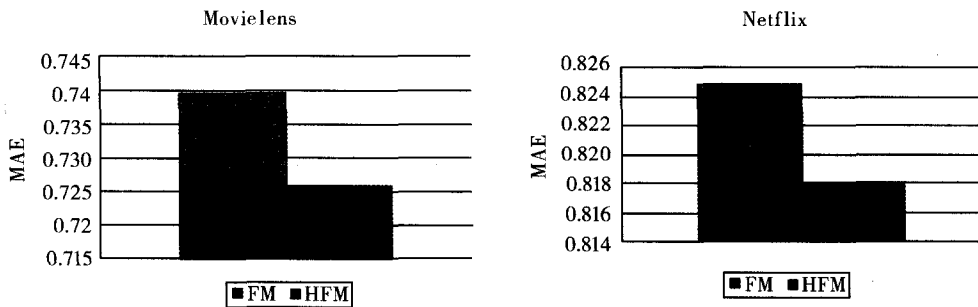


图3 MAE比较

4 结 语

本论文在原有 FM 模型的基础上,通过对上下文的进一步处理,层次化分割上下文,能更准确的体现评分数据之间的全局关系,并通过实验证明新的 HFM 模型的表现优于原有 FM 模型.除去对地点、时间较为显性的上下文环境进行层次化处理之外,对所有上下文变量做一般化的层次化模型处理是未来的研究重点.

参 考 文 献

- [1] Konstan J A, Riedl J. Recommender systems: from algorithms to user experience[J]. *User Modeling and User-Adapted Interaction*, 2012, 22(1/2):101-123.
- [2] 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, 2001.
- [3] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. *IEEE Computer*, 2009, 42(1):30-37.
- [4] Koren Y. Collaborative filtering with temporal dynamics[J]. *Communications of the Acm*, 2010, 53(4):89-97.
- [5] Rendle S. Factorization machines[C]. *10th IEEE International Conference on Data Mining (ICDM 2010)*, Sydney, 2010.
- [6] Adomavicius G, Tuzhilin A. *Context-aware recommender systems*[M]. New York: Springer, 2011.
- [7] Panniello U, Tuzhilin A, Gorgoglione M, et al. Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender system[C]. *RecSys*, New York, 2009.
- [8] Rendle S, L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation[C]. *WSDM*, New York, 2010.
- [9] Steffen Rendle, Zeno Gantner. Fast Context-aware Recommendations with Factorization Machines[C]. *SIGIR*, Beijing, 2011.
- [10] Panniello U, Tuzhilin A, Gorgoglione M. Comparing context-aware recommender systems in terms of accuracy and diversity[J]. *User Modeling and User-Adapted Interaction*, 2014, 24(1/2):35-65.

Recommendation System Research Based on Hierarchical Context Factorization Machines

QIN Dalu, LI Xiaoyu

(Information Engineering College, Zhengzhou University, Zhengzhou 450001, China)

Abstract: In the recommender systems based on collaborative filtering, the factorization machines model (FM) is a generalized model based on matrix factorization method which needs no specific support vectors. It can be applied in regression and classification and process sparse matrix exactly. By modifying FM it's possible to consider the context and implement hierarchical processing the context without improving the time complexity. Experiments on two group of real data have been done under different indices. It's proved that the modified model is better than the previous models in accuracy rate and learning speed.

Keywords: recommended system; collaborative filtering; matrix factorization; context; factorization machines