网络出版时间：2015-05-06 16:27
网络出版地址：http://www.cnki.net/kcms/detail/61.1450.TP.20150506.1627.019.html

# 一种基于 Spark 时效化协同过滤推荐算法

徐新瑞，孟彩霞，周雯，刘盈

（西安邮电大学 计算机学院,陕西 西安 710121）

**摘要**：针对传统的批量学习的基于模型的协同过滤算法对新用户（物品）更新缓慢，模型重训练成本高且扩展性不足，对噪音数据的处理有待提高，尤其是随着数据量的增长和时效性要求越来越高，挖掘其中的知识变得越来越困难等问题, 对置信权重在线协同过滤算法进行改进，引入自适应软边缘，提出二阶在线优化方法处理在线协同过滤中问题的新算法 SCWOCF(Soft Confidence Weighted Online Collaborative Filtering)，并在 Spark 流处理推荐框架下利用四组真实数据与相关算法作对比测试，实验结果表明，新算法能够及时处理用户（物品）的动态变化，并提升推荐的实时性和准确性, 降低计算成本，对噪声数据健壮性更强。

**关键词** 在线学习；自适应软边缘；软置信权重；二阶协同过滤；推荐系统；Hadoop；Spark on YARN

# A Spark-based Real-time Collaborative Filtering Recommendation Algorithm

XU Xin-rui, MENG Cai-xia, ZHOU Weng, LIU Ying

(School of Computer Science and Technology, Xi'an University of Posts&Telecommunications, Xi'an, 710121)

**Abstract:** Focusing on some drawbacks of traditional collaborative filtering algorithms based on batch learning, e.g., update slowly for new users or items, highly retraining cost and expand difficultly, and handle noise data need to be improved. Especially, it will be more and more difficult to mining knowledge from it with growing data and the requirement of real-time. In order to solve these problems, a new Algorithm named SCWOCF(Soft Confidence Weighted Online Collaborative Filtering) was proposed. In this algorithm, the adaptive soft margin was added to Confidence Weighted Online Collaborative Filtering and the second order online optimization methodology was used to solve online collaborative filtering problems. Finally, several experiments with four real-world datasets was conducted compared with some similar algorithms on the Spark stream processing recommendation framework. The results show that new algorithm can timely handle dynamic change of users and items, promote real-time and accuracy of recommendation, reduce cost of computation, increase robustness to noise data.

**Key words**: online learning; adaptive soft margin; soft confidence weight; second order collaborative filtering; recommender system; Hadoop; Spark on YARN

## 0 引言

当今，互联网高速发展，由此带来的数据爆炸式增长日益严重，如何对海量数据及时高效的存取并挖掘出其中隐藏的知识一直是学术界和工业界研究的热点[1-3]。

推荐系统（Recommender System）由于能够根据用户的差异性而个性化的推荐信息，相比于传统的搜索引擎特色显著。一个优秀的推荐系统不仅能够准确的预测用户的偏好从而提升用户对产品的体验，同时也让企业受益颇丰，据 VentureBeat 统

**收稿日期**：年-月-日；**修回日期**：2014年08月13日。

**基金项目**：国家自然科学基金资助项目(61105064)；陕西省自然科学基金资助项目(2014JM8303)；陕西省教育厅专项科研计划资助项目(11JK0988)；西安邮电大学研究生创新基金项目(ZL2013-42)

**作者简介**：徐新瑞（1989—），男，硕士研究生，中国计算机学会会员（E200039411G），研究方向为推荐系统、大数据挖掘；孟彩霞，教授，研究方向为算法设计、数据库与数据挖掘；周雯，硕士研究生，研究方向为数据挖掘；刘盈，本科生，研究方向为数据挖掘。

计，Amazon 的推荐系统为其提供了 35%的商品销售额[4]。协同过滤（Collaborative Filtering,CF）作为构建真实世界中推荐系统最成功的技术之一，通过对已有的部分用户的偏好分析来预测其他未知用户的偏好并提供个性化推荐，已在包括 Amazon、Netflix、Hulu、eBay、淘宝、豆瓣等商业网站得到广泛应用，并不断获得改进，如 Wang 等人[5]提出的在线多任务学习的 OMTCF 算法有效的提高了推荐准确率；Koren 等人[6]提出的将时间信息加入用户（物品）特征中解决兴趣度随时间漂移问题的 TimeSVD++算法效果显著；Ling 等人[7]提出的 SGD-RMF/DA-RMF 算法，较好的解决了用户或物品的动态变化问题；Jamali 等人[8]提出的 TrustWalker 随机游走模型很好的处理了兴趣变化；Liu 等人[9]利用“用户-兴趣-物品”三层模式，经过个性化排名的用户兴趣扩展来强化协同过滤，提出面向物品的基于模型的协同过滤算法 iExpand, 在较小的计算成本下提升了准确率并很好的处理了过度特征化和冷启动问题等。

从过去的 KB、MB、GB 级数据到如今的 TB、PB、EB、ZB、YB 级数据, 日均新增数 TB 数据已是不少商业网站面临的现实问题，怎样构建一套高可靠性优异的大数据挖掘和推荐系统架构近来一直是个热点话题。使用 Hadoop YARN(http://hadoop.apache.org)提供的一个通用计算框架，再引入流处理内存计算框架 Spark(http://spark.apache.org/)构建时效化推荐系统是应对 TB 级以上数据的一种选择方案。

推荐算法方面的研究包括协同过滤、近邻聚类、基于内容的推荐、贝叶斯网络、基于图的、关联规则等, 而协同过滤可分为基于内存和基于模型。基于内存的协同过滤是利用系统中已有用户的历史信息计算与目标用户的近邻，然后利用近邻对物品的喜好程度预测目标用户对该物品的喜好程度。基于模型的协同过滤算法则是利用已有信息训练出预测模型，通过模型进行预测。

在线学习算法[7]是一种快速、简单、较少的统计假设的算法。早期如 Crammer 等人提出的 Passive-Aggressive（PA）一阶在线学习算法，最近，又有研究者提出通过学习置信信息改进在线学习效果的二阶在线学习算法, 如 Dredze 等人[10]提出的 Confidence-Weight（CW）学习，通过维持高斯分布，并应用它控制参数更新大小和方向，其它相关的二阶在线学习规则如 Adaptive Regularization of Weight(AROW)[11]、New Adaptive Regularization of Weight(NAROW)[12]、Soft Confidence Weighted(SCW)[13]等。这些算法最初都是用于分类问题，相关的在线协同过滤算法也多是利用一阶优化方法，如梯度下降法 、均值等，而忽略了二阶信息，把 AROW 和二阶协同过滤和在线学习结合的如 Lu 等人[14]的 Confidence Weighted Online Collaborative Filtering(CWOCF) 算法。虽然 CWOCF 算法中的 AROW 学习规则效果较好，但其更新规则仍然较强，可能在某些情况下出现过拟合，并且 AROW 的学习对噪声数据的处理仍有待改进，计算成本仍可降低。

尽管基于协同过滤的推荐系统在实践中取得了很好的成绩，但传统的批量学习的基于模型的协同过滤方法往往对新用户（物品）更新迟缓，随着数据量的增长，对用户（物品）的动态变化无法及时处理，模型重训练成本高, 扩展性不足, 对噪声数据健壮性有待提高。

基于以上问题，本文基于 Hadoop YARN，在 Spark 集群上，提出在置信权重在线协同过滤算法 CWOCF[14]基础上，引入自适应软边缘的软置信权重（Soft Confidence Weighted，SCW），在二阶在线协同过滤中动态学习，实时处理用户（物品）变化，及时扩展数据并降低计算成本，提升推荐的实时性和准确性，提高对噪声数据的健壮性。

本文第 1 节阐述在大数据环境下基于 Hadoop YARN 时效化协同过滤推荐系统的 Spark 流处理推荐框架；第 2 节详述新算法各环节具体步骤；第 3 节介绍测试项目和实验环境，并通过多组对比实验，分析实验结果；第 4 节为全文的总结和展望。

## 1 Spark 流处理推荐框架

早期的大数据并行处理框架 Hadoop 受限于单点故障及计算模式相对单一，Hadoop2.0 引入 YARN 这一通用资源管理系统，提升了系统可靠性和整个集群的资源利用率，使其成为可以运行包括实时流处理框架 Storm(http://storm.incubator.apache.org/)及 Spark、MPI 等多种大数据处理框架及编程模式。

为了克服 MapReduce 在交互式及迭代式计算方面的不足，Spark 引入了弹性分布式数据集（Resilient Distributed DataSets，RDD）模型，以充分利用内存资源提升计算效率。与其它的大数据处理框架不同的是，Spark 可以在 Shark、MLlib、GraphX 和 Spark Streaming 的基础上利用一个引擎

高效的处理从 ETL 到 SQL 到机器学习再到流数据的处理, 例如，使用 Spark 加 Spark Streaming（或 Shark、BlinkDB）用于实时和批处理；使用 Spark Streaming 加 MLlib 用于流处理和机器学习；使用 Spark 加 GraphX 用于图流水线等。

图 1 为基于 Hadoop 构建的一套 Spark on YARN 实时推荐系统架构[15]。系统分为三层，由下至上依次为离线层、中间层和在线层。最底层的原始数据主要为用户行为日志（如点击链接、时间戳等）、评分等。对于一个如 Amazon、淘宝等这样的大型商业网站，每天产生的日志文件达到 TB 级，可以将这些数据存储在 Hadoop Hive 中，然后进行抽取、转化、装载得到目标数据。为了提高数据处理速度和推荐效率，直接利用 MLlib 推荐算法或基于 Spark 定制算法对目标数据分析建模；而在线层的 Spark Streaming 点击流等实时数据可存储在高性能列式数据库 HBase 中。模型确定后，在模型融合系统中进行多样化准确的推荐并反馈结果。 Spark 集群运行在 YARN 上主要有如下几步[16],

**步骤 1** 通过 YARN-Spark Client 将 Spark Application 提交到 YARN 上；

**步骤 2** ResourceManager 为 YARN-Spark Application 分配资源，并将之运行在一个节点上；

**步骤 3** YARN-Spark ApplicationMaster 在自己内部启动 ClusterScheduler，生成 DAG 图，启动 Web UI 服务等；

**步骤 4** YARN-Spark ApplicationMaster 按照用户配置向 ResourceManager 申请资源，并在申请到的 Container 中启动 StandaloneExecutorBackend 服务，StandaloneExecutorBacken-
d 通过 akka 向 ClusterScheduler 注册，以等待领取任务；

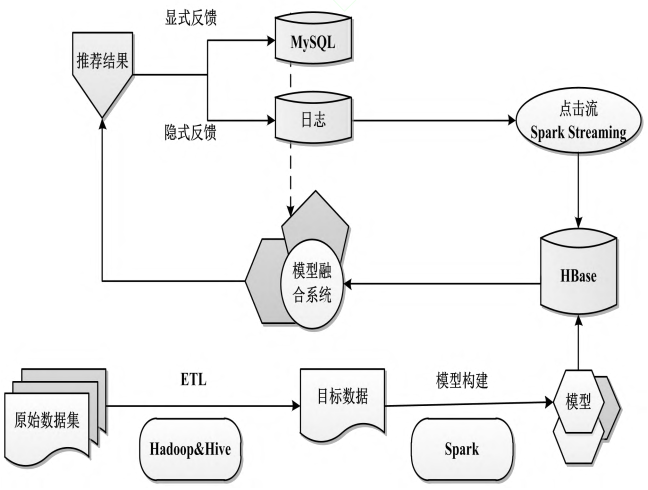


图 1 Spark on YARN 推荐系统架构

**步骤 5** ClusterSchedule 向 StandaloneExecutorBackend 分配新任务，StandaloneExecutorBackend 收到任务后执行它，当所有任务运行完成后，ApplicationMaster 归还所有资源，并退出。

## 2 软置信权重二阶在线协同过滤

### 2.1 问题背景

在传统的协同过滤中，假设有 n 个用户和 m 个物品，则用户 a 对物品 b 的评分可表示为 $r_{ab}$，其中 $r_{ab} \in \{1,2,3,4,5\}$，用户的评分形成一个有缺失值的评分矩阵 $R \in \mathbb{R}^{n \times m}$。通常，协同过滤的目标即根据评分矩阵 R 中已知的评分预测其中未知评分。

Koren 等人利用矩阵分解技术将用户和物品映射到联合低维潜在因子空间中，则用户和物品矩阵可分别表示为 $U \in \mathbb{R}^{k \times n}$，$V \in \mathbb{R}^{k \times m}$，用户 a 和物品 b 可以分别用向量 $U_a \in \mathbb{R}^k$ 和 $V_b \in \mathbb{R}^k$ 表示（$k << n,m$），评分 $r_{ab}= U_a^T V_b$，从而变为优化下式：

$$\arg\min_{U \in \mathbb{R}^{k \times n}, V \in \mathbb{R}^{k \times m}} \|R - U^T V\|_F^2 \quad (1)$$

其中，$\|A\|_F$ 为矩阵 A 的弗罗贝尼乌斯范数(Frobenius norm)。由于评分矩阵 R 的稀疏性，传统的如奇异值分解(Singular Value Decomposition，SVD)并不合适，可以转化为求解仅与已知评分的预测损失和有关的如下目标函数[14]。

$$\arg\min_{U \in \mathbb{R}^{k \times n}, V \in \mathbb{R}^{k \times m}} \sum_{(a,b) \in D} \ell(U_a, V_b, r_{ab}) \quad (2)$$

其中，D 为已知评分的集合，损失函数$\ell$为最小化某一评分矩阵。在此，用均方根误差(Root Mean Square Error,RMSE) 和平均绝对误差 (Mean Absolute Error,MAE)作为后文评价算法准确率的标准。其中，$\hat{r}_{a,b} = U_a^T V_b$ 为预测评分值，$T$ 为测试评分数量。

$$\mathrm{RMSE} = \sqrt{\frac{1}{|T|} \sum_{(a,b) \in T} (r_{a,b} - \hat{r}_{a,b})^2}$$

$$\mathrm{MAE} = \frac{1}{|T|} \sum_{(a,b) \in T} |r_{a,b} - \hat{r}_{a,b}|$$

从而对应 RMSE 与 MAE 的损失函数可转化为下式：

$$\mathrm{RMSE}: \quad \ell_1(U_a, V_b, r_{a,b}) = (r_{a,b} - U_a^T V_b)^2 \quad (3)$$

MAE: $\ell_2(U_a,V_b,r_{a,b}) = |r_{a,b} - U_a^T V_b|$ (4)

## 2.2 自适应软边缘的软置信权重学习

在分类问题中，对于 t 时刻的输入实例 $x_t \in \mathbb{R}^d$ 得到预测标签 $\hat{y}_t \in \{-1,+1\}$，并产生与真实标签 $y_t \in \{-1,+1\}$ 间的损失 $\ell(\hat{y}_t, y_t)$。AROW 学习算法假设权重向量 **w** 服从高斯分布：$N(\mu,\Sigma)$，其中，均值向量 $\mu \in \mathbb{R}^d$，协方差矩阵 $\Sigma \in \mathbb{R}^{d\times d}$，并最小化权重向量 **w** 间的 Kullback-Leibler 距离与置信惩罚向量之和，从而变为优化如下目标函数：

$$(\mu_{t+1},\Sigma_{t+1}) = \arg\min_{\mu,\Sigma} D_{KL}(N(\mu,\Sigma)\|N(\mu_t,\Sigma_t)) + \frac{1}{2\gamma}\ell^2(\mu;(x_t,y_t)) + \frac{1}{2\gamma}x_t^T\Sigma_t x_t \quad (5)$$

SCW 借鉴软间隔支持向量机，弱化了强更新规则，用于改进 NHERD、NAROW、CW 等学习算法在分类问题中的不足。与同类算法相比，SCW 在随机噪声上健壮性更强、较大的自适应软边缘使其对新数据更新更快并且计算时间更少，很好的处理了线性不可分数据，因此本文采用 SCW 学习规则。

引入损失函数 $\ell^{\phi}$，SCW 算法即为优化等式(7)，其中 C 为平衡参数，用于平衡更新规则强弱程度。

$$\ell^{\phi}(N(\mu,\Sigma);(x_t,y_t)) = \max(0, \phi\sqrt{x_t^T\Sigma x_t} - y_t\mu\cdot x_t) \quad (6)$$

$$T_E(\mu_{t+1},\Sigma_{t+1}) = \arg\min_{\mu,\Sigma} D_{KL}(N(\mu,\Sigma)\|N(\mu_t,\Sigma_t)) + C\,\ell^{\phi}(N(\mu,\Sigma);(x_t,y_t)) \quad (7)$$

## 2.3 自适应软边缘的二阶在线协同过滤算法

利用 SCW 学习规则，提出一种自适应软边缘的软置信权重二阶在线协同过滤算法(Soft Confidence Weighted Online Collaborative Filtering, SCWOCF)。假设用户和物品向量分别服从 $U_a \sim N(\mu_{ua},\sum_{ua})$ 和 $V_b \sim N(\mu_{vb},\sum_{vb})$ 的高斯分布，则可固定其一，再对另一个优化来降低两个高斯分布内积建模的复杂性[14]。对等式(7)分别更新用户向量 $U_a$ 和物品向量 $V_b$，即为求解如下函数，

$$T_U(\mu_{ua},\Sigma_{ua}) = \arg\min_{\mu,\Sigma} \frac{1}{2}\log\frac{\det\Sigma_{ua,t}}{\det\Sigma_{ua}} + \frac{1}{2}\mathrm{Tr}(\Sigma_{ua,t}^{-1},\Sigma) + \frac{1}{2}(\mu_{ua}-\mu_{ua,t})^T\Sigma_{ua,t}^{-1}(\mu_{ua}-\mu_{ua,t}) - \frac{k}{2} + C\,\ell^2(\mu_{ua},\Sigma_{ua},V_b,r_{ab}) \quad (8)$$

$$T_V(\mu_{vb},\Sigma_{vb}) = \arg\min_{\mu,\Sigma} \frac{1}{2}\log\frac{\det\Sigma_{vb,t}}{\det\Sigma_{vb}} + \frac{1}{2}\mathrm{Tr}(\Sigma_{vb,t}^{-1},\Sigma) + \frac{1}{2}(\mu_{vb}-\mu_{vb,t})^T\Sigma_{vb,t}^{-1}(\mu_{vb}-\mu_{vb,t}) - \frac{k}{2} + C\,\ell^2(U_a,\mu_{vb},\Sigma_{vb},r_{ab}) \quad (9)$$

以上目标函数一部分依赖μ，另一部分依赖Σ，从而转化为根据以下规则优化更新均值向量μ和协方差矩阵Σ。

**改进规则 1** 在 SCWOCF 中，给定一已知评分对 $(a,b,r_{ab})$，改进的对 RMSE 和 MAE 更新Σ规则如下式，

$$\sum_{ua,t+1} = \sum_{ua,t} - \frac{\tau\phi\sum_{ua,t}V_bV_b^T\sum_{ua,t}}{\sqrt{V_b^T\sum_{ua,t+1}V_b} + \tau\phi V_b^T\sum_{ua,t}V_b} \quad (10)$$

其中，$\tau = \frac{-m_t\upsilon_t(1+\frac{\phi^2}{2}) + \sqrt{\Delta}}{\upsilon_t^2(1+\phi^2)}$，$\Delta = m_t^2\upsilon_t^2(1+\frac{\phi^2}{2})^2 - \upsilon_t^2(1+\phi^2)(m_t^2-\phi^2\upsilon_t)$，$\phi = \Phi^{-1}(\eta)$，$\upsilon_t = V_b^T\sum_{ua,t}V_b$，$m_t = (V_b^T\mu_{ua})$，式(8)中 $\ell^2(\mu_{ua},\sum_{ua},V_b,r_{ab}) = \phi\sqrt{V_b^T\sum_{ua,t}V_b} - \mu_{ua}^TV_b - r_{ab}$，$\Phi$ 为正态分布的累积函数，$\eta$经交叉验证从{0.5,0.55,…,0.9, 0.95}中选择最优值。证明过程与文献[13]类似。

与式(10)类似，可以得到改进规则如下式，

$$\sum_{vb,t+1} = \sum_{vb,t} - \frac{\tau\phi\sum_{vb,t}U_aU_a^T\sum_{vb,t}}{\sqrt{U_a^T\sum_{vb,t+1}U_a} + \tau\phi U_a^T\sum_{vb,t}U_a} \quad (11)$$

其中，$\tau$、$\phi$与式(10)相同，且此处$\tau$、$\phi$中涉及的参数 $\upsilon_t = U_a^T\sum_{vb,t}U_a$，$m_t = (U_a^T\mu_{vb})$，式(9)中 $\ell^2(U_a,\mu_{vb},\sum_{vb},r_{ab}) = \phi\sqrt{U_a^T\sum_{vb,t}U_a} - \mu_{vb}^TU_a - r_{ab}$。

**改进规则 2** 在 SCWOCF 中，给定一已知评分对 $(a,b,r_{ab})$，改进的对 RMSE 更新μ规则如下式，

$$\mu_{ua,t+1} = \mu_{ua,t} - 2C\sum_{ua,t}V_b\left(\frac{\phi\sqrt{V_b^T\sum_{ua,t}V_b} - \mu_{ua,t}^TV_b - r_{ab}}{1-2CV_b^T\sum_{ua,t}V_b}\right) \quad (12)$$

$$\mu_{vb,t+1} = \mu_{vb,t} -$$

$$2C\Sigma_{vb,t}V_b\left(\frac{\phi\sqrt{U_a^T\Sigma_{vb,t}U_a}-\mu_{ua,t}^TU_a-r_{ab}}{1-2CU_a^T\Sigma_{vb,t}U_a}\right) \quad (13)$$

证明过程以式(12)为例，先对目标函数 $T_U(\mu_{ua},\Sigma_{ua})$ 求关于 $\mu_{ua}$ 的导数并令其为零求得 $\mu_{ua}$ 的表达式，对 $\mu_{ua}$ 的表达式两边取 $V_b$ 的点积所得的式子再带回 $\mu_{ua}$ 的表达式得到式(12)，其中，式(12)相关参数与式(10)相同, 式(13)相关参数与式(11)相同。

**改进规则 3**　在 SCWOCF 中，给定一已知评分对(a,b, $r_{ab}$)，对 MAE 更新μ如下式，证明过程与文献[14]类似。

若 $\hat{r}_{a,b}-r_{ab}>CV_b^T(\Sigma_{ua,t}V_b)$ 则，

$$\mu_{ua,t+1}=\mu_{ua,}-C\Sigma_{ua,t}V_b$$

若 $\hat{r}_{a,b}-r_{ab}<-CV_b^T(\Sigma_{ua,t}V_b)$ 则，

$$\mu_{ua,t+1}=\mu_{ua,t}+C\Sigma_{ua,t}V_b$$

其它：
$$\mu_{ua,t+1}=\mu_{ua,t} \quad (14)$$

若 $\hat{r}_{a,b}-r_{ab}>CU_a^T(\Sigma_{Vb,t}U_a)$ 则，

$$\mu_{vb,t+1}=\mu_{vb,t}-C\Sigma_{Vb,t}U_a$$

若 $\hat{r}_{a,b}-r_{ab}<-CU_a^T(\Sigma_{Vb,t}U_a)$ 则，

$$\mu_{vb,t+1}=\mu_{vb,t}+C\Sigma_{Vb,t}U_a$$

其它：
$$\mu_{vb,t+1}=\mu_{ua,t} \quad (15)$$

综上，改进 CWOCF-I 算法[14]后，将上述自适应软边缘的软置信权重二阶在线协同过滤算法(Soft Confidence Weighted Online Collaborative Filtering, SCWOCF)定义为 SCWOCF-I，算法总结如下：

*输入*：评分对序列{($a_t$, $b_t$, $r_{ab}$), t = 1,…,T}

*初始化*：随机用户矩阵 $U\in\mathbb{R}^{n\times k}$、物品矩阵 $V\in\mathbb{R}^{m\times k}$，

$\Sigma_n=I$

在 t = 1, 2,…,T 做如下*循环*：

当有用户 $a_t$ 对物品 $b_t$ 的评分预测请求时：

计算得到预测评分 $\hat{r}_{a_t,b_t}=U_{at}V_{bt}^T$，并由真实评分 $r_{at,bt}$，计算损失$\ell(U_a,V_b,r_{a,b})$

由规则 1~3 更新 $U_{at}$、$V_{bt}$、$\Sigma_{ua}$、$\Sigma_{vb}$

*结束循环*

## 2.4　SCWOCF 算法对大数据集优化策略

传统的协同过滤算法常假设在线学习的过程中，n 个用户和 m 个物品已知并固定不变，这种假设往往与现实中的系统环境不符。在用户和物品不断变化时，利用在线学习算法扩展已学习分布的参数即可动态实时更新用户和物品数据，做到及时高效的推荐，同时，为降低 SCWOCF-I 时间和空间复杂度，可只对协方差矩阵的对角元素更新，具体叙述如下：

**改进规则 4**　在 SCWOCF 中，给定一已知评分对(a,b, $r_{ab}$)，改进的对 RMSE 和 MAE 更新Σ协方差矩阵对角元素如下式，

$$\Sigma_{ua,t+1}=\Sigma_{ua,t}-\frac{\tau\phi\Sigma_{ua,t}\odot V_b\odot V_b\odot\Sigma_{ua,t}}{\sqrt{V_b^T(\Sigma_{ua,t+1}\odot V_b)}+\tau\phi V_b^T(\Sigma_{ua,t}\odot V_b)} \quad (16)$$

其中，$\tau$、$\phi$与式(10)相同，且此处$\tau$、$\phi$中涉及的参数 $u_t=V_b^T(\Sigma_{ua,t+1}\odot V_b)$，$\upsilon_t=V_b^T(\Sigma_{ua,t}\odot V_b)$，式(8)中$\ell^2(\mu_{ua},\Sigma_{ua},V_b,r_{ab})=\phi\sqrt{V_b^T(\Sigma_{ua,t}\odot V_b)}-\mu_{ua}^TV_b-r_{ab}$，$\odot$为智能内积，$\Sigma_{ua}$和$\Sigma_{vb}$在对角更新时使用 k 维列向量以节省空间。

与式(16)类似，可以得到改进规则如下式，

$$\Sigma_{vb,t+1}=\Sigma_{vb,t}-\frac{\tau\phi\Sigma_{vb,t}\odot U_a\odot U_a\odot\Sigma_{vb,t}}{\sqrt{U_a^T(\Sigma_{vb,t+1}\odot U_a)}+\tau\phi U_a^T(\Sigma_{vb,t}\odot U_a)} \quad (17)$$

其中，$\tau$、$\phi$与式(10)相同，且此处$\tau$、$\phi$中涉及的参数 $u_t=U_a^T(\Sigma_{vb,t+1}\odot U_a)$, $\upsilon_t=U_a^T(\Sigma_{vb,t}\odot U_a)$，式(9)中$\ell^2(U_a,\mu_{vb},\Sigma_{vb},r_{ab})=\phi\sqrt{U_a^T(\Sigma_{vb,t}\odot U_a)}-\mu_{vb}^TU_a-r_{ab}$。

**改进规则 5**　在 SCWOCF 中，给定一已知评分对(a,b, $r_{ab}$)，改进的对 RMSE 对角更新μ规则如下式，

$$\mu_{ua,t+1}=\mu_{ua,t}-2C(\Sigma_{ua,t}\odot V_b)\left(\frac{\phi\sqrt{V_b^T(\Sigma_{ua,t}\odot V_b)}-\mu_{ua,t}^TV_b-r_{ab}}{1-2CV_b^T(\Sigma_{ua,t}\odot V_b)}\right) \quad (18)$$

$$\mu_{vb,t+1}=\mu_{vb,t}-2C(\Sigma_{vb,t}\odot V_b)\left(\frac{\phi\sqrt{U_a^T(\Sigma_{vb,t}\odot U_a)}-\mu_{ua,t}^TU_a-r_{ab}}{1-2CU_a^T(\Sigma_{vb,t}\odot U_a)}\right) \quad (19)$$

其中，式(18)相关参数与式(16)相同, 式(19)相关参数与式(17)相同。

**改进规则 6**　在 SCWOCF 中，给定一已知评分对(a,b, $r_{ab}$)，对 MAE 对角更新μ如下式，证明过程与文献[14]类似。

若 $\hat{r}_{a,b} - r_{ab} > C V_b^T(\sum_{ua,t} \odot V_b)$ 则，

$$\mu_{ua,t+1} = \mu_{ua,} - C\sum_{ua,t} \odot V_b$$

若 $\hat{r}_{a,b} - r_{ab} < -C V_b^T(\sum_{ua,t} \odot V_b)$ 则，

$$\mu_{ua,t+1} = \mu_{ua,t} + C\sum_{ua,t} \odot V_b$$

其它：　$\mu_{ua,t+1} = \mu_{ua,t}$　(20)

若 $\hat{r}_{a,b} - r_{ab} > C U_a^T(\sum_{vb,t} \odot U_a)$ 则，

$$\mu_{vb,t+1} = \mu_{vb,t} - C\sum_{vb,t} \odot U_a$$

若 $\hat{r}_{a,b} - r_{ab} < -C U_a^T(\sum_{vb,t} \odot U_a)$ 则，

$$\mu_{vb,t+1} = \mu_{vb,t} + C\sum_{vb,t} \odot U_a$$

其它：　$\mu_{vb,t+1} = \mu_{ua,t}$　(21)

综上，改进 CWOCF-II 算法[14]后，新的大数据集的自适应软边缘的二阶在线协同过滤算法 SCWOCF-II 总结如下：

*输入*：评分对序列{($a_t$, $b_t$, $r_{ab}$), t = 1,…,T}

*初始化*：U=V=[], $\Sigma_u=\Sigma_v$=[]

在 t = 1, 2,…,T 做如下*循环*：

当有用户 $a_t$ 对物品 $b_t$ 的评分预测请求时：

*(i)* 如果用户 at 为新用户，则初始化 $U_a$ 为一随机向量，并扩展用户矩阵：U=[U;$U_{at}$]，扩展协方差矩阵：$\Sigma_u$=[$\Sigma_u$;**1**]

*(ii)* 如果物品 bt 为新物品，则初始化 $V_b$ 为一随机向量，并扩展物品矩阵：V=[V;$V_{bt}$]，协方差矩阵$\Sigma_v$=[$\Sigma_v$;**1**]

计算得到预测评分 $\hat{r}_{a_t,b_t} = U_{at}V_{bt}^T$，并由真实评分 $r_{at,bt}$，计算损失$\ell(U_a,V_b,r_{a,b})$

由规则 4~6 更新协方差矩阵对角$\Sigma u_a$、$\Sigma v_b$、$U_{at}$、$V_{bt}$

*结束循环*

# 3　实验与分析

## 3.1 实验数据与环境

为了实验的客观性和可比性，使用 GroupLens 研究小组提供的 MovieLens 和 HetRec 2011 测试数据(http://grouplens.org/datasets/movielens/)。如表 1 所示，数据分为四组，数据的稀疏度都较高，首先使用其中两组中等规模的数据集 MovieLens 100k 和 HetRec 2011 验证算法的基本表现，随后用两个较大规模的数据集 MovieLens 1M 和 MovieLens 10M 重点测试算法在较高负载情况下的实时性，以模拟其在大数据环境下实时推荐效果。

实验环境用虚拟机构建 Spark 集群，配置 5 个节点。其中，一个节点为 Master，担负 NameNode、SecondaryNamenode、ResourceManager 角色，其余 4 个节点分别为 Slave1~Slave4，担负 NodeManager、DataNode 角色，安装内存 8G，每个节点分配约 1.5G 内存，处理器为 Intel Core i5-430M（频率为 2.26~2.53GHz）软件环境为 CentOS-6.6，Hadoop-2.2.0，Spark-0.9.0，Scala-1.1，jdk1.7.0_51。

**表 1　实验数据基本参数统计**

| 数据集 | 评分数量 | 物品数量 | 用户数量 | 评分范围 | 评分矩阵稀疏度 |
|---|---|---|---|---|---|
| MovieLens 100k | 100,000 | 1,682 | 943 | 1-5 | 93.7% |
| HetRec 2011 | 855,598 | 10,109 | 2,113 | 1-5 | 96.0% |
| MovieLens 1M | 1,000,209 | 3,900 | 6,040 | 1-5 | 95.8% |
| MovieLens 10M | 10,000,054 | 10,681 | 71,567 | 1-5 | 98.7% |

## 3.2　相关对比算法与说明

这里从大规模应用考虑，选用 SCWOCF-II 算法和相关在线协同过滤算法做对比测试并给出测试数据。选用的具体可比性算法包括 DA-OCF(Dual-Averaging Online Collaborative Filtering)[17]、OMTCF-VI(Online Multi-Task Collaborative Filtering)[5]、CWOCF-II(Confidence Weighted Online Collaborative Filtering)[14]、SCWOCF-II(Soft Confidece Weighted Online Collaborative Filtering)。为了可比性，DA-OCF 中参数$\lambda_U$、$\lambda_V$与 CWOCF 中参数$\alpha_1$、$\alpha_2$在对应数据集中经一次实验随机搜索获得，且$\lambda_U$与$\lambda_V$范围从 $10^{-5}$到 $10^{-1}$，$\alpha_1$、$\alpha_2$从 1 到 100；SCWOCF-II 中参数 C、η在对应数据集中交叉验证获得，$C \in \{2^{-4}, 2^{-3}, \cdots, 2^3, 2^4\}$，$\eta \in \{0.5,0.55,\ldots,0.9,0.95\}$；潜在因子维度 k 固定为常数 5 和 10。用均方根误差(RMSE)和平均绝对误差(MAE)作为评价指标，每个算法运行 5 次取均值。

## 3.3　中等数据集测试与分析

这里用 MovieLens 100k 和 Hetrec 2011 数据集

在 k=5 和 k=10 时分别测试算法效果，结果汇总为表 2，加粗数据为本测试项的最优值。

从结果中可以看出：第一，SCWOCF-II 算法比选取的三种在线协同过滤算法在几乎未增加计算时间前提下 RMSE 和 MAE 值最小，准确度最高，表明新算法比现有的一阶和二阶在线协同过滤算法更优；第二，在相同评价指标相同算法中，k=10 的误差要小于 k=5 的误差，表明 k 对算法的误差有显著影响，文献[18]对 CWOCF 算法在四个数据集中测试表明 k 约为 12 时，算法误差最小，此处也验证了这一现象；第三，新算法同时学习一阶和二阶信息，且增加了自适应软边缘，有选择的更新，控制了计算时间，降低误差的同时保持健壮性。

### 3.4 大数据集测试与分析

为更接近真实应用，选取两个大数据集 MovieLens 1M 和 MovieLens 10M 测试算法效果，结果汇总为表 3。

表 2 中等数据集测试结果汇总

| MovieLens 100k | k=5 | | | |
|---|---|---|---|---|
| | RMSE | Time(s) | MAE | Time(s) |
| DA-OCF | 1.2516 | 1.33 | 0.9831 | 1.41 |
| OMTCF-VI | 1.0602 | 2.52 | 0.8704 | 2.98 |
| CWOCF-II | 1.0421 | 2.74 | 0.8243 | 2.96 |
| SCWOCF-II | **1.0303** | 2.71 | **0.8038** | 2.95 |

| MovieLens 100k | k=10 | | | |
|---|---|---|---|---|
| | RMSE | Time(s) | MAE | Time(s) |
| DA-OCF | 1.2309 | 1.34 | 0.9710 | 1.40 |
| OMTCF-VI | 1.0411 | 2.61 | 0.8433 | 2.62 |
| CWOCF-II | 1.0227 | 2.63 | 0.8072 | 2.60 |
| SCWOCF-II | **1.0101** | 2.59 | **0.7931** | 2.61 |

| HetRec 2011 | k=5 | | | |
|---|---|---|---|---|
| | RMSE | Time(s) | MAE | Time(s) |
| DA-OCF | 1.0509 | 9.19 | 0.7549 | 9.12 |
| OMTCF-VI | 0.8811 | 16.92 | 0.6751 | 19.76 |
| CWOCF-II | 0.8753 | 17.86 | 0.6662 | 19.97 |
| SCWOCF-II | **0.8711** | 18.87 | **0.6605** | 20.72 |

| HetRec 2011 | k=10 | | | |
|---|---|---|---|---|
| | RMSE | Time(s) | MAE | Time(s) |
| DA-OCF | 1.0513 | 9.79 | 0.7633 | 9.77 |
| OMTCF-VI | 0.8739 | 17.94 | 0.6723 | 17.91 |
| CWOCF-II | 0.8734 | 19.66 | 0.6611 | 20.69 |
| SCWOCF-II | **0.8688** | 19.85 | **0.6522** | 21.46 |

从结果中可以看出：第一，SCWOCF-II 算法获得了最低的 RMSE 和 MAE 值，表明新算法在大数据集环境中依然保持较好的准确度；第二，随着数据集的增大，SCWOCF-II 算法与其它算法对比，计算时间的降幅逐渐增大，表明新算法的自适应软边缘在更新策略上的灵活性，从而降低了计算次数，对噪声数据健壮性更强，误差更小，更适合大数据环境中的应用。

### 3.5 算法健壮性及集群扩展性测试与分析

相比于 CWOCF-II 算法中采用的 AROW 学习规则，SCWOCF-II 算法使用的 SCW 学习规则通过加入自适应软边缘，对不同的实例通过概率公式分配不同的边缘，因而对噪声数据健壮性更强同时提升了准确度和更新次数，从而降低更新时间。

与推荐算法的混合攻击（随机攻击和平均值攻击）类似，为测试算法的健壮性，分别构造 MovieLens 10M 数据集的 1%、3%、5%、10%、15%、20%的用户评分数据注入其中，令 k=10，测试各算法的 RMSE 值变化，绘制如下图 2。

表 3 大数据集测试结果汇总

| MovieLnes 1M | k=5 | | | |
|---|---|---|---|---|
| | RMSE | Time(s) | MAE | Time(s) |
| DA-OCF | 1.1158 | 10.86 | 0.8668 | 10.94 |
| OMTCF-VI | 0.9814 | 23.62 | 0.7783 | 26.31 |
| CWOCF-II | 0.9667 | 20.73 | 0.7673 | 23.68 |
| SCWOCF-II | **0.9568** | 19.49 | **0.7602** | 20.83 |

| MovieLens 1M | k=10 | | | |
|---|---|---|---|---|
| | RMSE | Time(s) | MAE | Time(s) |
| DA-OCF | 1.1012 | 11.69 | 0.8593 | 11.43 |
| OMTCF-VI | 0.9705 | 24.89 | 0.7702 | 25.87 |
| CWOCF-II | 0.9591 | 22.32 | 0.7630 | 23.96 |
| SCWOCF-II | **0.9502** | 20.13 | **0.7554** | 21.37 |

| MovieLens 10M | k=5 | | | |
|---|---|---|---|---|
| | RMSE | Time(s) | MAE | Time(s) |
| DA-OCF | 1.0803 | 117.10 | 0.7662 | 117.97 |
| OMTCF-VI | 0.9511 | 528.68 | 0.7430 | 541.43 |
| CWOCF-II | 0.9104 | 516.73 | 0.7133 | 533.73 |
| SCWOCF-II | **0.9031** | 502.27 | **0.7012** | 506.42 |

| MovieLens 10M | k=10 | | | |
|---|---|---|---|---|
| | RMSE | Time(s) | MAE | Time(s) |
| DA-OCF | 1.0781 | 122.88 | 0.7653 | 127.92 |
| OMTCF-VI | 0.9415 | 898.34 | 0.7303 | 901.21 |
| CWOCF-II | 0.9039 | 879.21 | 0.7037 | 899.86 |
| SCWOCF-II | **0.9012** | 853.92 | **0.7002** | 870.88 |

这里将 NameNode 节点加入为 DataNode 节点，使用五个数据节点测试，并选取 MovieLens 10M 为测试数据，计算 RMSE 花费时间为测试项，设 k=10。由于各算法自身运行时间的差异，为了测试节点的增加对算法运行时间的减少大小，将 1 个 DataNode 时四个算法各自运行花费的时间作为各自基准线分别量化为值 100，依次增加 DataNode 个数得到四个算法各自运行时间并以各自基准运行时间为依据，将时间量化为(0,100]的值，绘制如下图 3。

从图中可看出：第一，随着数据节点个数的增加，四个算法的运行时间都明显下降，在 Spark 集群中，如果软硬件条件未受限，只需添加节点个数即可获得计算时间的相应降低，响应时间的提升；第二，在 1 至 3 个节点时，随着节点的增加，计算时间基本呈线性减少，超过 3 个节点时，由于实验平台处理器限制及节点间数据 I/O 增加等原因，时间降幅逐渐下降；第三，数据节点的增加对 SCWO

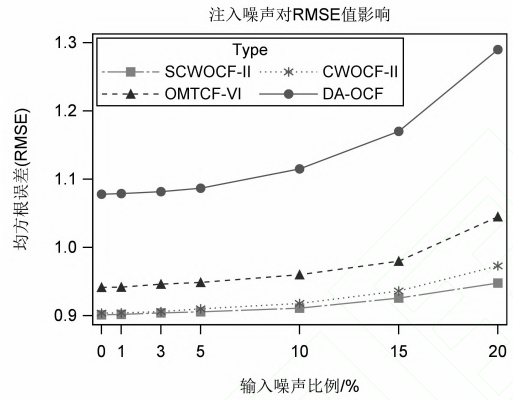


图 2　注入噪声对 RMSE 值影响

CF-II 算法的计算时间的降低效果最好，表明自适应边缘的更新在数据量大的环境中相比于其它三种算法，扩展性更强。

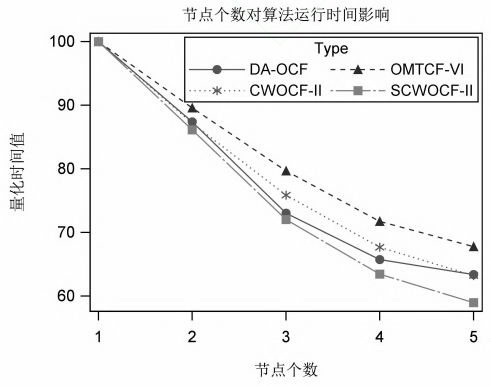


图 3　节点个数对算法运行时间影响

## 4 结束语

针对大数据环境的推荐系统，提出基于 Spark 平台，在置信权重在线协同过滤基础上，加入自适应软边缘的二阶在线协同过滤算法，提升算法准确率，降低了计算时间，增强了对噪声数据的健壮性，表明新算法更适合实际应用环境。

但受限于处理器性能和核心个数，部分影响了效果，且测试中未对算法在 Spark 环境中运行做任务优化处理。接下来的工作包括优化算法在 Spark 中的执行过程，同时可以利用其它实时流处理框架如 Storm 或 MPI 编程以进一步提升系统并行化和实时性。

## 参　考　文　献


[1] Diaz-Aviles E, Nejdl W, Drumond L, et al. Towards real-time collaborative filtering for big fast data[C]//Proceedings of the 22nd international conference on World Wide Web companion. International World Wide Web Conferences Steering Committee, 2013: 779-780.

[2] Cao J, Wu Z, Wang Y, et al. Hybrid Collaborative Filtering algorithm for bidirectional Web service recommendation[J]. Knowledge and information systems, 2013, 36(3): 607-627.

[3] Ge Y, Xiong H, Tuzhilin A, et al. Cost-Aware Collaborative Filtering for Travel Tour Recommendations[J]. ACM Transactions on Information Systems (TOIS), 2014, 32(1): (Article No.)4. http://dl.acm.org/citation.cfm?id=2576772.2559169&coll=DL&dl=GUIDE&CFID=532823498&CFTOKEN=83439047.

[4] Liu JG, Zhou T, Wang BH. Research progress of personalized recommendation system. Progress in National Science, 2009,19(1):1-15(
in Chinese with English abstract)

[5] Wang J, Hoi S C H, Zhao P, et al. Online multi-task collaborative filtering for on-the-fly recommender systems[C]//Proceedings of the 7th ACM conference on Recommender systems. New York: ACM, 2013: 237-244.

[6] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008: 426-434.

[7] Ling G, Yang H, King I, et al. Online learning for collaborative filtering[C]//Neural Networks (IJCNN), The 2012 International Joint Conference on. IEEE, 2012: 1-8.

[8] Jamali M, Ester M. TrustWalker: A random walk model for combining trust-based and item-based recommendation[C]//Proc of the 15th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2009:397-406.

[9] Liu Q, Chen E, Xiong H, et al. Enhancing collaborative filtering by user interest expansion via personalized ranking[J]. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2012, 42(1): 218-233.

[10] Dredze M, Crammer K, Pereira F. Confidence-weighted linear classification[C]//Proceedings of the 25th international conference on Machine learning. ACM, 2008: 264-271.

[11] Crammer K, Kulesza A, Dredze M. Adaptive regularization of weight vectors[J]. Machine learning, 2013, 91(2): 155-187.

[12] Orabona F, Crammer K. New Adaptive Algorithms for Online Classification[C]//Neural Information Processing Systems(NIPS). 2010: 1840-1848.

[13] Wang J, Zhao P, Hoi S C H. Exact soft confidence-weighted learning[J]//The 29th International Conference on Machine Learning(ICML). Edinburgh, Scotland, arXiv preprint arXiv:1206.4612, 2012.

[14] Lu J, Hoi S, Wang J. Second Order Online Collaborative Filtering[C]//Asian Conference on Machine Learning. 2013: 325-340.

[15] Spark on YARN[EB/OL]. [2013-12-07]. http://wenku.baidu.com/link?url=fy49VE9Ytg6djOqCR9EJB2vXQ0dv0fMp7FSYyalj14bnavUE7BqC7k5IxsKbHXOf7xXqsIUagveOw9WQKtyN5iRS_OBpt6cKCKOEKFJPvq.

[16] 董西成. Hadoop 技术内幕：深入解析 YARN 架构设计与实现原理[M]. 北京: 机械工业出版社, 2014:316-317.

[17] Zhao P, Hoi S C H, Jin R. Double updating online learning[J]. The Journal of Machine Learning Research, 2011, 12: 1587-1615.