文章编号: 1001-0920 (2014) 09-1553-09 **DOI:** 10.13195/j.kzyjc.2013.0720

基于最小包含球的异质空间大数据集快速相似度学习算法

董爱美[1,2], 王士同[1], 蒋亦樟[1], 黄成泉[1,3]

(1. 江南大学 数字媒体学院，江苏 无锡 214122；2. 齐鲁工业大学 信息学院，济南 250353；3. 贵州民族大学 理学院，贵阳 550025)

摘 要: 针对跨空间数据相似度学习问题提出的跨空间相似度学习 (CSAL) 算法表现出了良好的性能, 并已成功地应用于各类推荐系统中. 但构建一个完善的推荐系统, 其待处理的数据量常呈现大样本特征, 而 CSAL 算法并不具备大样本快速处理能力. 针对此不足, 提出了跨空间相似度学习-最小包含球 (CSAL-MEB) 方法和跨空间相似度学习-核向量机 (CSAL-CVM) 快速方法. CSAL-CVM 方法既具有渐近线性时间复杂度和空间复杂度的优点, 同时又继承了 CSAL 的良好性能. 相关实验亦验证了所提出方法的有效性.

关键词: 最小包含球；大数据；异质空间；相似度学习；推荐系统

中图分类号: TP181 **文献标志码:** A

Fast affinity learning algorithm for heterogeneous space large scale datasets using minimum enclosing ball

DONG Ai-mei[1,2], *WANG Shi-tong*[1], *JIANG Yi-zhang*[1], *HUANG Cheng-quan*[1,3]

(1. School of Digital Media，Jiangnan University，Wuxi 214122，China；2. Information School，Qilu University of Technology，Ji'nan 250353，China；3. Science School，Guizhou Minzu University，Guiyang 550025，China. Correspondent：DONG Ai-mei，E-mail：amdong@163.com)

Abstract: For the cross-space affinity learning problem, the recently-proposed cross-space affinity learng(CSAL) algorithm exhibits its good performance, and successfully applies to various recommendation systems. The data of a perfect recommendation system often have characteristics of large scale sample, but the CSAL algorithm does not have the capacity of fast processing big data. To solve this problem, the cross-space affinity learng-minimal enclosing ball(CSAL-MEB) method and cross-space affinity learng-core vector machine(CSAL-CVM) method are proposed. The CSAL-CVM method has the merits of asymmetric time complexity and space requirements independent of data scale as well as inherits the good performance of the CSAL algorithm. Experiments are given to verify the effectiveness of the proposed method.

Key words: minimum enclosing ball；big data；heterogeneous spaces；affinity learning；recommendation system

0 引 言

数据相似度学习已经成功地应用于机器学习领域中的聚类、分类、回归等数据挖掘问题[1-3]. 传统的数据相似度度量的是定义在同质空间上相同的特征空间. 然而, 在许多情况下需要度量异质空间数据的相似度, 很多应用问题的本质都可以理解为根据相似度来匹配异质数据, 最常见的应用便是推荐系统. 具体地, 如在电影推荐系统中[4-9], 用户和电影来自两个不同的特征空间, 该用户对所有电影的历史评分记录、个人信息 (年龄、性别、职业等) 和感兴趣的电影类型等用来描述用户信息, 所有用户对该电影的历史评分记录、电影名称、发布日期和电影类型 (动作、冒险、动画、喜剧等) 用来描述电影信息. 电影推荐系统的主要任务是为每个用户推荐感兴趣的电影, 用户与电影的相似度度量与用户对电影的评分记录成正比. 用户评分高的电影可以认为与该用户具有很高的相似度. 因此, 电影推荐系统需要学习用户与电影之间的相似度, 根据相似度来设计推荐策略. 除电影推荐系统外, 许多其他应用也可以归结为异质空间数据相似度学习问题. 网络搜索的主要任务是对不同的用户

收稿日期: 2013-06-02；**修回日期:** 2013-10-01.

基金项目: 国家自然科学基金项目(61170122, 61202311)；江苏省自然科学基金项目(BK2012552)；山东省高等学校科研计划项目(J14LN05).

作者简介: 董爱美(1978−), 女, 讲师, 博士生, 从事人工智能、模式识别的研究；王士同(1964−), 男, 教授, 博士生导师, 从事模式识别、数据挖掘、模糊神经网络等研究.

查询检索相关文档, 并根据相关性对文档进行排序, 查询与文档的相关性可以理解为两种异质实例间的相似度, 而搜索本质上就是为查询匹配相似文档. 图片标注的主要任务是用一些关键词描述图片的内容, 这个过程实际上是在匹配图片和关键词, 图片和关键词显然对应了两种异质数据, 它们之间的相关性可以通过相似度度量. 机器翻译的主要任务是将一种语言下的表达翻译成另一种语言下的表达, 为一种语言找寻翻译的过程也可以理解成根据相似性来匹配两种异质数据. 总之, 度量异质数据的相似度, 并根据相似度匹配两种异质的实例具有广泛的应用背景.

2011 年, Tang 等[10]引入跨空间二阶张量模型的概念来学习异质空间数据相似度, 提出了跨空间相似度学习 (CSAL) 的方法, 并将其应用于电影推荐系统. 此外, CSAL 算法由两个迭代的二次规划问题组成, 这两个迭代的二次规划问题的时间复杂度依赖于训练集的大小和异质空间维度的大小. 假设训练集表示为 T, 异质空间的维度分别为 p 和 q, 则每次迭代过程的时间复杂度为 $O(|T|(p+q))$. 在具有异质结构的不同特征空间上, 假设跨空间数据对的数量为 v, 则训练集的大小 $|T|$ 为 $v\times(v-1)/2$. 在电影推荐系统中, 异质空间数据表现为用户 user 和电影 item, 随着用户和电影数目的增加, 跨空间维度 p 和 q 成比例增长, 数据对 (user, item) 的数量 l 以平方的比例增长, 训练集的大小 $|T|$ 是庞大的. 因此, 在面对大规模异质空间数据集时, CSAL 算法有很大的局限性. 如何找到一种新的方法, 既能保持 CSAL 算法的良好性能, 又适用于大样本数据集正是本文的出发点. 本文引入分类间隔器的思想, 将 CSAL 算法中的分类间隔由 1 设为 ρ, 并采用 L2-SVM 形式设计出跨空间数据相似度学习方法 (CSAL-MEB 算法). 该方法吸收了 CSAL 算法的优点; 在该算法的求解过程中证明了 CSAL-MEB 算法可以转化为两个中心约束的最小包含球 (CCMEB) 问题, 从而可以解决大样本问题并可以采用快速近似算法核向量机[11-12]来求解; 最后提出了 CSAL-MEB 算法的近似快速求解算法, 最大优点是其渐近时间复杂度与训练样本容量呈线性关系, 空间复杂度与训练样本容量无关. 以核向量机算法为基础的大样本数据快速算法在概率密度估计、分类、聚类中已经得到一定程度的应用[13-15], 实验结果验证了本文方法的有效性.

1 相关概念与 CSAL 算法

由文献 [10] 得到的结果可知, CSAL 算法因其具备独特的异质空间度量学习能力, 使其在当前应用需求十分广泛的推荐系统中得到了较好的应用. 由该算法得到的智能推荐结果对于用户而言具有很高的参考价值. 下面对该算法的基本原理作简单的回顾.

为了便于理解, 给出如下相关符号和概念定义.

1) 给定两种类型实例 $\boldsymbol{X}=\{x_i\}_{i=1}^n$, $\boldsymbol{Y}=\{y_j\}_{j=1}^m$, 分别属于两个具有异质结构的不同特征空间. 其中: $x_i\in\boldsymbol{R}^p, y_j\in\boldsymbol{R}^q$, n 和 m 分别是两种类型对象的数目, p 和 q 分别是两种类型对象的维度, 下标代表对象在所在数据空间中的顺序, 上标代表对象的维数.

2) $A(x,y)$ 表示具有异质结构的两个实例之间的距离, 即相似度.

3) 经过预处理后的训练实例集为

$$T=\{x_i,y_s,y_t,h_{s,t}^i\};$$

$$h_{s,t}^i=\begin{cases}1,\ A(x_i,y_s)\geqslant A(x_i,y_t);\\ -1,\ A(x_i,y_s)<A(x_i,y_t);\end{cases}$$

训练样本集大小为 l.

1.1 CSAL 算法

CSAL 算法的基本思想是构建跨空间张量, 用来表示异质空间数据对的相互关系, 采用 SVM 理论学习跨空间数据相似度模型的张量参数, 将跨空间异质数据对的相似度定义为两个跨空间张量的内积, 从而得到任意两个跨空间异质数据的相似度. 跨空间数据实例 x 和 y 构建的跨空间张量为 $x\otimes y$, 其中 $x\in\boldsymbol{R}^p$, $y\in\boldsymbol{R}^q$. 引入跨空间张量 w, 则跨空间异质数据对相似度定义为

$$A(x,y)=\langle w,x\otimes y\rangle=\sum_{a=1}^{p}\sum_{b=1}^{q}w_{a,b}x^{(a)}y^{(b)}. \tag{1}$$

其中: $\otimes$ 为张量积; $w\in\boldsymbol{T}^{p,q}$ 为一个二阶张量, $w_{a,b}\in w$, $\langle\cdot,\cdot\rangle$ 为两个张量的内积; $x^{(a)}$ 为实例 x 的第 a 维, $y^{(b)}$ 为实例 y 的第 b 维; 用下标代表训练实例在训练实例集中的位置, 用上标代表训练实例的维度; $x^{(a)}\cdot y^{(b)}$ 为所属空间 $\boldsymbol{R}^p$、$\boldsymbol{R}^q$ 中第 a 维和第 b 维间的相关性, $w_{a,b}$ 为计算跨空间相似度的权向量. 基于式 (1), Tang 等根据经典 SVM 理论将跨空间具有异质结构的数据相似度学习转化为

$$\min_{w,\varepsilon_{i,s,t}}\ J=\frac{1}{2}||w||_F^2+\lambda\sum_{i,s,t}\varepsilon_{i,s,t};$$

$$\text{s.t. } h_{s,t}^i(\langle w,x_i\otimes y_s\rangle-\langle w,x_i\otimes y_t\rangle)\geqslant 1-\varepsilon_{i,s,t},$$

$$\varepsilon_{i,s,t}\geqslant 0,\ \{x_i,y_s,y_t,h_{s,t}^i\}\in T. \tag{2}$$

Tang 等进一步研究发现, 式 (2) 中二阶张量 w 的求解需要 $p\times q$ 个变量, 如果两个异质空间的数据维度特别大, 则易导致维数灾难问题. 因此, Tang 等又给出了另一种变通的方案: 将原始具有满阶张量参数的跨空间相似度模型分解为单阶模型, 从而式 (1) 可重新写为

$$A(x,y)=\langle w_x\otimes w_y,x\otimes y\rangle=$$

$$\sum_{a=1}^{p}w_x^{(a)}x^{(a)}\cdot\sum_{b=1}^{q}w_y^{(b)}y^{(b)}. \tag{3}$$

这种分解所具有的优势如下: 从机器学习的观点看, 是为了避免过拟合问题, 使得利用较少的训练实例数量便可学到具有较少参数的模型[16]; 从优化技巧的角度看, 具有较少参数的模型可以更加高效地优化得到. 尽管高阶结构比单阶结构具有更好的表达力, 但高阶结构本身带有的线性结构限制了跨空间中的非线性结构的情况. 另外, 式 (3) 所表达的分解方法可以大大降低计算复杂度和空间复杂度. 相应地, 式 (2) 重写为

$$\begin{aligned}&\min_{w,\varepsilon_{i,s,t}} J=\frac{1}{2}||w_x\otimes w_y||_F^2+\lambda\sum_{i,s,t}\varepsilon_{i,s,t};\\&\text{s.t. } h_{s,t}^i(\langle w_x\otimes w_y,x_i\otimes y_s\rangle-\langle w_x\otimes w_y,x_i\otimes y_t\rangle)\geqslant\\&\quad 1-\varepsilon_{i,s,t},\ \varepsilon_{i,s,t}\geqslant 0,\ \{x_i,y_s,y_t,h_{s,t}^i\}\in T.\end{aligned}\tag{4}$$

有关式 (4) 的具体优化求解过程可参见文献 [10].

1.2　CSAL 算法与大样本

众所周知, 对于一个完善的推荐系统, 大量的用户数据是必不可少的. 由经典的统计学理论可知, 数据量越多, 数据的真实分布越清晰. 因此, 若要构建一个足够确定的推荐系统, 使其推荐的结果更加符合用户的内在需求, 则必须面对大样本数据集. 然而, 目前的CSAL 算法存在以下局限性使其无法面对大样本: CSAL 算法由两个迭代的二次规划问题组成, 每次迭代过程中, 求解二次规划问题的时间复杂度和空间复杂度依赖于训练集的大小和异质空间维度的大小. 具体而言, 假设训练集表示为 T, 异质空间的维度分别为 p 和 q, 则每次迭代过程的时间复杂度为 $O(|T|(p+q))$, 空间复杂度为 $O(|T|^2)$. 而在一个完善的推荐系统中, 随着异质空间数据对的增多, $|T|$ 的数量呈指数增长, 因此该算法在面对大样本数据集时有很大的局限性.

综上所述, 虽然CSAL 算法无法面对大样本数据集, 但是其独特的异质空间度量学习能力仍是可以借鉴的. 为此, 本文致力于研究一种新型的具备大样本处理能力的CSAL 拓展算法. 该算法不仅能够保有CSAL 算法的异质空间度量学习能力, 而且能够使其在大样本的环境下具备快速处理的能力.

2　CSAL-MEB 原理与方法

为了使得CSAL 算法能够具备大样本快速处理的能力, 以及更符合现实的应用环境, 本文根据文献 [15-16] 的方法, 对CSAL 的目标函数 (4) 引入分类器的分类间隔思想, 将间隔由 1 设为 ρ, 并采用L2-SVM 思想提出了 CSAL-MEB 算法, 其目标函数形式如下:

$$\begin{aligned}&\min_{w_x,w_y,\xi_{i,s,t}} J=\frac{1}{2}||w_x\otimes w_y||_F^2+\frac{\lambda}{2}\sum_{i,s,t}\xi_{i,s,t}^2-\rho;\\&\text{s.t. } h_{i,s,t}(\langle w_x\otimes w_y,x_i\otimes y_s\rangle-\\&\quad\langle w_x\otimes w_y,x_i\otimes y_t\rangle)\geqslant\rho-\xi_{i,s,t},\\&\quad\{x_i,y_s,y_t,h_{i,s,t}\}\in T.\end{aligned}\tag{5}$$

在给出定理之前, 首先简单介绍最小包含球的相关理论.

2.1　最小包含球问题

最小包含球问题的原始目标函数为

$$\begin{aligned}&\min R^2;\\&\text{s.t.}\|c-\varphi(x_i)\|^2\leqslant R^2,\ i=1,2,\cdots,m.\end{aligned}\tag{6}$$

其中: φ 为对应于核 k 的特征映射, $B(c,R)$ 为核化特征空间中的最小包含球. 原始问题 (6) 相应的对偶问题的形式是二次规划问题, 即

$$\begin{aligned}&\max\alpha^{\mathrm{T}}\mathrm{diag}(K)-\alpha^{\mathrm{T}}K\alpha;\\&\text{s.t. }\alpha^{\mathrm{T}}1=1,\ \alpha\geqslant 0.\end{aligned}\tag{7}$$

其中: $\alpha=[\alpha_1,\cdots,\alpha_m]^{\mathrm{T}}$ 为拉格朗日乘子向量,

$$K_{m\times m}=[k(x_i,x_j)]=[\varphi(x_i)^{\mathrm{T}}\varphi(x_j)]$$

为相应的核矩阵. 假设核 k 满足

$$k(x,x)=\kappa\tag{8}$$

是常数, 式 (7) 可以写为

$$\begin{aligned}&\max\ -\alpha^{\mathrm{T}}K\alpha;\\&\text{s.t. }\alpha^{\mathrm{T}}1=1,\ \alpha\geqslant 0.\end{aligned}\tag{9}$$

最小包含球理论指出, 若核 k 满足式 (8), 则任何形如式 (9)的二次规划问题都可视为最小包含球问题.

2.2　中心约束的最小包含球问题

中心约束的最小包含球问题的原始目标函数为

$$\begin{aligned}&\min\ R^2;\\&\text{s.t. }\|c-\varphi(x_i)\|^2+\delta_i^2\leqslant R^2,\ i=1,2,\cdots,m.\end{aligned}\tag{10}$$

其中: $\delta_i\in\boldsymbol{R}$ 为 $\varphi(x_i)$ 的扩展维, 由此样本点 x_i 在特征空间上对应着点 $\begin{bmatrix}\varphi(x_i)\\ \delta_i\end{bmatrix}$, 相应的扩展一维的最小包含球的球心为 $\begin{bmatrix}c\\ 0\end{bmatrix}$. 原始问题 (10) 对应的对偶问题的形式也是二次规划问题, 即

$$\begin{aligned}&\max\alpha^{\mathrm{T}}(\mathrm{diag}(K)+\Delta)-\alpha^{\mathrm{T}}K\alpha;\\&\text{s.t. }\alpha^{\mathrm{T}}1=1,\ \alpha\geqslant 0.\end{aligned}\tag{11}$$

其中: $\Delta=(\delta_1^2,\delta_2^2,\cdots,\delta_n^2)\geqslant 0$ 为用户定义的值, 目的是使目标函数 (11) 中 α 的一次项系数为常数. 为了保证 Δ 的非负性, 在式 (11) 中增加新项 $-\varsigma\alpha^{\mathrm{T}}1$, $\varsigma\in\boldsymbol{R}$ 为常数. 由于 $-\varsigma\alpha'1=-\varsigma$, 增加一个常量不会改变问题的求解, 式 (11) 可以改写为

$$\begin{aligned}&\max\alpha^{\mathrm{T}}(\mathrm{diag}(K)+\Delta-\varsigma 1)-\alpha^{\mathrm{T}}K\alpha;\\&\text{s.t. }\alpha^{\mathrm{T}}1=1,\ \alpha\geqslant 0.\end{aligned}\tag{12}$$

其中 $\Delta=\varsigma 1-\mathrm{diag}(K)\leqslant 0$.

中心约束的最小包含球理论指出: 任何形如式

(12) 且 $\Delta \geqslant 0$ 的二次规划问题都可以视为中心约束的最小包含球问题.

定理 1 式 (5) 可分解为两个中心约束的最小包含球问题.

证明 式 (5) 相应的拉格朗日函数为

$$L(w_x, w_y, \rho, \xi_{i,s,t}, \alpha_{i,s,t}) = \frac{1}{2}||w_x \otimes w_y||_F^2 + \frac{\lambda}{2}\sum_{i,s,t}\xi_{i,s,t}^2 - \rho - \sum_{i,s,t}\alpha_{i,s,t}[h_{i,s,t}\langle w_x, x_i\rangle\langle w_y, y_s - y_t\rangle - \rho + \xi_{i,s,t}]. \quad (13)$$

由 KKT 条件, $L(w_x, w_y, \rho, \xi_{i,s,t}, \alpha_{i,s,t})$ 取得极值时有

$$\frac{\partial L}{\partial w_x} = 0,\ \frac{\partial L}{\partial w_y} = 0,\ \frac{\partial L}{\partial \xi_{i,s,t}} = 0,\ \frac{\partial L}{\partial \rho} = 0,$$

$$\frac{\partial L}{\partial w_x} = 0 \Rightarrow w_x = \sum_{i,s,t}\alpha_{i,s,t}h_{i,s,t}x_i\frac{\langle w_y, y_s - y_t\rangle}{||w_y||^2} = \sum_{i,s,t}\alpha_{i,s,t}h_{i,s,t}x_i\gamma_{s,t}, \quad (14)$$

$$\frac{\partial L}{\partial w_y} = 0 \Rightarrow w_y = \sum_{i,s,t}\alpha_{i,s,t}h_{i,s,t}(y_s - y_t)\frac{\langle w_x, x_i\rangle}{||w_x||^2} = \sum_{i,s,t}\alpha_{i,s,t}h_{i,s,t}(y_s - y_t)\eta_i, \quad (15)$$

$$\frac{\partial L}{\partial \rho} = 0 \Rightarrow \sum_{i,s,t}\alpha_{i,s,t} = 1, \quad (16)$$

$$\frac{\partial L}{\partial \xi_{i,s,t}} = 0 \Rightarrow \xi_{i,s,t} = \frac{\alpha_{i,s,t}}{\lambda}. \quad (17)$$

其中

$$\gamma_{s,t} = \frac{\langle w_y, y_s - y_t\rangle}{||w_y||^2}, \quad (18)$$

$$\eta_i = \frac{\langle w_x, x_i\rangle}{||w_x||^2}. \quad (19)$$

简单起见, 本文给出核化形式如下:

$$||w_x||^2 = \sum_{i',s',t'}\sum_{i,s,t}\alpha_{i',s',t'}\alpha_{i,s,t}\times h_{i',s',t'}h_{i,s,t}\gamma_{s',t'}\gamma_{s,t}\langle x_{i'}, x_i\rangle \quad (20)$$

$$||w_y||^2 = \sum_{i',s',t'}\sum_{i,s,t}\alpha_{i',s',t'}\alpha_{i,s,t}h_{i',s',t'}h_{i,s,t}\times \eta_{i'}\eta_i\langle y_{s'} - y_{t'}, y_s - y_t\rangle \quad (21)$$

$$\theta_{s,t} = \langle w_y, y_s - y_t\rangle = \sum_{i',s',t'}\alpha_{i',s',t'}h_{i',s',t'}\eta_{i'}\langle y_{s'} - y_{t'}, y_s - y_t\rangle, \quad (22)$$

$$\omega_i = \langle w_x, x_i\rangle = \sum_{i',s',t'}\alpha_{i',s',t'}h_{i',s',t'}\gamma_{s',t'}\langle x_{i'}, x_i\rangle. \quad (23)$$

辅助参数 $||w_x||^2, ||w_y||^2, \eta_i, \gamma_{s,t}, \omega_i, \theta_{s,t}$ 分为两组, 如表 1 所示.

表 1 两组辅助参数

w_x 相关的辅助参数	w_y 相关的辅助参数
$||w_x||^2, \eta_i, \omega_i$	$||w_y||^2, \gamma_{s,t}, \theta_{s,t}$

将式 (16) ∼ (18), (20), (23) 代入 (13), 得到对偶问题

$$\max_{\alpha_{i,s,t}} -\sum_{i,s,t}\sum_{i',s',t'}\alpha_{i,s,t}\alpha_{i',s',t'}\Big[h_{i,s,t}h_{i',s',t'}\gamma_{s',t'}\times \Big(\theta_{s,t} - \frac{1}{2}||w_y||^2\gamma_{s,t}\Big)\langle x_{i'}, x_i\rangle + \frac{\delta_{i,s,t;i',s',t'}}{2\lambda}\Big].$$

$$\text{s.t.}\ \sum_{i,s,t}\alpha_{i,s,t} = 1,\ \alpha_{i,s,t} \geqslant 0;$$

$$\delta_{i,s,t;i',s',t'} = \begin{cases} 1,\ (i,s,t) = (i',s',t'); \\ 0,\ (i,s,t) \neq (i',s',t'). \end{cases} \quad (24)$$

将式 (16), (17), (19), (21), (22) 代入 (13), 得到对偶问题

$$\max_{\alpha_{i,s,t}} -\sum_{i,s,t}\sum_{i',s',t'}\alpha_{i,s,t}\alpha_{i',s',t'}\Big[h_{i,s,t}h_{i',s',t'}\eta_{i'}\times \Big(\omega_i - \frac{1}{2}||w_x||^2\eta_i\Big)\langle y_{s'} - y_{t'}, y_s - y_t\rangle + \frac{\delta_{i,s,t;i',s',t'}}{2\lambda}\Big].$$

$$\text{s.t.}\ \sum_{i,s,t}\alpha_{i,s,t} = 1,\ \alpha_{i,s,t} \geqslant 0;$$

$$\delta_{i,s,t;i',s',t'} = \begin{cases} 1,\ (i,s,t) = (i',s',t'); \\ 0,\ (i,s,t) \neq (i',s',t'). \end{cases} \quad (25)$$

对于式 (24), 定义矩阵 Q_1, 令

$$[Q_1]_{i,s,t;i',s',t'} = h_{i,s,t}h_{i',s',t'}\gamma_{s',t'}\Big(\theta_{s,t} - \frac{1}{2}||w_y||^2\gamma_{s,t}\Big)\langle x_{i'}, x_i\rangle,$$

矩阵 Q_1 是非对称矩阵, 不改变式 (24) 的值. 定义矩阵

$$[K_1]_{i,s,t;i',s',t'} = \Big[\frac{Q_1 + {Q_1}^{\mathrm{T}}}{2}\Big]_{i,s,t;i',s',t'},$$

则推导得到

$$[K_1]_{i,s,t;i',s',t'} = h_{i,s,t}h_{i',s',t'}\frac{\theta_{s,t}\theta_{s',t'}}{2||w_y||^2}\langle x_{i'}, x_i\rangle. \quad (26)$$

定义矩阵

$$[\widetilde{K_1}]_{i,s,t;i',s',t'} = [K_1]_{i,s,t;i',s',t'} + \frac{\delta_{i,s,t;i',s',t'}}{2\lambda} = h_{i,s,t}h_{i',s',t'}\frac{\theta_{s,t}\cdot\theta_{s',t'}\langle x_i, x_{i'}\rangle}{2||w_y||^2} + \frac{\delta_{i,s,t;i',s',t'}}{2\lambda}, \quad (27)$$

容易证明矩阵 $\widetilde{K}_1$ 是半正定矩阵. 于是式 (24) 变为如下形式的二次规划问题:

$$\max_{\alpha_1} -{\alpha_1}^{\mathrm{T}}\widetilde{K}_1\alpha_1;$$
$$\text{s.t.}{\alpha_1}^{\mathrm{T}}1 = 1,\ \alpha_1 \geqslant 0. \quad (28)$$

对式 (28) 稍作变形, 得到

$$\max {\alpha_1}^{\mathrm{T}}(\mathrm{diag}(\widetilde{K}_1) + \Delta_1 - \varsigma_1 1) - {\alpha_1}^{\mathrm{T}}\widetilde{K}_1\alpha_1;$$
$$\text{s.t.}{\alpha_1}^{\mathrm{T}}1 = 1,\ \alpha_1 \geqslant 0. \quad (29)$$

其中: $\varsigma_1 \in \boldsymbol{R}$ 为常数, $\Delta_1 = \varsigma_1 1 - \mathrm{diag}(\widetilde{K}_1) \geqslant 0$. 因此式 (24) 可视为中心约束的最小包含球问题. Δ_1 和 ς_1 的值需要预先确定, 因此需要先计算出核矩阵 $\widetilde{K_1}$ 的对角线元素的值. 对于式 (25), 定义矩阵

$$[Q_2]_{i,s,t;i',s',t'} = h_{i,s,t}h_{i',s',t'}\eta_{i'}\Big(\omega_i - \frac{1}{2}||w_x||^2\eta_i\Big)\langle y_{s'} - y_{t'}, y_s - y_t\rangle.$$

矩阵 Q_2 是非对称矩阵, 不改变式 (25) 的值, 定义矩阵

$$[K_2]_{i,s,t;i',s',t'} = \Big[\frac{Q_2 + {Q_2}^{\mathrm{T}}}{2}\Big]_{i,s,t;i',s',t'},$$

则推导得到

$$[K_2]_{i,s,t;i',s',t'} = h_{i,s,t}h_{i',s',t'}\frac{\omega_i\cdot\omega_{i'}}{2||w_x||^2}[k_Y(y_s,y_{s'})+ k_Y(y_t,y_{t'}) - k_Y(y_t,y_{s'}) - k_Y(y_s,y_{t'})]. \quad (30)$$

定义矩阵

$$[\widetilde{K_2}]_{i,s,t;i',s',t'} = [K_2]_{i,s,t;i',s',t'} + \frac{\delta_{i,s,t;i',s',t'}}{2\lambda} = h_{i,s,t}h_{i',s',t'}\frac{\omega_i\cdot\omega_{i'}}{2||w_x||^2}\langle y_{s'}-y_{t'}, y_s-y_t\rangle + \frac{\delta_{i,s,t;i',s',t'}}{2\lambda}, \quad (31)$$

容易证明矩阵 $\widetilde{K_2}$ 是半正定矩阵. 于是式 (25) 变为如下形式的二次规划问题:

$$\max_{\alpha_2} \ -{\alpha_2}^{\mathrm{T}}\widetilde{K_2}\alpha_2; \quad \text{s.t. } {\alpha_2}^{\mathrm{T}}1 = 1,\ \alpha_2 \geqslant 0. \quad (32)$$

对式 (32) 稍作变形, 得到

$$\max {\alpha_2}^{\mathrm{T}}(\mathrm{diag}(\widetilde{K_2}) + \Delta_2 - \varsigma_2 1) \ - {\alpha_2}^{\mathrm{T}}\widetilde{K_2}\alpha_2; \quad \text{s.t. } {\alpha_2}^{\mathrm{T}}1 = 1,\ \alpha_2 \geqslant 0. \quad (33)$$

其中: $\varsigma_2 \in \boldsymbol{R}$ 为常数, $\Delta_2 = \varsigma_2 1 - \mathrm{diag}(\widetilde{K_2}) \geqslant 0$. 因此式 (25) 可视为中心约束的最小包含球问题. Δ_2 和 ς_2 的值需要预先确定, 因此需要先计算出核矩阵 $\widetilde{K_2}$ 的对角元素的值. 至此, 式 (5) 分解成为两个中心约束的最小包含球问题. □

由定理 1 的证明, 可以得证式 (5) 的求解过程是由两个中心约束的最小包含球问题构成, 通过求解这两个中心约束的最小包含球问题即可得到优化的参数 w_x 和 w_y. 然而, 由于参数 w_x 和 w_y 在各自所对应的优化问题中相互影响, 经研究, 本文选用模糊聚类、模糊神经网络等技术中经常采用的交替迭代策略[17-22]对上述两参数进行优化, 具体的实施步骤如下: 1) 固定 w_y 及其有关参数, 利用式 (28) (或 (29)) 及二次规划理论优化 w_x; 2) 固定 w_x 及其有关参数, 利用式 (32) (或 (33)) 及二次规划理论优化 w_y.

算法 CSAL-MEB 的描述如下:

输入: 大数据集 $T = \{x_i, y_s, y_t, h^i_{s,t}\}$, 初始化与 w_y 有关的辅助参数 $\theta_{s,t}, ||w_y||^2, \gamma_{s,t}$.

Repeat: 对式 (27) 和 (28) 用二次规划方法求解, 得到式 (28) 的解 α, 用 $\alpha^{(28)}$ 表示, 有

$$\omega_i \leftarrow \sum_{(i',s',t')\in T}\alpha^{(28)}_{i',s',t'}h_{i',s',t'}\gamma_{s',t'}\langle x_{i'}, x_i\rangle,$$

$$||w_x||^2 \leftarrow \sum_{(i,s,t)\in T}\alpha^{(28)}_{i,s,t}h_{i,s,t}\gamma_{s,t}\omega_i,$$

$$\eta_i \leftarrow \frac{\omega_i}{||w_x||^2}.$$

对式 (31)、(32) 用二次规划方法求解, 得到式 (32) 的解 α, 用 $\alpha^{(32)}$ 表示, 有

$$\theta_{s,t} \leftarrow \sum_{(i',s',t')T}\alpha^{(32)}_{i',s',t'}h_{i',s',t'}\eta_{i'}\langle y_{s'}-y_{t'}, y_s-y_t\rangle,$$

$$||w_y||^2 \leftarrow \sum_{(i,s,t)\in T}\alpha^{(32)}_{i,s,t}h_{i,s,t}\eta_i\theta_{s,t},$$

$$\gamma_{s,t} = \frac{\theta_{s,t}}{||w_y||^2}.$$

Until: 收敛.

输出: $\alpha^{(28)}_{i,s,t}\alpha^{(32)}_{i,s,t}\gamma_{s,t}\eta_i$.

2.3 算法收敛问题分析

对于所给出的 CSAL-MEB 算法, 以第 iter +1 步的迭代学习为例对其收敛性作如下分析:

1) $\theta_{s,t}, ||w_y||^2$ 固定, 目标函数 (28) 写为

$$\underset{\alpha_{i,s,t}}{\mathrm{argmin}}\ J(\alpha_{i,s,t}) = \underset{\alpha_{i,s,t}}{\mathrm{argmin}}\Big(\sum_{i,s,t}\sum_{i',s',t'}\alpha_{i,s,t}\alpha_{i',s',t'}\times \Big(h_{i,s,t}h_{i',s',t'}\frac{\theta_{s,t}\theta_{s',t'}\langle x_i, x_{i'}\rangle}{2\|w_y\|^2} + \frac{\delta_{i,s,t;i',s',t'}}{2\lambda}\Big)\Big),$$

进而求得关于 $\alpha_{i,s,t}$ 的一阶偏导数

$$\frac{\partial J}{\partial \alpha_{i,s,t}} = \sum_{i',s',t';(i,s,t\neq i',s',t')}\alpha_{i',s',t'}h_{i,s,t}h_{i',s',t'}\theta_{s,t}\theta_{s',t'}\times \frac{\langle x_i, x_{i'}\rangle}{2\|w_y\|^2} + 2\alpha_{i,s,t}\Big(h_{i,s,t}h_{i,s,t}\theta_{s,t}\theta_{s,t}\frac{\langle x_i, x_i\rangle}{2\|w_y\|^2} + \frac{1}{2\lambda}\Big),$$

二阶偏导数

$$\frac{\partial^2 J}{\partial {\alpha_{i,s,t}}^2} = \frac{\theta^2_{s,t}}{\|w_y\|^2} + \frac{1}{\lambda}.$$

由于 $\lambda > 0$, 可知 $\partial^2 J/\partial {\alpha_{i,s,t}}^2 > 0$, 因此 $J(\alpha_{i,s,t})$ 是变量 $\alpha_{i,s,t}$ 的严格凸函数. 由于 $J(\alpha_{i,s,t})$ 是 $\alpha_{i,s,t}$ 的凸函数, 可知式 (28) 所得的 $\alpha^{\mathrm{iter}+1}_{i,s,t}$ 是 $J(\alpha_{i,s,t})$ 的全局最优解, 从而有 $J(\alpha^{\mathrm{iter}+1}_{i,s,t}) \leqslant J(\alpha^{\mathrm{iter}}_{i,s,t})$.

2) 同理可证, 固定 $\omega_i, ||w_x||^2$, 目标函数 (32) 得到的 $\alpha^{(32)}_{i,s,t}$ 也是其全局最优解, 从而其优化目标函数也是递减的.

值得指出的是, 虽然 CSAL-MEB 算法每一步优化过程均是递减收敛的, 但其整个迭代优化过程并不能保证严格收敛, 其通常能收敛于某个局部最优解或鞍点[17-22]. 虽然此类算法不能保证严格收敛, 但已有的采用交替迭代优化技术的算法[17-22]表明此优化技术在大多数场合是非常简单有效的.

3 CSAL-CVM 算法

3.1 CSAL-CVM 算法描述

由定理 1 得出, CSAL-MEB 算法可以转化为两个中心约束的最小包含球问题, 因此可以采用近似快速求解算法 CVM (core vector machines)[11-12]; 进一步,

CSAL-MEB 算法的求解可以转化为两个中心约束的最小包含球的交替迭代优化问题. 给出交替迭代优化的近似快速求解算法 CSAL-CVM 如下:

输入: 初始化与 w_y 有关的辅助参数 $\theta_{s,t}, ||w_y||^2$, $\gamma_{s,t}$.

Repeat:

Step 1: 大数据集 $T=\{x_i, y_s, y_t, h^i_{s,t}\}$, 核心集逼近精度 ε_1, 参数 λ, ς_1, Δ_1, 对式 (27) 和 (29) 表示的二次规划求最小包含球 (用 $R^{(29)}$、$c^{(29)}$、$S^{(29)}$ 表示).

Step 1.1: 初始化最小包含球核心集 $S_0^{(29)}$, 球心为 $c_0^{(29)}$, 半径为 $R_0^{(29)}$, 迭代计数器 $t=0$;

Step 1.2: 若所有点都被球 $B(c_t^{(29)}, (1+\varepsilon)R_t^{(29)})$ 包围, 则转 Step 1.6;

Step 1.3: 在扩展的特征空间中找到离中心点 $c_t^{(29)}$ 最远的点 $z^{(29*)}$, 将该点加入核心集

$$S_{t+1}^{(29)} = S_t^{(29)} \bigcup \{z^{(29*)}\};$$

Step 1.4: 求解新的 MEB($S_{t+1}^{(29)}$), 且

$$c_{t+1}^{(29)} = c^{(29)}_{\mathrm{MEB}(S_{t+1}^{(29)})},\ R_{t+1}^{(29)} = R^{(29)}_{\mathrm{MEB}(S_{t+1}^{(29)})};$$

Step 1.5: $t=t+1$, 转 Step 1.2;

Step 1.6: 终止训练, 返回权重系数和核心集, 用 $\alpha^{(29)}$ 和 $S_t^{(29)}$ 表示, 有

$$\omega_i \leftarrow \sum_{(i',s',t')\in T} \alpha^{(29)}_{i',s',t'} h_{i',s',t'} \gamma_{s',t'} \langle x_{i'}, x_i\rangle,$$

$$||w_x||^2 \leftarrow \sum_{(i,s,t)\in T} \alpha^{(29)}_{i,s,t} h_{i,s,t} \gamma_{s,t} \omega_i,$$

$$\eta_i \leftarrow \frac{\omega_i}{||w_x||^2}.$$

Step 2: 大数据集 $T=\{x_i, y_s, y_t, h^i_{s,t}\}$, 核心集逼近精度 ε_2, 参数 λ, ς_2, Δ_2, 对式 (31) 和 (33) 表示的二次规划问题求最小包含球 (用 $R^{(33)}$、$c^{(33)}$、$S^{(33)}$ 表示).

Step 2.1: 初始化最小包含球核心集 $S_0^{(33)}$, 球心为 $c_0^{(33)}$, 半径为 $R_0^{(33)}$, 迭代计数器 $t=0$;

Step 2.2: 若所有点都被球 $B(c_t^{(33)}, (1+\varepsilon)R_t^{(33)})$ 包围, 则转 Step 2.6;

Step 2.3: 在扩展的特征空间中找到离中心点 $c_t^{(33)}$ 最远的点 $z^{(33*)}$, 将该点加入核心集

$$S_{t+1}^{(33)} = S_t^{(33)} \bigcup \{z^{(33*)}\};$$

Step 2.4: 求解新的 CCMEB($S_{t+1}^{(33)}$), 且

$$c_{t+1}^{(33)} = c^{(33)}_{\mathrm{MEB}(S_{t+1}^{(33)})},\ R_{t+1}^{(33)} = R^{(33)}_{\mathrm{MEB}(S_{t+1}^{(33)})},$$

$$c_{t+1}^{(33)} = c^{(33)}_{\mathrm{MEB}(S_{t+1}^{(33)})},\ R_{t+1}^{(33)} = R^{(33)}_{\mathrm{MEB}(S_{t+1}^{(33)})};$$

Step 2.5: $t=t+1$, 转 Step 2.2;

Step 2.6: 终止训练, 返回权重系数和核心集, 用 $\alpha^{(33)}$ 和 $S_t^{(33)}$ 表示, 有

$$\theta_{s,t} \leftarrow \sum_{(i',s',t')T} \alpha^{(33)}_{i',s',t'} h_{i',s',t'} \eta_{i'} \langle y_{s'} - y_{t'}, y_s - y_t\rangle,$$

$$||w_y||^2 \leftarrow \sum_{(i,s,t)\in T} \alpha^{(33)}_{i,s,t} h_{i,s,t} \eta_i \theta_{s,t},$$

$$\gamma_{s,t} = \frac{\theta_{s,t}}{||w_y||^2}.$$

Until: 收敛.

输出: $\alpha^{(29)}_{i,s,t}$, $\alpha^{(33)}_{i,s,t}$, $\gamma_{s,t}$, η_i, $S_t^{(29)}$, $S_t^{(33)}$.

CSAL-CVM 算法主要步骤实现说明:

1) 初始化 w_y 及有关参数 $\theta_{s,t}, ||w_y||^2, \gamma_{s,t}$ 时, 需要满足关系 $\gamma_{s,t} = \theta_{s,t}/||w_y||^2$.

2) 在 Step 1.1 和 Step 2.1 中, 初始化核心集时虽然可以任选 $z\in T$ 来初始化核心集 $S_0^{(29)}=\{z\}$, $S_0^{(33)}=\{z\}$, 但是好的初始化能提高算法的性能[17-18]. 本文采用这样的策略: 任意选取原始大数据集的一个子集 $S^{(29)}_{\mathrm{sub}}, S^{(33)}_{\mathrm{sub}}$, 从 $S^{(29)}_{\mathrm{sub}}$ 和 $S^{(33)}_{\mathrm{sub}}$ 中任取一点 $z^{(29)}, z^{(33)}$. 从 $S^{(29)}_{\mathrm{sub}}$ 和 $S^{(33)}_{\mathrm{sub}}$ 中找出距离 $z^{(29)}$ 和 $z^{(33)}$ 最远的点 $z_1^{(29)}, z_1^{(33)}$, 从 $S^{(29)}_{\mathrm{sub}}$ 和 $S^{(33)}_{\mathrm{sub}}$ 中找出距离 $z_1^{(29)}$ 和 $z_1^{(33)}$ 最远的点 $z_2^{(29)}, z_2^{(33)}$. 初始核心集为 $S_0^{(29)}=\{z_1^{(29)}, z_2^{(29)}\}$, $S_0^{(33)}=\{z_1^{(33)}, z_2^{(33)}\}$.

3) Step 1.2、Step 1.3 和 Step 2.2、Step 2.3 分别为计算样本集 T 中的点到中心点 $c_t^{(29)}$, $c_t^{(33)}$ 的距离, 时间复杂度分别为 $O(|S_t^{(29)}|^2 + |T||S_t^{(29)}|), O(|S_t^{(33)}|^2 + |T||S_t^{(33)}|)$. 当样本规模相当大时非常耗时, 本文实验采用 Smola 等[24]提出的一种加速方法. 该方法指出, 在样本集 T 中随机找一个样本子集 $T'^{(29)}$, $T'^{(33)}$, 在子集 $T'^{(29)}$ 和 $T'^{(33)}$ 中分别寻找离中心点 $c_t^{(29)}$, $c_t^{(33)}$ 最远的点来近似代替样本集 T 中的最远点, 并证明当子集大小为 59 时, 最远点包含在 $T'^{(29)}$, $T'^{(33)}$ 中的可能性为 95%, 时间复杂度降为 $O(|S_t^{(29)}|^2 + |T'^{(29)}||S_t^{(29)}|)$ 和 $O(|S_t^{(33)}|^2 + |T'^{(33)}||S_t^{(33)}|)$. 本文试验中, $|T'^{(29)}| = |T'^{(33)}| = 59$.

4) 在 Step 1.3 和 Step 2.3 中找到离中心点 $c_t^{(29)}$, $c_t^{(33)}$ 最远的点 $z^{(29*)}$, $z^{(33*)}$, 在扩展的特征空间中任意点 z 到中心点 c_t 的距离公式为

$$\left\| \begin{bmatrix} \varphi(z_l) \\ \delta_l \end{bmatrix} - \begin{bmatrix} c_t \\ 0 \end{bmatrix} \right\|^2 = \left\| \varphi(z_l) - c_t \right\|^2 + \delta_l^2.$$

其中

$$\delta_l^{(29)^2} = \varsigma - \widetilde{K}_1(z_l^{(29)}, z_l^{(29)}),$$

$$\delta_l^{(33)^2} = \varsigma - \widetilde{K}_2(z_l^{(33)}, z_l^{(33)}).$$

5) Step 1.4 中, 新的 CCMEB($S_{t+1}^{(29)}$) 可通过式 (29) 的 QP 问题求得; Step 2.4 中, 新的 CCMEB($S_{t+1}^{(33)}$) 可通过式 (33) 的 QP 问题求得. 这两个 QP 问题都是用 SMO 算法求得, 核心集规模远小于样本总体规模, 解决子问题的时间复杂度远小于解决所有样本 QP 问题的时间复杂度.

3.2 CSAL-CVM 算法时间复杂度和空间复杂度

CSAL-CVM 算法是基于 CCMEB 近似算法的一个特例, 其系统开销可参考 CVM的 计算复杂度[11-23], 并具有以下性质:

1) 对于给定的 MEB 近似误差 ε, 由 CSAL-CVM 算法求得的核心集数量的上界为 $O(1/\varepsilon)$, 算法迭代次数的上界为 $O(1/\varepsilon)$.

2) 对于给定的误差 ε, CSAL-CVM 算法的时间复杂度的上界为 $O(N/\varepsilon^2 + 1/\varepsilon^4)$.

3) 对于给定的 ε, 其空间复杂度上界为 $O(1/\varepsilon^2)$, 可使用存储核心集代替所有样本.

性质 1) 给出了在最坏情况下的算法迭代次数; 性质 2) 给出了在最坏情况下的算法运行时间, 与数据集的容量 N 呈线性关系; 性质 3) 给出了在最坏情况下算法的存储空间要求. 事实上, 在实验中发现, 面对大数据集时, 算法实际迭代次数、运行时间和存储空间要求远低于理论最坏值, 这也说明了 CSAL-CVM 算法对大数据集的处理是非常有效的.

3.3 算法收敛问题说明

CSAL-CVM 算法与 CSAL-MEB 算法求解策略一致, 亦采用了交替迭代法, 并且其优化过程也通过两个二次规划计算得到, 因此其收敛性与 CSAL-MEB 算法原理一致, 在此不再赘述.

4 实验结果与分析

本节将对本文所提出算法 CSAL-MEB 及其快速算法 CSAL-CVM 进行验证. 首先考察引入分类器分类间隔参数 ρ 后对推荐性能的影响; 然后考察快速算法 CSAL-CVM 的推荐性能及其在处理大样本数据时的运行速度.

实验环境: 操作系统为 Windows7, 内存为 2 GB, 主要软件 Matlab R2009a.

4.1 实验所用方法和数据集

实验所用方法见表 2. 实验采用电影推荐系统的基准数据集 MovieLens100K(http://www.grouplens.org.MovieLens 100 K), 该数据集包括 943 个用户 1 682 部电影的 10 万条评分记录, 每个用户至少对 20 部电影评分. 在实验预处理阶段, 采用文献 [10] 使用的余弦核函数分别计算出用户和电影的余弦核; 20% 的数据作为训练数据, 80% 的数据作为测试数据, 采用五重交叉验证方法最后得到的平均值作为最终性能结果; 若训练数据中有 v 条评分记录, 则顺序约束对数目为 $v \times (v-1)/2$ 个, 即 $|T| = v \times (v-1)/2$. 因此, 使用一般方法对 CSAL 方法、CSAL-MEB 方法求解二次规划时, 当 T 中的顺序约束对数目超过一定数量时, 发生内存溢出. 为避免发生内存溢出现象, T 中的顺序约束对随机选择. 具体地说, 首先选择用户 x_i, 然后选择该用户评分的两个不同的电影 y_s 和 y_t 组成一组顺序约束对 $(x_i, y_s, y_t, h_{i,s,t})$, 若用户 x_i 对电影 y_s 的评分大于对电影 y_t 的评分, 则 $h_{i,s,t} = 1$, 反之, $h_{i,s,t} = -1$. CSAL-CVM 算法中参数 ς_1, ς_2 取为 1 比较合适, 这样能满足 $\Delta_1 = \varsigma_1 1 - \mathrm{diag}(\widetilde{K}_1) \geqslant 0, \Delta_2 = \varsigma_2 1 - \mathrm{diag}(\widetilde{K}_2) \geqslant 0$ 的条件. 如果 ς_1, ς_2 取值过小, 则满足不了上述条件; 如果 ς_1, ς_2 取值过大, 则算法收敛速度受到影响. 通过大量实验表明, 取值为 1 效果较好. 3 个算法中参数 λ 取值均通过网格搜索的方法在 $\{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50\}$ 中寻优. 在算法推荐性能评估中, 采用流行的 normalized discounted cumulative gain(NDCG)[25] 作为评估准则. 对于每一个用户求得 NDCG@k($k = 1, 2, \cdots, 10$), 然后取平均值作为最终性能评价. 对于在测试数据中出现而在训练数据中未出现的新用户, 本实验忽略该用户, 这也是本文下一步要深入研究的问题之一.

表 2　本文实验中的方法

算法	所用数学模型及求解方法
CSAL	对式(4)用SMO[24]方法求解
CSAL-MEB	采用迭代寻优方法基于式(28)、(32)求解二次规划
CSAL-CVM	采用迭代寻优方法基于式(29)、(33)用快速算法求解

4.2 实验结果及分析

本文从 3 个算法的推荐性能 NDCG@k($k = 1, 2, \cdots, 10$) 和执行时间两方面进行实验比较. 计算 NDCG@k 时, 由于 CSAL 和 CSAL-MEB 两种算法样本容量超过 3 750 后, 在本文实验环境中出现“内存溢出”现象而无法继续执行, 为此在该实验部分本文选择样本容量大小为 3 750 的情况来分析各算法性能. 在训练数据集上运行 10 次, 取其平均值作为最终结果. 3 个算法的推荐性能 NDCG@k 如图 1 所示.

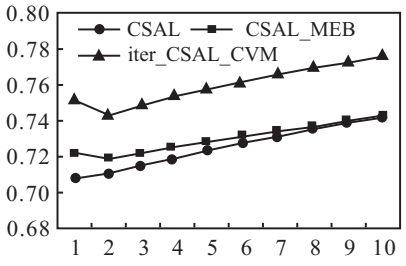

图 1　3 种算法在数据集 MovieLens100K 上推荐性能 NDCG@k($k = 1, 2, \cdots, 10$) 的变化比较

从图 1 可以看出, 本文算法 CSAL-MEB 的推荐性能优于原算法 CSAL, 这是由于本文算法引入了分类间隔器的概念; 近似算法 CSAL-CVM 的推荐性能优于另外两个算法, 这是由于该算法以大样本数据作为训练集, 而另外两个算法以原始样本中的小部分子集作为训练集. 根据经典的统计学理论, 数据量越多, 数据的真实分布越清晰.

本文对3种算法在数据集MovieLens100K上的求解执行时间进行了比较. 针对不同的训练数据量, 对每种算法进行了5次求解实验, 得到其平均求解时间, 求解执行时间如表3所示. 本文对3种算法在数据集MovieLens100K上训练样本数量逐渐增大时的推荐性能进行了比较. 针对不同的训练数据量, 对每种算法进行了5次求解实验, 最后得到其平均推荐性能NDCG@k, 实验结果如表4($k=1$)和表5($k=10$)所示. 表3～表5中"–"表示运行该方法时本实验环境中Matlab内存溢出, 算法无法继续执行. 从表3可以看出: 当样本数量少于1 000时, CSAL-MEB方法在问题求解时是可行的; 但当样本量达到10^3的数量级时, CSAL-CVM方法在求解速度上体现出了明显的优势. 值得说明的是, CSAL-CVM算法在本质上是一个依赖CVM技术特殊抽样的快速学习方法, 因而特别适合于大样本, 对于小样本计算时间变化不大. 文中所做的小样本实验, 说明在小样本情况下CSAL-CVM能够达到与CSAL、CSAL-MEB差不多的性能, 即在样本容量不超过5 000时, 核心集的数量相差不多, 因此其计算时间变化不大; 而对于样本容量为1 000时, 其核心集数量小于样本容量为750的核心集数量, 因此样本容量为1 000时的计算时间快于样本容量为750的执行时间, 这可能是抽样的随机性所致. 在本文Matlab实验环境中, 用CSAL方法和CSAL-MEB方法求解问题时处理能力不超过5 000个数据量; 而CSAL-CVM方法并无这个限制; 随着样本量逐渐增加, CSAL-CVM方法的求解时间上升较缓慢. 从表4和表5可以看出, 随着样本数量的增大, 3种算法的推荐性能逐渐提高, 且本文算法CSAL-MEB的推荐性能优于原算法CSAL, 近似算法CSAL-CVM的推荐性能优于另外两个算法, 这是由于CSAL-MEB算法引入了分类间隔的概念, CSAL-CVM算法以大样本数据作为训练集, 根据经典的统计学理论, 数据量越多, 数据的真实分布越清晰, 其推荐性能越高.

表3 各方法的求解时间对比表

样本容量	求解时间/s		
	CSAL	CSAL-MEB	CSAL-CVM
250	16.629 2	16.091 8	22.384 8
500	21.904 2	19.521 0	22.880 8
750	31.106 2	22.480 1	23.588 3
1 000	41.053 9	26.814 4	23.550 7
1 500	72.294 6	35.011 1	24.058 8
2 500	166.877 2	57.914 4	26.307 8
3 750	886.319 4	104.459 3	26.365 7
5 000	–	–	28.723 8
7 500	–	–	44.015 4
10 000	–	–	61.560 9
20 000	–	–	138.751 1
50 000	–	–	853.454 6
100 000	–	–	13 732.191 0
150 000	–	–	30 837.505 0

表4 各方法的推荐性能对比表(1)

样本容量	推荐性能($k=1$)		
	CSAL	CSAL-MEB	CSAL-CVM
250	0.693 4	0.707 9	0.736 0
500	0.695 7	0.709 4	0.737 1
750	0.698 6	0.710 7	0.746 9
1 000	0.701 4	0.713 2	0.747 7
1 500	0.703 8	0.717 9	0.748 2
2 500	0.705 9	0.720 1	0.749 9
3 750	0.707 2	0.721 5	0.751 3
5 000	–	–	0.753 8
7 500	–	–	0.758 2
10 000	–	–	0.760 9
20 000	–	–	0.766 8
50 000	–	–	0.771 3
100 000	–	–	0.776 3
150 000	–	–	0.781 9

表5 各方法的推荐性能对比表(2)

样本容量	推荐性能($k=10$)		
	CSAL	CSAL-MEB	CSAL-CVM
250	0.732 0	0.735 8	0.767 1
500	0.732 8	0.736 7	0.767 7
750	0.735 8	0.738 9	0.769 7
1 000	0.738 1	0.739 7	0.770 9
1 500	0.740 0	0.741 0	0.772 3
2 500	0.741 7	0.742 6	0.774 7
3 750	0.742 1	0.743 1	0.776 8
5 000	–	–	0.779 4
7 500	–	–	0.780 1
10 000	–	–	0.782 0
20 000	–	–	0.783 5
50 000	–	–	0.789 6
100 000	–	–	0.794 6
150 000	–	–	0.807 3

参数λ对本文算法CSAL-MEB和CSAL-CVM的推荐性能的影响如图2和图3所示. 从图2可以看出, 当λ取值为10时, 算法CSAL-MEB取得最优推荐性能; 从图3可以看出, 当λ取值为2时, 算法CSAL-CVM取得最优推荐性能.

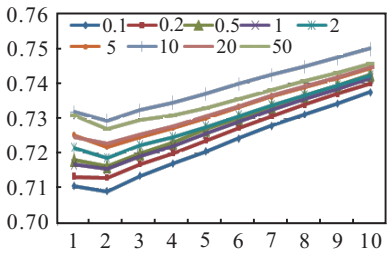

图2 参数λ对算法CSAL-MEB的推荐性能的影响

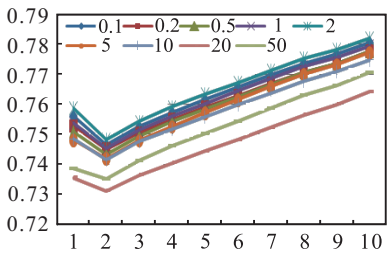

图3 参数λ对算法CSAL-CVM的推荐性能的影响

5 结　论

本文针对大规模跨空间异质数据的相似度学习问题, 提出了CSAL-MEB算法, 同时结合近似最小包含球理论, 提出了本文的中心算法CSAL-CVM, 并将该算法应用到推荐系统中. 本文方法吸收了CSAL的良好性能, 而CSAL-CVM算法良好的时间性能又使其在面对大数据集时仍能获得相对快速的决策. 实验中的推荐性能及快速性验证了本文方法的有效性. 当然, CSAL-CVM仍需对推荐系统面临的诸如"冷启动"等问题作进一步的研究, 即如何处理新用户和新项目问题, 这将是下一步研究的重点.

参考文献(References)

[1] Chen J, Ji S, Ceran B, et al. Learning subspace kernels for classification[C]. ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York, 2008: 106-114.

[2] Cristianini N, Shawe-Taylor J, Elisseeff A, et al. On kernel-target alignment[C]. Proc of NIPS. Cambridge: MIT Press, 2002: 367-373.

[3] Qamar A-M, Gaussier E, Chevallet J-P, et al. Similarity learning for nearest neighbor classification[C]. Proc of IEEE ICDM. Pisa, 2008: 983-988.

[4] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluating collaborative filtering recommender systems[J]. ACM Trans on Information Systems, 2004, 22(1): 5-53.

[5] Wang J, Vries A P D, Reinders M J T. Unifying user-based and item-based collaborative filtering approaches by similarity fusion[C]. Proc of SIGIR. New York: ACM Press, 2006: 501-508.

[6] Paterek A. Improving regularized singular value decomposition for collaborative filtering[C]. Proc of KDD Cup and Workshop. San Jose, 2007: 39-42.

[7] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model[C]. ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York, 2008: 426-434.

[8] Deshpande M, Karypis G. Item-based top-n recommendation algorithms[J]. ACM Trans on Information Systems, 2004, 22(1): 143-177.

[9] Weimer M, Karatzoglou A, Le Q V, et al. Cofirank-maximum margin matrix factorization for collaborative ranking[C]. Proc of NIPS. Cambridge: MIT Press, 2007: 1593-1600.

[10] Tang J H, Qi G J, Zhang L Y, et al. Cross-space affinity learning with its application to movie recommendation[J]. IEEE Trans on Knowledge and Data Engineering, 2011, 25(7): 1510-1519.

[11] Tsang I, Kwok J, Zurada J. Generalized core vector machines[J]. IEEE Trans on Neural Networks, 2006, 17(5): 1126-1139.

[12] Tsang I, Kwok J, Cheung P. Core vector machines: Fast SVM training on very large data sets[J]. J of Machine Learning Research, 2005, 6: 363-392.

[13] Deng Z H, Chung F L, Wang S T. FRSDE:Fast reduced set density estimator using minimal enclosing ball approximation[J]. Pattern Recognition ,2008, 41(4): 1363-1372.

[14] 钱鹏江, 王士同, 邓赵红. 基于最小包含球的大样本数据集快速谱聚类算法[J]. 电子学报, 2010, 38(9): 2035-2041.
(Qian P J, Wang S T, Deng Z H. Fast spectral clustering for large data sets using minimal enclosing ball[J]. Acta Electronica Sinica, 2010, 38(9): 2035-2041.)

[15] 胡文军, 王士同, 王娟, 等. 一般化最小包含球的大样本快速学习方法[J]. 自动化学报, 2012, 38(11): 1831-1840.
(Hu W J, Wang S T, Wang J, et al. Fast learning of generalized minimum enclosing ball for large datasets[J]. Acta Automatica Sinica, 2012, 38(11): 1831-1840.)

[16] Mitchell T. Machine learning, chapter computational learning theory[M]. McGraw-Hill,1997.

[17] Domeniconi C, Gunopulos D, Ma S, et al. Locally adaptive metrics for clustering high dimensional data[J]. Data Mining and Knowledge Discovery J, 2007, 14(1): 63-97.

[18] Wu K L, Yu J, Yang M S. A novel fuzzy clustering algorithm based on a fuzzy scatter matrix with optimality tests[J]. Pattern Recognition Letters, 2005, 26(5): 639-652.

[19] Yu J, Cheng Q S, Huang H K. Analysis of the weighting exponent in the FCM[J]. IEEE Trans on Systems, Man, and Cybernetics — Part B: Cybernetics, 2004, 34(1): 164-176.

[20] Yang S, Yan S, Zhang C, et al. Bilinear analysis for kernel selection and nonlinear feature extraction[J]. IEEE Trans on Neural Networks, 2007, 18(5): 1442-1452.

[21] Deng Z H, Choi K S, Chung F L, et al. Enhanced soft subspace clustering integrating within-cluster and between-cluster information[J]. Pattern Recognition, 2010, 43(3): 767-781.

[22] 蒋亦樟, 邓赵红, 王士同. ML型迁移学习模糊系统[J]. 自动化学报, 2012, 38(9): 1393-1409.
(Jiang Y Z, Deng Z H, Wang S T. Mamdani-Larsen type transfer learning fuzzy system[J]. Acta Automatica Sinica, 2012, 38(9): 1393-1409.)

[23] Badoiu M, Clarkson K L. Optimal core sets for balls computational geometry[J]. Theory and Applications, 2008, 40(1): 14-22.

[24] Platt J. Fast training of support vector machines using sequential minimal optimization[C]. Advances in Kernel Methods-Support Vector Learning. Cambridge: MIT Press, 2000: 185-208.

[25] Jarvelin, Kekalainen J. Ir evaluation methods for retrieving highly relevant documents[C]. Proc of SIGIR. New York, 2000: 41-48.

(责任编辑: 孙艺红)