

专题：统计物理和复杂系统

排名聚合算法在少量长列表聚合中的性能比较分析*

陈玟宇¹⁾ 朱章黔²⁾ 王晓蒙¹⁾ 贾韬^{1)†}

1) (西南大学计算机与信息科学学院 软件学院, 重庆 400715)

2) (中国人民解放军陆军勤务学院国防经济系, 重庆 500106)

(2019年10月17日收到; 2019年11月24日收到修改稿)

排名聚合将多个排名列表聚合成一个综合排名列表, 可应用于推荐系统、链路预测、元搜索、提案评选等. 当前已有工作从不同角度对不同排名聚合算法进行了综述、比较, 但存在算法种类较少、数据统计特性不清晰、评价指标不够合理等局限性. 不同排名聚合算法在提出时均声称优于已有算法, 但是用于比较的方法不同, 测试的数据不同, 应用的场景不同, 因此何种算法最能适应某一任务在很多情况下仍不甚清楚. 本文基于 Mallows 模型, 提出一套生成统计特性可控的不同类型的排名列表的算法, 使用一个可应用于不同类型排名列表的通用评价指标, 介绍 9 种排名聚合算法以及它们在聚合少量长列表时的表现. 结果发现启发式方法虽然简单, 但是在排名列表相似度较高、列表相对简单的情况下, 能够接近甚至超过一些优化类方法的结果; 列表中平局数量的增长会降低聚合排名的一致性并增加波动; 列表数量的增加对聚合效果的影响呈现非单调性. 整体而言, 基于距离优化的分支定界方法 (FAST) 优于其他各类算法, 在不同类型的排名列表中表现非常稳定, 能够很好地完成少量长列表的排名聚合.

关键词: 排名聚合, 不等长列表, Mallows 模型, 有偏等级重叠**PACS:** 02.50.-r, 89.75.-k, 87.15.Qt, 87.15.nr**DOI:** 10.7498/aps.69.20191584

1 引言

排序是在复杂系统的研究中经常使用的方法, 例如通过节点中心性对重要节点排序^[1,2], 链路预测中对可能存在的边排序^[3,4]. 在很多场景中, 我们会面对基于不同参数, 采用不同机制所得到的不同排名, 如不同人群对候选人给出的投票排名, 不同推荐系统给出的推荐排名, 不同算法给出的链路预测排名. 将多个排名列表聚合成一个综合排名列表, 就是排名聚合 (rank aggregation, RA) 要解决的问题. 排名聚合也称为 Kemeny 排名聚合^[5]、偏好聚合^[6]、共识排名问题^[7], 可以应用于推荐系统、

元搜索、期刊排名、提案评选、复杂网络等领域^[2,3,8-20]. 作为一个研究主题, 排名聚合已有超过两百多年的研究历史, 最早可追溯到 1781 年法国数学家 Borda^[21] 解决的法国科学院选举问题. 基于“总体大于各部分之和”的基本思想^[22], 不同研究领域提出了大量排名聚合方法, 大致可分为启发式方法和优化类方法. 启发式方法基于某种直观经验或规则将多个排名综合为一个排名, 具有简单、快速的特点. 优化类方法通过优化某一目标函数, 得到一个与已有排名之间存在某种优化量的排名. 虽然优化类方法在理论上更加完备, 但是因为其对应的优化问题是 NP (non-deterministic polynomial) 难的, 难以保证得到最优解. 对不同排

* 国家自然科学基金 (批准号: 61603309) 资助的课题.

† 通信作者. E-mail: tjia@swu.edu.cn

名聚合方法进行对比分析,能帮助在不同应用场景和数据条件下选择最合适的聚合方法,提高效率和可信度,具有重要的意义.

少量长列表的排名聚合是很多实际场景中需要解决的问题,这些排名列表相互不等长,且可能包含平局,例如不同机构给出的大学排名、基于不同基因数据库的靶点预测排名、不同搜索引擎对相关主题给出的推荐排名.选择何种算法能最好地处理少量长列表的排名聚合尚没有明确的答案.虽然当前已有综述类工作^[23,24]对不同聚合算法进行了列举和介绍,但是一般缺少不同方法之间的性能比较.一些工作^[25–30]虽然涉及了一部分排名聚合算法的性能比较,但是用于比较的聚合算法较少,使用的排序列表类型不够丰富,评价指标也有不够合理之处.同时,当前的绝大多数算法对比工作均是基于少量实证数据,由于数据的统计特征缺失,难以区分算法优劣是源自特定数据集,还是基于一系列数据集的共同统计特征,算法性能的泛化能力难以估测.最后,不同的聚合方法在提出时,均会声称优于已有的算法,但是用于比较基础算法不尽相同,用于测试的数据也往往大相径庭,使得我们往往无法真实知晓在具体应用场景下何种算法性能最佳.

针对这些局限性,本文基于 Mallows 模型提出一套用于生成统计特性可控的不同类型的排名列表的算法,使用一个可应用于不同类型排名列表的通用评价指标,介绍 9 种排名聚合算法,比较它们在少量长列表排名聚合下的表现.结果发现启发式方法虽然简单,但是在排名列表相似度较高、列表相对简单的情况下,能够接近甚至超过一些优化类方法的结果;列表中平局数量的增长会降低聚合排名的一致性并增加波动;列表数量的增加对各聚合结果的影响呈现非单调性.整体而言,基于距离优化的分支定界方法 (FAST) 优于其他各类算法,在不同类型的排名列表中表现非常稳定,能够很好地完成少量长列表的排名聚合.

2 基本概念

给定包含 $|S|$ 个对象的集合 S , 一个块序列 (bucket order) 是一种可传递二元关系 \prec , 存在块集合 $B_1, \dots, B_t (1 \leq t \leq |S|)$ 分割 S , 即块数量为 t . 如果 $x \in B_i$, 称 B_i 是 x 的块. 如果 $i < j$, 则称块 B_i

位于 B_j 之前. 对象 $x \prec y$ 当且仅当对于 $i < j$ 有 $x \in B_i$ 且 $y \in B_j$. 如果一个给定块中包含多个对象, 则形成一个平局, 即块内的对象具有相同的位置. 直观上看, 一个块序列就是一个可能包含平局的严格线性序列, 块中对象 x 的位置定义为 $p(x|x \in B_i) = \sum_{j < i} |B_j| + 1$, 即采用“1224”列表类型来表达对象的排名偏好信息, 而非采用块中对象的平均位置^[31]. “1224”列表类型表示为第 2 和第 3 个对象具有相同的位置, 形成一平局. 本工作主要针对少量长列表的聚合, 根据相关应用场景, 主要针对如下 3 种类别的列表^[32].

完全列表 (full list, FL): 块序列中所有块大小均为 1 且 $t = |S|$, 即包含 S 中所有对象且不含平局;

平局列表 (full list with ties, TL): 块序列中至少有一个块的大小大于 1 且 $1 \leq t \leq |S|$, 即包含 S 中所有对象且存在平局;

不完全列表 (incomplete list, IL): 由 S 中部分对象形成的块序列, 可能包含平局.

例如, 假设 S 包含 A, B, C 和 D 四个对象, 现有如下 5 个块序列:

块序列 1: [A], [B], [C], [D];

块序列 2: [A], [B, C], [D];

块序列 3: [A], [C];

块序列 4: [A], [B], [D];

块序列 5: [B], [C, A],

其中, 一个 “[]” 代表一个块, 即一个位置. 块序列 1 包含 S 中所有对象, 并且块数量等于 S 中对象数, 因此是 FL 类型. 块序列 2 包含 S 中所有对象, 但 B 与 C 都在块 2 中, 即 B 与 C 形成一平局, 是 TL 类型. 块序列 3—5 都只包含 S 中部分对象, 均为 IL 类型.

3 相关研究

Ali 和 Meila^[25] 重点探讨了不同排名聚合方法在搜索时间与结果表现之间的均衡性, 发现决定均衡的重要因素是数据集的一致程度 (degree of consensus). Schalekamp 和 Zuylen^[26] 也针对相似问题开展了工作, 并且将平局情况引入比较过程, 给出多个聚合方法表现的下界, 但是在计算过程中一些方法没有考虑打破平局的代价. 上述两个工作均使用真实数据集测试不同算法, 但是所使用数据量较小. Brancotte 等^[27] 针对平局比较了一系列排

名聚合方法,标注了不同方法是否适用于 TL 数据,研究了平局等数据集特征对所讨论方法的影响.但是以上几个工作都没有考虑更为复杂的 IL 数据.

Cohen-boulakia 等^[29]基于广义肯德尔距离,提出了一个新的启发式排名聚合算法,此方法可适用于所有列表类型,但是算法验证和对比仅仅基于一个小型生物医学数据集和人工数据(4—8 个对象).Lin^[24]从排名聚合方法在生物信息学中的应用角度,综述和对比了多个聚合算法,但是该工作使用的数据较小并且对 IL 数据的情况考虑不充分.Sculley^[33]比较了不同算法在大量短列表聚合下表现,但是其比较的方法较少,且均为启发式算法;Xiao 等^[30]基于其提出的数据生成模型对四种排名聚合算法进行了比较,并通过一个预设的真实排名(ground truth ranking)比较不同算法结果的质量.由于该数据生成模型对于平局数量、IL 数据生成过程不可控,同时真实排名在现实场景中难以获得,其算法质量评估在真实场景中的应用具有局限性.

在评价指标的选择上,大多数工作使用斯皮尔曼等级相关系数或肯德尔 τ 距离.这两个经典量只适用于排名列表包含所有对象的情况,不能应用在 IL 数据中,同时它们也没有考虑不同排名位置的不同权重.在真实场景中,靠前的排名应比靠后的排名具有更高的权重,例如第 1 名和第 2 名、第 50 名与 51 名之间均只相差 1 个排位,但前者的排名差距权重比后者更大.

综上所述,当前很多排名聚合算法比较工作存在数据量小、评估指标不够合理、数据统计特征缺失或列表类型单一等问题.针对以上问题,本文提出一个数据生成模型,生成列表类型、列表长度(即对象数量)、列表数量、一致程度等特征可控数据用于算法比较.本文同时采用更为合理的评价指标,更好地处理平局、排序权重等之前工作中未能充分考虑的问题,从而更合理地比较不同算法在不同情况下的表现.

4 排名聚合方法及其分类

排名聚合方法可根据相关特征从不同角度进行分类^[23,24,27,31,34].文献^[23]是最早对排名聚合方法进行分类的排名聚合公理派综述文章,将排名聚合方法分为启发式方法和优化类方法两类.文献^[24]将所有排名聚合方法分为基于分布的方法、启发式

方法和随机搜索方法三类.文献^[27]从能否处理平局的角度分为基于广义肯德尔距离的方法、基于传统肯德尔距离的方法和位置类方法三类.文献^[31]按是否需要训练数据分为监督类方法^[35,36]和非监督类方法两类.本文采用第一种分类标准.

4.1 启发式方法

4.1.1 KwikSort

KwikSort^[37]是一种分治算法,其核心思路是给定一组对象,随机选择一个对象作为中心点,并将其他所有对象按照一定规则置于该中心点前后的两个块中,从而使每一个对象与该中心点的违例数最少(为与下文排名聚合方法 MVR 中违例数相区分,此处违例数定义为对于两个对象,如果其在两个列表中的相对位置不一致,则形成一个违例).van Zuylen 和 Williamson^[38]提出了 KwikSort 去随机化版本.事实上,对象有一定概率与中心点形成平局.Brancotte 等^[27]通过这种策略对 KwikSort 进行了改进使其可以处理平局.

4.1.2 FaginSmall

Fagin 等^[31]针对平局提出几种指标并利用动态规划方法提出一种新的近似算法.Cohen-boulakia 等^[29]对上述方法做了修改得到 FaginLarge 和 FaginSmall,前者得到的结果包含平局,而后者不包含平局,因此,本文选用 FaginSmall.

4.1.3 BioConsert

给定 M 个对象和 N 个排名列表 $R_{\mu_1}, \dots, R_{\mu_N}$ (μ_1, \dots, μ_N 是每个列表对应的排名机制),通用目标函数为

$$R_C = \arg \min_{R_\mu} \left\{ \sum_{k=1}^N w_k d(R_\mu, R_{\mu_k}), R_\mu \in U \right\}, \quad (1)$$

其中 w 是权重向量,用于指定有关排名列表相对重要性或可靠性的先验信息.本文取为元素全为 1 的向量,以表征所有排名列表一样重要,此时该问题也称为 Kemeny 排名聚合问题.本文不考虑最终得到的聚合排名中包含平局的情况,所以 U 表示由这 M 个对象组成的所有 FL 集合,则 $|U| = M!$. d 是某种距离函数,如肯德尔 τ 距离^[39]、斯皮尔曼简捷距离^[40]、KS 距离^[23]以及 Hausdorff 距离^[41]等. R_C 在本文中叫聚合排名,而在社会选择和离散

数学文献中叫共识排名 (consensus ranking), 在统计学文献中称为中值排名 (median ranking).

BioConsert^[29] 的主要思路是从一个初始排名列表 R 开始, 通过不断迭代选择执行两类操作以减少上述通用目标函数值. 当任何操作均不能再减少目标函数值时, 则将此时的排名列表作为最终的聚合排名. 这两类操作是: 1) 将一个对象从原来的块中取出放入另一个块中; 2) 将一个对象从原来的块中取出并在另一个新的位置新建单对象块. 如果执行上述某一操作后目标函数变小, 则执行该操作; 反之, 则不执行. 不同初始排名列表 R 会产生不同的聚合排名结果, 极大影响算法的表现^[27]. 本文选用波达计数法的聚合排名 R_C 作为初始排名列表.

4.1.4 波达计数法 (BordaCount)

波达计数法^[21] 是最简单直观的排名聚合方法: 在每个排名列表中, 根据排名顺序对每个对象赋值一个分数, 将每个列表中该对象分数相加并对每个对象总分数进行排序就得到了最终的聚合排名. 同一对象在不同排名列表中的总分数除使用简单相加以外, 也可使用其他聚合函数, 例如中值函数、几何平均函数、 p 范数等^[24]. 本文采用最简单的波达计数法, 即某一列表中对象分数为该对象所击败的对象数量 (例如在长度为 M 的排列中, 第 1 名的分数为 $M - 1$, 而第 M 名的分数为 0), 并采用求和的方法计算各对象总分数.

4.1.5 MedRank

MedRank^[42] 的核心思路是给定阈值 $q \in [0, 1]$, 逐一并行读取所有 N 个排名列表, 一旦有对象出现次数首次超过 $N \times q$, 则根据对象出现的先后顺序依次添加至最终排名列表, 直至达到指定列表长度. 本文将阈值 q 取为 0.5.

4.1.6 马尔科夫链方法 (MC3)

Dwork 等^[11] 使用成对比较信息, 基于马尔科夫链提出了一系列的排名聚合方法, 其核心思路是将所有对象当作状态, 构建一各态历经的马尔科夫链转移矩阵, 从而其稳态分布会给予排在前面的状态更高的概率, 对象转移概率的不同赋值方法取决于我们的目标. Lin^[24] 对上述方法进行改进以使其可以适用于不同类型列表. 本文采用文献^[24] 中的 MC3 方法, 给定 M 个对象和 N 个排名列表

$R_{\mu_1}, \dots, R_{\mu_N}$ (μ_1, \dots, μ_N 是每个列表对应的排名机制), 对于 u, v 属于 M , 且 u 不等于 v , 则 $u \rightarrow v$ 的概率为

$$P(u \rightarrow v) = \frac{\sum_l^N I[R_{\mu_l}(u) > R_{\mu_l}(v)]}{N|M|}, \quad (2)$$

其中当 $I(\cdot)$ 所包含的条件满足时, $I(\cdot) = 1$, 否则 $I(\cdot) = 0$. 此时, 定义 $P(u \rightarrow u) = 1 - \sum_{v \neq u} P(u \rightarrow v)$. 同时, 对象转移概率与将待转向对象排在当前对象前面的列表数量成正比. 根据 (2) 式建立相应的状态之间的概率转移矩阵, 同时建立以状态为节点, 状态间转移概率作为边权重的加权有向图 $G(V, E)$, V 为状态集合, E 为边集合, 此时, 将问题转变为寻找最大的有向非循环连通子图.

4.1.7 PageRank

依据经典 PageRank 算法^[43] 的排名聚合算法 PageRank^[44] 的主要思想是给定排名列表构造图 $G(V, E)$, V 为节点集合, E 为边集合, 将每个对象看作图 G 中一个节点, 对于每一个排名, 如果对象 u 排名高于对象 v , 则建立一条加权有向边 $e(v, u)$, 其权重为所有给定排名列表的差值. 此外, 对所有权重进行归一化处理, 以便每个节点的出度边权重总和为 1. 对每个节点 u 构建 $\text{Pg}(u)$ 值作为排名得分, $\text{Pg}(u)$ 定义为

$$\text{Pg}(u) = (1 - \alpha)p_u + \alpha \sum_{e(v,u) \in E} \frac{\text{Pg}(v) \times w(u,v)}{\text{outdeg}(v)}, \quad (3)$$

其中 outdeg 是节点的出度, indeg 为节点的入度, $p_u = \frac{\text{indeg}(u)}{\sum_{v \in V} \text{indeg}(v)}$. 本文 α 取值为 0.85, 将算法给出的节点排序列表作为最终的聚合排名.

4.2 优化类方法

上述启发式方法尽管在运算速度上有优势, 但是并不能在理论上保证最终排名的性能最优性. 针对这一不足, 一些学者提出了优化类方法, 通过优化基于某一性能指标的目标函数, 获得聚合排名. 在衡量两个排名之间一致性情况下, 采用不同的性能指标 (如距离函数、等级相关系数和违例数等) 会得到不同的优化方法^[7,12,13,45-47]. 不同性能指标之间有时可相互转化, 比如在 FL 情况下, KS 距离和肯德尔 τ 距离完全等价^[48]; 此外, KS 距

离函数和 τ_x 等级相关系数是等价的, 具有线性变换关系 [45].

4.2.1 分支定界方法 (FAST)

肯德尔提出了 τ_a 和 τ_b 等级相关系数 [39], 前者适用于 FL, 而后者还适用于 TL. 因为 τ_b 在处理平局方面存在问题, Emond 和 Mason [45] 基于 τ_b 提出 τ_x 等级相关系数. 在一个包含 M 个对象的列表中, 定义分数矩阵 \mathbf{A} 为一个方阵, 对于任意两个对象 u 和 v , 如果 u 排在 v 前面或者与 v 形成平局, 则 $a_{uv} = 1$; 如果 u 排在 v 后面, 则 $a_{uv} = -1$; 如果 u 和 v 相等, 则 $a_{uv} = 0$, 即 \mathbf{A} 对角线上元素始终为 0. 因此, τ_b 在处理平局时将其分数矩阵对应元素置为 0, 而 τ_x 置为 1, 故而分数矩阵中除对角线以外的 0 元素表征无比较信息. 给定两个列表 R_{μ_1} 和 R_{μ_2} , τ_x 等级相关系数定义为

$$\tau_x(R_{\mu_1}, R_{\mu_2}) = \frac{\sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ij}}{M(M-1)}. \quad (4)$$

Emond 和 Mason [45] 基于 τ_x 提出一个分支定界算法来寻找聚合排名, 即寻找一个聚合排名 R_C 使平均加权 τ_x 等级相关系数最大 (或平均加权 KS 距离最小), 其目标函数为

$$\max \left[\sum_{k=1}^N w_k \tau_x(R_C, R^{(k)}) / \sum_{k=1}^N w_k \right].$$

注意: 此处的目标函数与通用目标函数 (1) 式是一致的, 因为 τ_x 与 KS 距离具有线性变换关系. 本文所有基础排名列表的权重都取 1, 以表征所有基础排名列表一样重要. 上述目标函数化简为

$$\max \sum_{i=1}^m \sum_{j=1}^m s_{ij} c_{ij}, \quad (5)$$

其中 $c_{ij} = \sum_{k=1}^n r_{ij}^{(k)}$. Amodio 等 [7] 基于 Emond 和 Mason 的分支定界算法提出 FAST 算法, 效果更好, 计算时间更短, 计算效率得到了很大提高, 是本文采用的方法.

4.2.2 最少违例数方法 (MVR)

在给定对象两两比较信息情况下, Pendings 等 [49] 利用 0—1 线性整数规划来寻找一个聚合排名以使对象不一致数量最少. 如果一个方阵每一行从左到右元素依次增大, 每一列从上至下依次减小, 那么该方阵就称为坡型矩阵. 对于一个方阵,

违反坡型结构的元素对数就是违例数 (注意: 此处违例数与 KwikSort 违例数 [27] 定义不一样). MVR 旨在尽可能寻找这样一个坡型结构, 从而使违例数最少. MVR 目标函数为

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \quad (6)$$

其中方阵 \mathbf{X} 是决策矩阵, 其值 $x_{ij} = 1$ 表示将 m_i 置于 m_j 前面, 否则置于后面, 并且需满足三个条件: 1) $x_{ij} \in \{0, 1\}$; 2) $x_{ij} + x_{ji} = 1$; 3) $x_{ij} + x_{jk} + x_{ki} \leq 2$. 方阵 \mathbf{C} 用于计算与坡型矩阵的违例数. 对任意 i 和 j , $c_{ij} = \#\{k | d_{ik} < d_{jk}\} + \#\{k | d_{ki} > d_{kj}\}$, 其中 d_{ij} 表示将 m_i 排在 m_j 前面的列表数量与将 m_j 排在 m_i 前面的列表数量的差值, 以衡量输入排名列表之间的一致程度. 值得注意的是, MVR 采用对象成对比较方式来表达偏好信息, 其最原始的应用场景是各类循环赛, 但在排名聚合背景下, 可将基础排名列表转换为两两比较信息. 如果出现平局或比较的两个对象至少有一个并未参与某一排名, 则不计算这两个对象在该排名中的得分, 因为未参与该排名, 故而无法得知孰强孰弱.

4.3 算法比较

本文主要基于 Mallows 模型生成的完全列表、包含平局的列表和不完全列表, 利用相似性指标来考察不同类型的列表数据对算法的影响. 从算法自身设计角度来说, 除少数算法以外, 多数都可以直接处理不同类型的列表. 例如, 除了 BioCount, MedRank 和 MVR 方法需要对列表作稍微调整外, 其余方法均可以处理包含平局的列表; 除 BioConsert, BordaCount 和 MedRank 需要对列表作稍微调整外, 其余方法均能处理 IL 数据.

以上 KwikSort, FaginSmall, BioConsert, BordaCount 和 MedRank 聚合算法使用文献 [27] 所提供的在线平台进行相关实验, MC3, PageRank, FAST 和 MVR 使用相关程序进行本地运算.

5 评价指标

为对比不同的排名聚合方法在不同列表类型上的表现, 需要一个通用评价指标来表征各排名之间的相似性. 一个合理的相似性度量指标需要能够处理对象未同时出现在排名中的情况, 即列表不等

长; 赋予高排名对象比低排名对象更多的权重; 同时相似度取值随着排名列表长度的增长而最终收敛. 在排名聚合领域, 有大量指标可用于计算列表之间相似性^[50–54], 但是很多都不能同时满足以上三个要求. 比如, 经常用于排名相似度计算的斯皮尔曼等级相关系数只能处理完全列表, 当列表不等长或列表元素存在不同时不再适用; 肯德尔 τ 距离以及广义肯德尔距离都可计算将一个排名转换为另一个排名所需的相邻对象交换次数, 但是却不能给排在前面的对象更高权重, 同时取值随着列表长度的增加而发散. Webber 等^[55] 基于简单概率用户模型实现了一个满足上述三个条件的指标——有偏等级重叠 (rank-biased overlap, RBO). RBO 通过在给定评估深度下计算一个基本分数 (下界) 和一个最大分数 (上界) 来提供单调性. 需要点估计时, 也可以计算出一个介于上下界之间的分数. $RBO \in [0, 1]$, 0 表示两个列表中对象完全不同, 1 表示两个列表包含相同对象且相对顺序一致. RBO 中包含一个参数 p , 其决定对象被加权的程度, 决定权重下降陡峭程度, p 越小, 指标对前面的对象加权越大. 本文根据文献^[55], 选取参数 $p = 0.9$.

为考察聚合算法的效能, 本文根据不同的列表类型, 使用 2 种指标. 对于 FL 数据, 由于列表生成模型基于一个中心排名均匀的产生随机样本, 因此可以认为中心排名即为最佳的聚合排名. 聚合排名与中心排名的差异性则体现出算法效能:

$$RBO = RBO(R_C, R_0), \quad (7)$$

其中 R_C 为聚合排名, R_0 为中心排名.

对于 TL 和 IL 数据, 由于在添加平局和构造不等长列表的过程中, 不可避免地使得列表不再基于中心排名列表随机分布, (7) 式不再适用. 因此我们使用聚合排名与原排名的平均相似度来度量聚合效果:

$$\overline{RBO} = \frac{1}{N} \sum_i^N RBO(R_i, R_C), \quad (8)$$

其中 $RBO(R_i, R_C)$ 代表原排名列表 R_i 与聚合排名列表 R_C 的相似度. 这一度量也同样适用于 FL 数据类型.

对于算法搜索时间比较方面, 由于各方法使用不同的平台、语言和优化工具, 本文不做比较.

6 数据生成模型

为评估和比较各种排名聚合方法, 首先需要生成具有不同统计特征的数据集, 包含多个相似但不相同的排名列表. TL 和 IL 数据都可作为 FL 数据的变体, 因此首先介绍 FL 数据的生成模型. 当前已有多种 FL 类型数据生成模型^[56–62], 本文选用理论和应用研究中应用广泛的 Mallows 模型^[56,57], 以更好地分析数据一致程度对排名聚合方法表现的影响. Mallows 模型是一个基于排名列表之间距离的指数模型, 包含两个参数: 中心排名 R_0 和离散分布参数 θ . Mallows 模型会给每一个排列赋予一个概率值

$$P(R|\theta, R_0) \propto e^{-\theta d(R, R_0)}, \quad (9)$$

其中 d 代表某种距离函数, 如肯德尔 τ 距离、海明距离、Cayley 距离和 Ulam 距离等^[58]. 本文使用肯德尔 τ 距离. θ 控制生成的排名与中心排名的距离. 当 $\theta = 0$ 时, 生成的排名 R 与 R_0 无关; θ 越大, 概率衰减越快, 生成的排名 R 越集中于 R_0 附近.

(9) 式虽然理论上非常简洁, 但是在实际操作中却存在难度, 一方面归一化常数的解析解难以获得, 另一方面也很难直接通过距离随机地生成一个排名. 因此在生成随机排名样本的过程中, 我们具体使用公式

$$P(R|\theta, n, d(R, R_0) = d) \propto S(n, d)e^{-\theta d}, \quad (10)$$

其中 $S(n, d)$ 为中心列表长度为 n 时, 所有与其距离为 d 的列表的数量. 在样本生成过程中, 首先基于中心排名, 穷举出所有相关的排名列表组合, 获得 $S(n, d)$ (这一步骤通常为 $O(n^3)$ 的复杂度). 根据 (10) 式随机生成距离 d , 再从所有与中心排名距离为 d 的排名列表中随机选择一个排名. 由于 (10) 式中的 $S(n, d)$ 随 d 增长, $e^{-\theta d}$ 随 d 下降, 因此最终获得的随机样本, 与中心排名的距离应该满足一钟型分布, 即概率存在一个峰值, 并在峰值距离两端迅速下降.

由于在 Mallows 模型中使用肯德尔 τ 距离生成排名序列, 而在聚合效果的度量中使用 RBO 相似度, 为验证生成的排名序列在 RBO 度量下也具有同样的统计特性, 基于不同的参数 θ 生成 3 组序列, 计算其与中心排名的相似度 RBO (图 1), 获得了预期中的钟形分布, 随着参数 θ 的增长, 生成的随机排名与中心排名相似度逐渐增加.

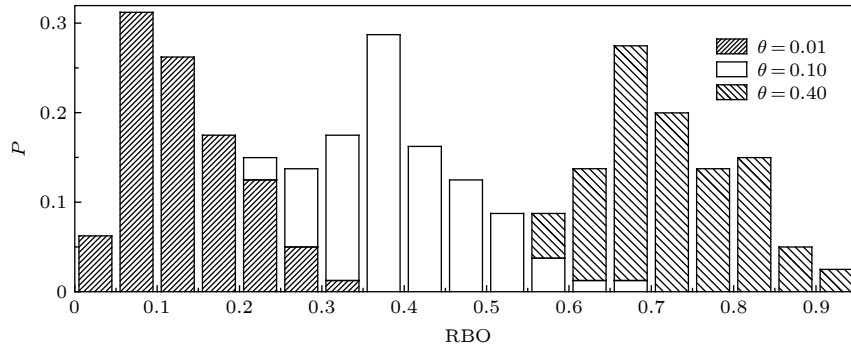


图 1 Mallows 模型在 RBO 度量下的表现
Fig. 1. The Mallows model under the RBO metric.

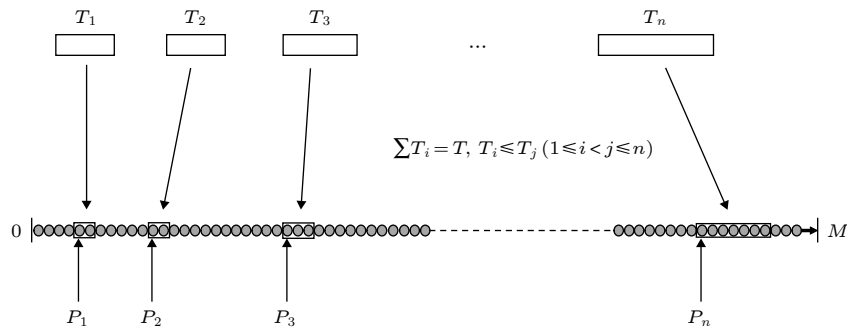


图 2 FL2 TL 示意图
Fig. 2. An illustration of FL2TL.

基于 Mallows 模型生成的 FL 数据, 本文提出两种方法将其转换为 TL 和 IL 类型数据.

1) FL2TL(r_t)

由 (10) 式获得长度为 M 的 FL 后, 生成一随机变量 $T \sim U[0, r_t * M]$, 作为列表中的平局总量, 其中 r_t 定义为平局比例, 取值范围为 $[0, 1]$. 令 $T' = T$, 在 2 至 T' 之间生成一随机数 t , 将一平局子块的长度设为 $\min(t, T' - t)$. 令 $T' = \max(t, T' - t)$, 重复以上步骤, 直到 $T' \leq 3$. 假设总共获得了 n 个和为 T 的随机数, 代表着对总长度为 T 的平局块的随机划分. 将 n 个随机数由小到大排序, 获得平局子块长度序列 $\{T_1, T_2, T_3, \dots, T_n\}$. 在排序列表中随机选择 n 个对象, 按照排名先后获得对象序列 $\{P_1, P_2, P_3, \dots, P_n\}$. 结合序列 T_i 与 P_i , 将序列中排在 P_i 到 $P_i + T_i - 1$ 范围内的对象设置为平局, 若两个平局之间存在相同的对象, 则合并这两个平局为一大平局. 整个过程如图 2 所示.

此方法在 FL 数据中加入了位置随机、长度随机的平局块, 并且兼顾了经验规律: 排名位置靠后的对象更容易形成包含对象数量较多的平局. 同时此方法只有一个参数, 便于进行相关分析. 为验证模型的可行性, 我们基于不同的参数 θ 和 r_t 生成

100 个排序, 计算了平均平局对 (平均的总个数) 和平局块 (平局集中出现在多少分块中) 的数量 (图 3), 发现通过控制 r_t 可以很好地控制平局对与平均块的数量. 作为验证, 我们也使用了一些多参数的复杂模型, 控制平局块长度和位置, 发现不同的算法并不改变本文所获得的聚合算法性能的整体结论.

2) TL2IL($r_k, \Delta k$)

考虑到真实数据中, 不同排名列表对靠前的排名个体差异并不大, 因此使用前端截取的方法生成不完全列表. 由 1) 中获得长度为 M 的 TL 数据后, 可以从列表前端截取一个长度为 L 的列表获得 IL 数据, 其中随机变量 $L \sim U[r_k * M - \Delta k, r_k * M + \Delta k]$, $r_k \in [0, 1]$ 定义为列表长度比例. 参数 r_k 控制 IL 数据的平均长度, 参数 Δk 控制列表长度的波动区间.

7 实验与讨论

本文主要讨论少量不等长列表的聚合情况, 如无特殊说明, 则使用列表长度 $M = 100$, 列表数量 $N = 20$, 初始数据一致程度 $\theta = 0.7$, $r_t = 0.2$, $r_k = 0.8$ 和 $\Delta k = 0.2$ 为参数. 本文对数据生成过程、聚合排名算法实现以及评价指标重复进行

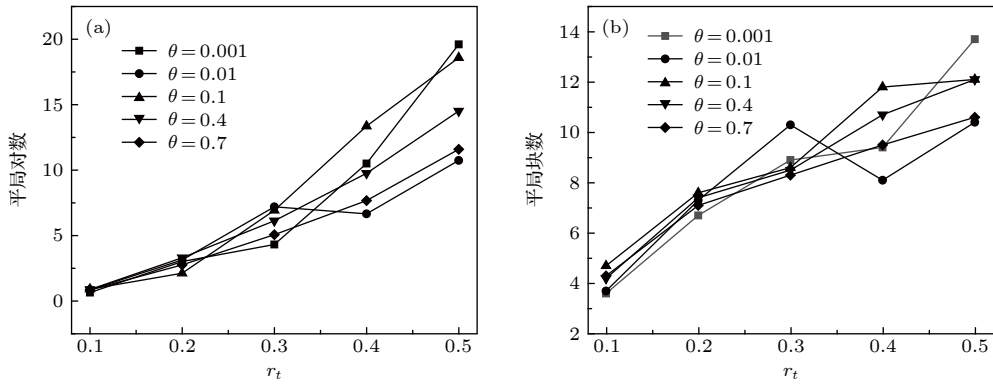


图 3 (a) r_t 与平均对数量的关系; (b) r_t 与平均块数量的关系

Fig. 3. (a) The relationship between r_t and the number of ties; (b) the relationship between r_t and the block number of ties.

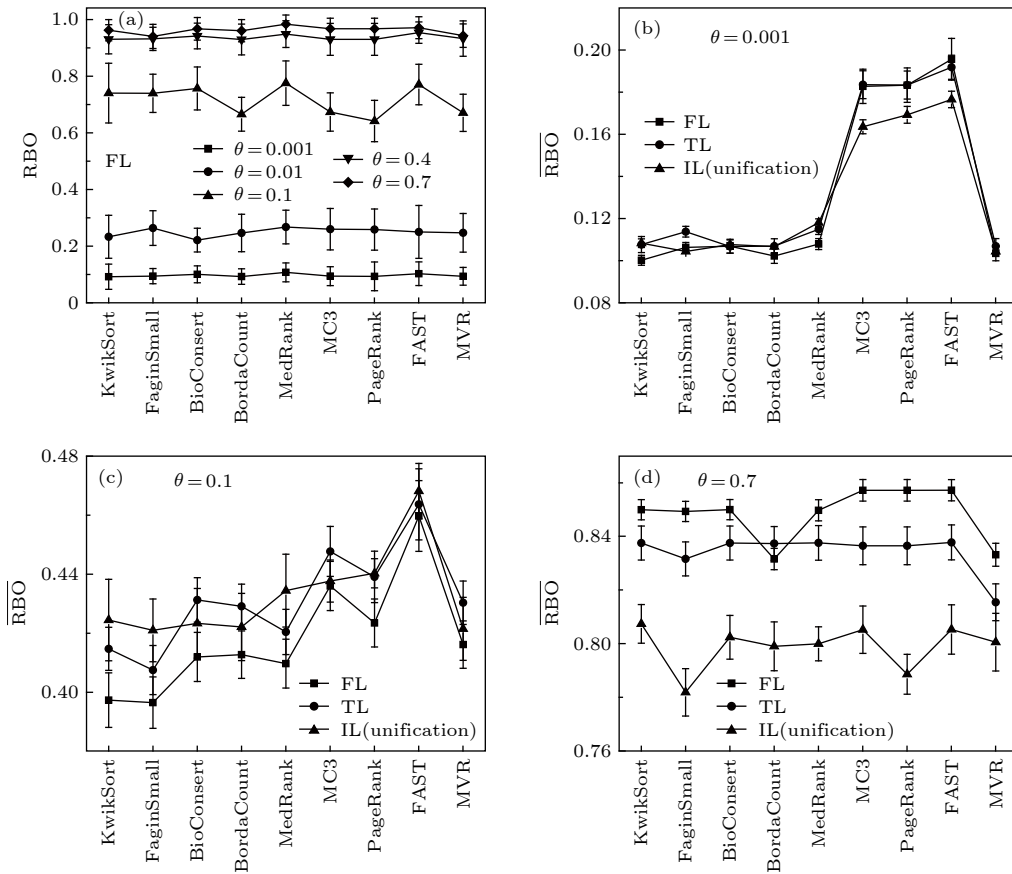


图 4 θ 值对算法表现的影响

Fig. 4. Impact of the degree of consistency θ on the algorithm performance.

10 次实验, 以确保结果更稳定. 由于 Xiao 等^[30] 探究了在不同 r_k 值下, 列表长度不等对排名聚合方法表现的影响, 故本文不再考察 r_k 和 Δk 组合对聚合方法的影响. 对于个别不能直接处理 IL 数据的算法, 我们将 IL 数据进行了规范化 (unification) 处理, 将其变换为算法可以处理的数据类型, 再将结果与其他算法结果对比. 在 7.4 节中将具体讨论规范化处理对结果的影响.

7.1 θ 值对各方法表现的影响

整体而言, 算法的表现主要由数据的发散程度决定. 初始数据一致程度越高 (θ 越大), 列表的发散程度越低, 最终的聚合效果也越好 (图 4). 对于 FL 数据类型, 如以聚合排名与中心排名的距离作为衡量标准 (图 4(a)), 各算法之间的表现差异较大, 即使在排名列表一致性较高的情况下, 启发式

算法也有可能不能给出与中心排名相似的结果. 优化类算法 MVR 更适用于例如循环比赛成绩综合等列表长度短、数量多的场景, 在少数长列表的情况下, 表现较差, 往往还不及启发式算法. 当 θ 过小时, 列表非常发散, 任何算法均无法获得与中心排名相似的聚合排名. 如以聚合排名与其他排名的平均距离 \overline{RBO} 作为衡量标准 (图 4(b)—图 4(d)), 在排名列表发散程度较低的情况下, 平局的引入 (由 FL 到 TL) 一定程度上减少了 \overline{RBO} , 但是在发散程度较高的情况下, 平局对于 \overline{RBO} 的影响较小. 同时, 对于 TL 和 IL 数据, 考察聚合排名与中心排名的 RBO 值时, 发现在不同的一致性程度下, 得到了与 \overline{RBO} 类似的结论, 这里只显示了 \overline{RBO} 的结果. 整体而言, 优化类方法 FAST 在不同参数 θ 下, 其表现均优于其他算法.

7.2 N 值对各方法表现的影响

为探究数据一致程度相同的情况下, 列表数量 N 对聚合效果的影响, 我们固定列表特征参数 ($M=100, \theta=0.7, r_t=0.2, r_k=0.8, \Delta k=0.2$), 改变

列表数量 N , 结果如图 5 所示.

对于 FL 数据, 聚合排名与中心排名的相似度基本随列表数量的增加而增加. 这与我们的直观感知一致, 数据量越大, 则生成的排名列表越均匀, 得到的综合排名也越接近于中心排名. 但也存在少数情况, 如波达计数法 (BordaCount) 与 MVR 表现随列表数量增加呈现不规则变化. 对于 TL 和 IL 数据, 聚合排名与中心排名的相似度与列表数量的关系与 FL 数据的结果类似, 不能较好地体现聚合算法的差异, 因此在考察 N 值对算法表现的影响时, 主要考虑聚合排名与其他排名的平均相似度 \overline{RBO} 的变化情况.

对于所有数据类型, 聚合排名与其他排名的平均相似度 \overline{RBO} 随列表数量的变化呈现不规则变化. 我们推测这是由于两方面的原因共同影响. 一方面列表数量的增加使得 N 个排名之间的差异性呈现不规则变化, 即图 5(b)—(d) 中 N 个排名与聚合排名的方差与列表数量呈非单调的关系; 而另一方面列表数量的增加提供更多的数据点, 使得聚合排名更加稳定. 这两个相互竞争的因素导致了平均相似

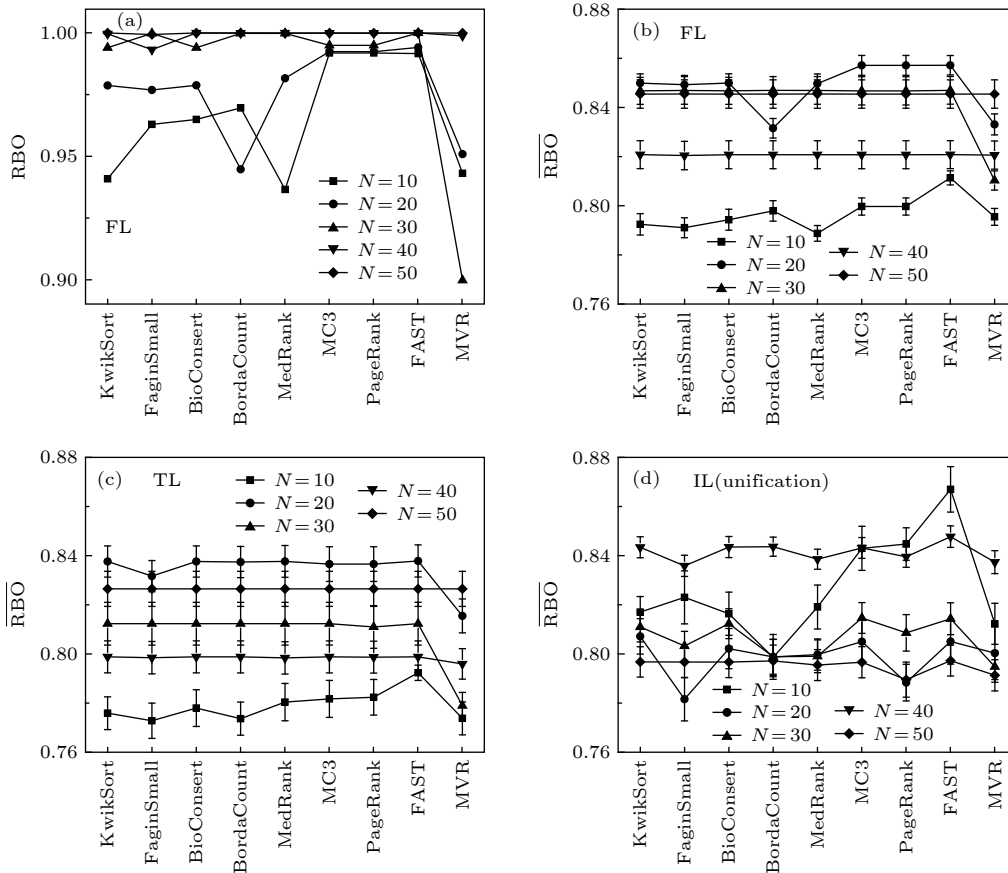


图 5 N 值对算法表现的影响

Fig. 5. Impact of the number of rank lists on the algorithm performance.

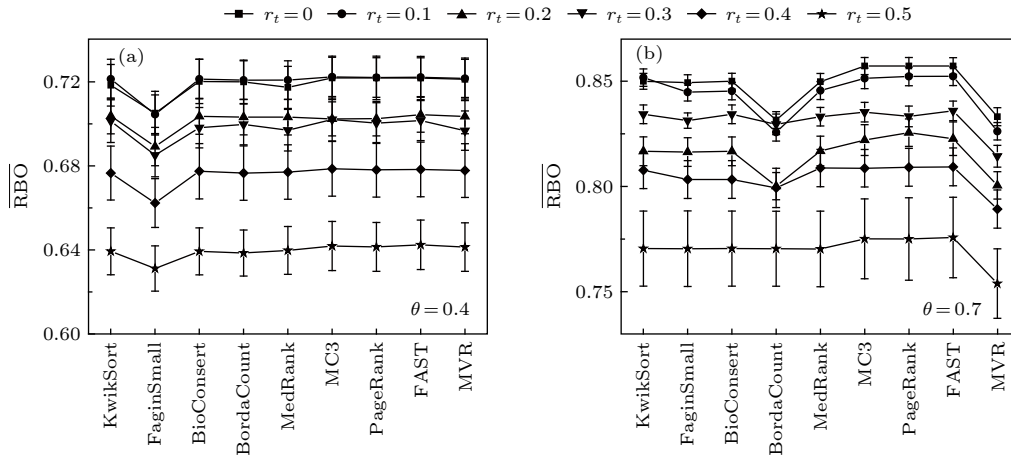


图 6 r_t 值对算法表现的影响
Fig. 6. Impact of the parameter r_t on algorithm performance.

度 \overline{RBO} 与列表数量的非单调关系。

同时, 对于所有列表类型, 随着列表数量的增加, 各聚合算法之间的相对差异也逐渐减少. 整体上优化类方法 FAST 表现也都较为稳定, \overline{RBO} 与 \overline{RBO} 表现普遍更好.

7.3 平局数量对各方法表现的影响

FL 数据和 TL 数据的区别在于 TL 包含平局. 在实际问题中, TL 比 FL 更常见. 我们调节参数 r_t 以分析平局数量对聚合效果的影响, 结果如图 6 所示.

图 6(a) 和图 6(b) 分别表示在 $\theta = 0.4$ 和 0.7 下各聚合方法的最终结果. 当 $r_t = 0$ 时, 对应数据类型是完全列表, 随着 r_t 逐渐增大, 平局块大小及其数量都在增加, 聚合排名与其他排名的平均相似度 \overline{RBO} 逐渐下降, 且波动性增加 (方差增大). 不同算法对平局情况的处理结果存在一定差异, 例如 FaginSmall 方法受整体序列一致性程度影响较大, 而波达计数法 (BordaCount) 在序列整体一致性较高但存在平局时的表现较弱, 考虑到波达计数法被广泛运用于简单的投票统计, 这也一定程度上说明为何一般投票中均没有平局.

7.4 Unification 规范化对算法表现的影响

对于一些不能直接处理 IL 类型数据的排名聚合算法, 如 MedRank, BordaCount, BioConsert 等, 可以通过对数据的规范化处理, 将 IL 数据变为 TL 数据, 使得这些算法可以发挥作用. 一般而言有两种规范化处理方法: Projection 和 unification. Projection 只考虑同时出现在所有列

表中的对象, 将未同时出现在所有给定列表中的对象从每个列表中删除. Unification 将未出现在本列表中的对象全部放到一个块中, 置于该列表底部, 形成一个多对象平局. 例如 2 节示例中的块序列 3—5 经过 unification 处理后变为:

块序列 3: [A], [C],

块序列 3 (unification): [A], [C], [B, D];

块序列 4: [A], [B], [D],

块序列 4 (unification): [A], [B], [D], [C];

块序列 5: [B], [C, A],

块序列 5 (unification): [B], [C, A], [D].

在 7.1 节与 7.2 节的性能比较中, 为了使所有算法均可以参与比较, 我们对排名列表进行了 unification 处理, 将不同算法应用到 unification 后的数据进行比较. 对于不能处理 IL 数据的算法, 利用 unification 处理数据是合理且必须的, 但是对于可以处理 IL 数据的算法, unification 可能会带来算法性能的高估或低估. 从另一个角度来看, 初始数据决定着可用信息量, 而经过规范化或简单推理以后, 实际上增加了一些人为规则信息, 必然会影响最终的聚合排名结果. 为探究 unification 规范化过程对算法表现的影响, 我们选用可以处理 IL 数据的优化类方法 FAST 和启发式方法 PageRank, 分析在处理前和处理后的算法性能变化.

为消除列表不等长对结果的影响, 令 $\Delta k = 0$, 改变参数 r_k , 比较 PageRank 和 FAST 不执行规范化和执行规范化的结果 (图 7). 结果发现, 聚合排名与其他排名的 \overline{RBO} 均值随列表长度呈不规则变化, 同时, 不执行规范化的结果总是优于执行规范化的结果, 这也说明 7.1 节和 7.2 节中的结果,

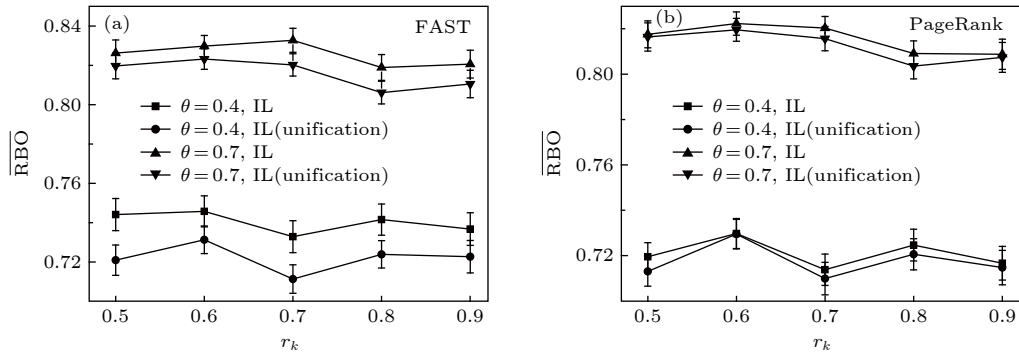


图7 Unification对算法表现的影响

Fig. 7. Impact of unification on algorithm performance.

一定程度上低估了如 FAST 这类可以直接处理 IL 数据的算法, 进一步说明了 FAST 算法在整体表现上的优越性. 与此同时, 尽管是否进行 unification 所带来的 \overline{RBO} 值差异不大, 但是对于需要更精细结果的应用而言, 谨慎处理初始数据和规范化也非常重要.

8 结 论

使用 Mallows 模型生成 FL 排名数据, 并提出一套 TL 和 IL 数据生成机制以生成人工可控的排名数据, 同时, 结合有偏等级重叠指标来探究数据特征对不同排名聚合方法的影响. 通过本文提出的数据生成机制, 可以生成具有不同统计特征的可控数据集. 实验结果表明, 启发式方法虽然简单, 但是在排序列表相似度较高、列表相对简单的情况下, 能够接近甚至超过一些优化类方法的结果, 例如在完全列表情况下, 波达计数法简单可行, 最终结果也基本可以接受. 整体而言, 基于距离优化的分支定界方法 (FAST) 优于其他各类算法, 在不同类型的排序列表中表现非常稳定, 能够很好地完成少量长列表的排名聚合.

参考文献

[1] Liao H, Mariani M S, Medo M, Zhang Y C, Zhou M Y 2017 *Phys. Rep.* **689** 154
 [2] Liu J G, Ren Z M, Guo Q, Wang B H 2013 *Acta Phys. Sin.* **62** 178901 (in Chinese) [刘建国, 任卓明, 郭强, 汪秉宏 2013 物理学报 **62** 178901]
 [3] Pujari M, Kanawati R 2012 *Proceedings of the 21st International Conference on World Wide Web* Lyon, France, April 16–20, 2012 p11
 [4] Tabourier L, Libert A S, Lambiotte R 2016 *EPJ Data Sci.* **5** 1
 [5] Snell J L, Kemeny J G 1962 *Mathematical Models in the Social Sciences* (Boston: Introduction to Higher Mathematics)

pp3–23
 [6] Davenport A J, Kalagnanam J 2004 *Conference on 19th National Conference on Artificial Intelligence* San Jose, USA, July 25–29, 2004 p697
 [7] Amodio S, D’ambrosio A, Siciliano R 2016 *Eur. J. Oper. Res.* **249** 667
 [8] Meila M, Phadnis K, Patterson A, Bilmes J 2012 arXiv: 1206.5265 [cs.LG]
 [9] Baskin J P, Krishnamurthi S 2009 *Proceedings of the Third ACM Conference on Recommender Systems* New York, USA, October 23–25, 2009 p337
 [10] Lü L Y, Medo M, Yeung C H, Zhang Y C, Zhang Z K, Zhou T 2012 *Phys. Rep.* **519** 1
 [11] Dwork C, Kumar R, Naor M 2001 *Proceedings of the 10th International Conference on World Wide Web* Hong Kong, May 1–5, 2001 p613
 [12] Cook W D, Raviv T A L, Richardson A J 2010 *Accounting Perspectives* **9** 217
 [13] Cook W D, Golany B, Penn M 2007 *Comput. Oper. Res.* **34** 954
 [14] Guo C H, Li M Q 2018 *Data Analysis and Knowledge Discovery* **2** 10 (in Chinese) [郭崇慧, 李敏谦 2018 数据分析与知识发现 **2** 10]
 [15] Jia T, Wang D, Szymanski B K 2017 *Nat. Hum. Behav.* **1** 0078
 [16] Zhang H X, LÜ Z, Zhang C T 2018 *Journal of University of Electronic Science and Technology of China* **47** 112 (in Chinese) [张海霞, 吕振, 张传亭 2018 电子科技大学学报 **47** 112]
 [17] Jia T, Xia F 2019 *Big Data Res.* **04** 38 (in Chinese) [贾韬, 夏锋 2019 大数据 **04** 38]
 [18] Wang X, Ran Y, Jia T 2020 *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30** 013101
 [19] Liu W, Wang Y B 2011 *Acta Phys. Sin.* **60** 070301 (in Chinese) [刘文, 王永滨 2011 物理学报 **60** 070301]
 [20] Han Z M, Wu Y, Tan X S, Duan D G, Yang W J 2015 *Acta Phys. Sin.* **64** 58902 (in Chinese) [韩忠明, 吴杨, 谭旭升, 段大高, 杨伟杰 2015 物理学报 **64** 58902]
 [21] Borda J C de 1781 *Histoire de l’Academie Royale des Sciences* 657
 [22] Langville A N, Meyer C D 2012 *Who’s # 1?: The Science of Rating and Ranking* (Princeton: Princeton University Press) pp159–231
 [23] Cook W D 2006 *Eur. J. Oper. Res.* **172** 369
 [24] Lin S 2010 *Wiley Interdiscip. Rev. Comput. Stat.* **2** 555
 [25] Ali A, Meila M 2012 *Math. Soc. Sci.* **64** 28
 [26] Schalekamp F, Zuylen A 2009 *Proceedings of the Eleventh*

- Workshop on Algorithm Engineering and Experiments* New York, USA, January 3, 2009 p38
- [27] Brancotte B, Yang B, Blin G, Cohen B S, Denise A, Hamel S 2015 *Proceedings of the VLDB Endowment* **8** 1202
- [28] Fagin R, Kumar R, Sivakumar D 2003 *SIAM J. Discrete Math.* **17** 134
- [29] Cohen-boulakia S, Denise A, Hamel S 2011 *International Conference on Scientific and Statistical Database Management* Portland, USA, July 20–22, 2011 p73
- [30] Xiao Y, Deng Y, Wu J 2017 *Nav. Res. Logist.* **64** 556
- [31] Fagin R, Kumar R, Mahdian M 2004 *Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* Paris, France, June 14–16, 2004 p47
- [32] Li X, Wang X, Xiao G 2017 *Brief. Bionform.* **20** 178
- [33] Sculley D 2007 *Proceedings of the 2007 SIAM International Conference on Data Mining* Minneapolis, Minnesota, USA, April 26–28, 2007 p587
- [34] Deng K, Han S, Li K J 2014 *J. Am. Stat. Assoc.* **109** 1023
- [35] Liu Y T, Liu T Y, Qin T 2007 *Proceedings of the 16th international conference on World Wide Web* Banff, Alberta, Canada, May 8–12, 2007 p481
- [36] Freund Y, Iyer R, Schapire R E 2003 *J. Mach. Learn. Res.* **4** 933
- [37] Ailon N, Charikar M, Newman A 2008 *J. ACM* **55** 23
- [38] Van Zuylen A, Williamson D P 2009 *Math. Oper. Res.* **34** 594
- [39] Kendall M G 1948 *Rank correlation methods* (London: Griffin)
- [40] Diaconis P, Graham R L 1977 *J. R. Stat. Soc. B* 262
- [41] Fagin R, Kumar R, Mahdian M, Sivakumar D, Vee E 2006 *SIAM J. Discrete. Math.* **20** 628
- [42] Fagin R, Kumar R, Sivakumar D 2003 *Proceedings of the 2003 ACM SIGMOD International Conference on Management of data* San Diego, California January 9–12, 2003 p301
- [43] Brin S, Page L 1998 *Comput. Networks ISDN Syst.* **30** 107
- [44] Adali S, Hill B, Magdon-Ismael M 2007 *J. Digital Information Management(JDIM)* **5** 292
- [45] Emond E J, Mason D W 2002 *J. Multi-Crit. Decis. Anal.* **11** 17
- [46] Ailon N 2010 *Algorithmica* **57** 284
- [47] Lin S, Ding J 2009 *Biometrics* **65** 9
- [48] Heiser W J, D’ambrosio A (edited by Lausen B, Dirk V P) 2013 *Algorithms from and for Nature and Life* (New York: Springer) pp19–31
- [49] Pedings K E, Langville A N, Yamamoto Y 2012 *Optim. Eng.* **13** 349
- [50] Bar-Ilan J, Mat-Hassan M, Levene M 2006 *Comput. Networks* **50** 1448
- [51] Lin Z W, Yi L, Guo X L 2017 arXiv: 1704.08464 [cs.AI]
- [52] Ekstrom C T, Gerds T A, Jensen A K 2018 *Biostatistics* **20** 582
- [53] Kumar R, Vassilvitskii S 2010 *Proceedings of the 19th International Conference on World Wide Web* Raleigh, North Carolina, USA, April 26–30, 2010 p571
- [54] Sakai T, Nicola F 2014 *Metrics, Statistics, Tests in: Bridging between Information Retrieval and Databases* (Heidelberg: Springer) pp116–163
- [55] Webber W, Moffat A, Zobel J 2010 *ACM T. Inform. Syst.* **28** 1
- [56] Mallows C L 1957 *Biometrika* **44** 141
- [57] Critchlow D E, Fligner M A, Verducci J S 1991 *J. Math. Psychol.* **35** 294
- [58] Irurozki E, Calvo B, Lozano J A 2016 *J. Stat Softw.* **71** 1575
- [59] Fligner M A, Verducci J S 1986 *J. R. Stat. Soc. B* 359
- [60] Smith B B 1950 *J. R. Stat. Soc. B* 12 41
- [61] Luce R D 2012 *Individual Choice Behavior: A Theoretical Analysis* (North Chelmsford: Courier Corporation) pp68–70
- [62] Thurstone L L 1927 *Psychol. Rev.* **34** 273

SPECIAL TOPIC—Statistical physics and complex systems

Comparison of performance of rank aggregation algorithms in aggregating a small number of long rank lists*

Chen Wen-Yu¹⁾ Zhu Zhang-Qian²⁾ Wang Xiao-Meng¹⁾ Jia Tao^{1)†}¹⁾ (*College of Computer & Information Science, Southwest University, Chongqing 400715, China*)²⁾ (*Department of National Defense Economy, Army Logistics University of Chinese People's**Liberation Army, Chongqing 500106, China*)

(Received 17 October 2019; revised manuscript received 24 November 2019)

Abstract

Rank aggregation aims to combine multiple rank lists into a single one, which has wide applications in recommender systems, link prediction, metasearch, proposal selection, and so on. Some existing studies have summarized and compared different rank aggregation algorithms. However, most of them cover only a few algorithms, the data used to test algorithms do not have a clear statistical property, and the metric used to quantify the aggregated results has certain limitations. Moreover, different algorithms all claim to be superior to existing ones when proposed, the baseline algorithms, the testing samples, and the application scenario are all different from case to case. Therefore, it is still unclear which algorithm is better for a particular task. Here we review nine rank aggregation algorithms and compare their performances in aggregating a small number of long rank lists. We assume an algorithm to generate different types of rank lists with known statistical properties and cause a more reliable metric to quantify the aggregation results. We find that despite the simplicity of heuristic algorithms, they work pretty well when the rank lists are full and have high similarities. In some cases, they can reach or even surpass the optimization-based algorithms in performance. The number of ties in the list will reduce the quality of the consensus rank and increase fluctuations. The quality of aggregated rank changes non-monotonically with the number of rank lists that need to be combined. Overall, the algorithm FAST outperforms all others in three different rank types, which can sufficiently complete the task of aggregating a small number of long rank lists.

Keywords: rank aggregation, incomplete rank list, Mallows model, rank-biased overlap**PACS:** 02.50.-r, 89.75.-k, 87.15.Qt, 87.15.nr**DOI:** 10.7498/aps.69.20191584

* Project supported by the National Natural Science Foundation of China (Grant No. 61603309).

† Corresponding author. E-mail: tjia@swu.edu.cn