

基于 Worker 权重差分进化与 Top- k 排序的结果汇聚算法

邢玉萍^{1,2}, 詹永照^{1,2}

(1. 江苏大学计算机科学与通信工程学院, 江苏 镇江 212013; 2. 江苏省工业网络安全技术重点实验室, 江苏 镇江 212013)

摘 要: 针对众包结果汇聚中最优排序结果选取的时效性问题, 提出了 Worker 权重的高效快速汇聚算法。其中 Worker 权重的差分进化算法重点考虑众包 Worker 完成排序任务存在的差异性问题, 基于目标函数和约束条件中 Worker 完成任务的不确定性和差异性影响, 建立基于差分进化算法的 Worker 权重优化模型, 获取多数据项场景下候选结果最优权重, 实现 Worker 权重与任务对结果性能需求匹配的最大化; 提出基于 Top- k 排序的优化模型求解算法, 针对多数据项场景下候选结果的 Top- k 排序选取, 在合适的 k 值下可快速求解上述模型, 获得各 Worker 的优化权重。所提出的基于优化的 Worker 权重可实现结果汇聚的匹配性与匹配速度优化, 即在提升结果汇聚速度的同时, 具有优化的汇聚结果性能。定性分析证明了算法的正确性, 仿真实验结果也验证了算法的效果, 与相关算法对比, 所提算法的综合性能最优。

关键词: 众包; 结果汇聚; 差分进化算法; 排序学习

中图分类号: TP393

文献标识码: A

DOI: 10.11959/j.issn.1000-436x.2021010

Result aggregation algorithm based on differential evolution and Top- k ranking in learning Worker's weight

XING Yuping^{1,2}, ZHAN Yongzhao^{1,2}

1. School of Computer Science and Telecommunications Engineering, Jiangsu University, Zhenjiang 212013, China

2. Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace, Zhenjiang 212013, China

Abstract: To solve the problem of quickly obtaining the optimal ranking result in the crowdsourcing result aggregation, an efficient and effective aggregation algorithm of Worker's weight was proposed. The Worker's weight optimization model based on differential evolution algorithm focused on the uncertainties and differences of Workers completing ranking tasks, the uncertainties and differences were reflected in the objective function and constraint conditions of the model. This model obtained the optimal weight of candidate results, and maximized the matching between Worker's weight and result performance. Then, the optimization model solving method based on Top- k ranking was proposed to quickly obtain the optimal Worker's weight with the appropriate k value for specific multi-data items ranking scenario. The optimization of Worker's weight could realize optimized performance and speed of the result aggregation. The correctness of the algorithm is verified by qualitative analysis, the effectiveness and efficiency of the algorithm is verified by the simulation results, and the comparison with the relevant algorithms shows the optimal comprehensive performance of the algorithm.

Keywords: crowdsourcing, result aggregation, differential evolution algorithm, learning to rank

收稿日期: 2020-08-19; 修回日期: 2020-11-13

基金项目: 国家重点研发计划基金资助项目 (No.2017YFB1400703); 中国博士后科学基金资助项目 (No.2019M651738); 国家自然科学基金资助项目 (No. 61702230)

Foundation Items: The National Key Research and Development Program of China (No.2017YFB1400703), China Postdoctoral Science Foundation (No.2019M651738), The National Natural Science Foundation of China (No.61702230)

1 引言

众包^[1]利用群体 Worker 的智慧解决问题,已成为数据处理的有力机制,特别是非结构化数据,如图像、视频和文本,出现了使用人工完成数据处理的任务,包括排序^[2]、聚类^[3]、最大值求解^[4]、过滤^[5]和去重^[6]等。

目前,传统的数据库管理系统和搜索引擎难以较好地完成排序任务,例如对全球大学排名排序、某领域权威论文排序,原因有以下几点。1) 相对封闭世界的假设,即数据都已存储在数据库里,不在数据库里的数据就是不存在的,并且数据库中缺少明确标注的相关信息,不能根据模糊标准进行匹配、排序或聚合结果。但是互联网环境下信息量急剧增加,未存储在数据库的数据并不意味着不存在,人们有能力在多个搜索引擎和参考资料等工具的帮助下,找到目前没有的信息。2) 缺乏对语义的理解,特别是排序任务,并不能很好地从数据库中将相关信息提取出来自动排序。但是,人们却比较擅长这些计算机很难或者不可能完成的任务。

快速获得高质量的解决方案是众包请求者的目的。完成任务的 Worker 主要来自普通大众,一般不具有提供高质量解决方案的特征,因此请求者通常采用任务冗余发放的算法,将相同任务发放给多个 Worker,然后对 Worker 的提交结果进行汇聚得到合适的解决方案。发布排序任务一般有 2 种形式: 1) 将任务拆分成微任务,以成对比较的形式进行分发,利用推理减少任务量,这种方式在任务量大时任务拆分困难、代价大、完成时间和完成质量难以保证^[7]; 2) 直接发布该排序任务,由不同 Worker 独立完成。由于不同 Worker 在不同专业领域可能具有不同水平,有的甚至无法完成某些任务,为了得到高质量的汇聚结果,需要从有噪声的答案中推断出高质量的结果。结果汇聚算法的优劣直接决定任务的完成质量,一般以二次众包或者统计算法自动完成。常用的基于统计的结果汇聚算法包括 Listwise 算法和数据融合算法,这 2 类算法都考虑了待排序对象的排名/位置信息。

Listwise 算法将一个任务下的所有数据项的排序结果列表作为一个训练样例,全面考虑一个任务下不同数据项之间的序列关系,优化目标是输出的汇聚结果和输入的提交结果损失函数最小或者性

能最优。文献[8]指出,合适的众包 Worker 约占 Worker 总数的 55%,其平均准确率约为 75%,因此,基于 Listwise 的结果汇聚算法容易受到不合格 Worker (如恶意 Worker 和搭便车 Worker^[9]) 提交结果的影响,从而降低汇聚结果的质量。Listwise 算法的模型复杂度和训练时间的长短依赖于待排序数据项的数量阶乘,训练复杂度很高^[2]。

数据融合算法的优化目标是汇聚结果的性能最大化。线性组合(LC, linear combination)算法是一类有监督的数据融合算法,其一般流程为: 首先多个 Worker 给出训练任务中待排序数据项的全排序结果,然后选择模型依据标注信息训练出所有 Worker 的优化权重向量,最后用该权重向量对测试排序任务进行结果汇聚。文献[10-11]分别使用遗传算法和差分进化算法获得优化的权重向量,但是这类基于演化计算的算法解决众包结果汇聚任务存在以下 2 个问题。

1) 即使是合格的 Worker,也很难给出所有数据项的准确全排序,排序结果会存在很多噪声,并且增加 Worker 的额外负担。实际应用中,Worker 能够很快给出 Top- k 数据项,例如,在图像检索任务中,给定 1 000 张餐馆的照片,希望找到最吸引人和最能描述该餐馆的 Top- k 照片。使用基于 Top- k 的排序可以降低任务难度,Worker 能在给出更准确的排序结果的同时,甄别出不合格的 Worker。如果 Worker 在训练任务的 Top- k 排序上是不合格的,那么该 Worker 在测试任务上依然是不合格的。

2) 基于演化计算的算法即使在排序列表很短的情况下,在解空间寻优依然是 NP-hard 问题,存在耗时过长的问题,不能有效应用于实时性要求较高的结果汇聚场合。相比全排序,基于 Top- k 能快速学习出各 Worker 的权重。

针对上述 2 个问题,本文提出基于 Worker 权重的差分进化和 Top- k 排序的结果汇聚算法。该算法更合理地考虑各个 Worker 的差异性和不确定性,且快速有效地提高汇聚结果的质量。这项研究具有较好的理论意义和现实价值。

本文主要的研究工作如下。

1) 建立众包排序任务结果汇聚的 Worker 权重优化模型,实现 Worker 权重与任务对结果性能需求匹配的最大化。针对多任务分配中众包 Worker 完成排序任务存在差异性,基于目标函数和约束条件中 Worker 完成任务的不确定性和差异性影响,建

立基于差分进化算法的 Worker 权重优化模型，获取多数据项场景下候选结果最优权重。该模型是一个非凸、非线性、多元优化问题，很难直接获得最优解。

2) 提出基于 Top-k 排序的优化模型求解算法。针对多数据项场景下候选结果的 Top-k 排序选取，在合适的 k 值下可快速对 1) 中模型求解，获得各 Worker 的优化权重。所提算法可实现结果汇聚的匹配性与匹配速度优化，即在提升结果汇聚速度的同时，具有优化的汇聚结果性能。通过定性分析证明算法的正确性。

3) 仿真实验结果表明，所提算法在大幅提升结果汇聚速度的同时，具有优化的汇聚结果性能。与相关算法对比，所提算法综合性能最优。

2 相关工作

近 20 多年来，数据融合在物联网^[12]、信息检索/Web 搜索、众包^[13]、推荐^[14]等多个不同的研究领域和应用中得到了广泛的研究和应用。研究者致力于数据融合算法的研究，提出了一系列算法，如无监督的 CombSUM^[15]和 CombMNZ^[15]算法，有监督的 LC 算法^[16]。权重分配是影响 LC 算法汇聚结果性能的最主要的因素。

文献[17]首先提出将 LC 算法用于文本数据融合，将各个提交结果的性能作为 LC 算法的权重，然而实验结果表明该权重分配策略并不优于使用相同权重的 CombSUM 算法。文献[18]使用共轭梯度优化各个提交结果的权重，最大化汇聚结果性能，该算法的不足在于非常耗时、仅能够汇聚 2~3 个提交结果、针对指定任务各个提交结果仅返回前 15 个数据项。文献[19-20]基于多元线性回归模型优化各个提交结果的权值，实现汇聚结果性能的最大化。文献[21]提出 ProbFuse 算法，该算法根据数据项在排序列表中的位置估计数据项的相关概率。文献[22]利用各个提交结果性能的幂函数作为权重分配策略优化汇聚结果的性能，并提出混合权重分配权重模式（权重是性能和差异性的乘积）优化汇聚结果的性能，该算法的不足在于数据项更新快的环境下效率较低。文献[23]提出基于神经网络的权重分配策略。文献[24]基于聚类算法提出 ClustFuse 算法。文献[25]基于期望最大化算法提出 MixModel 算法。文献[26]基于深度神经网络提出 Lambda-Merge 算法。文献[27]基于无监督的 RRF 和 Condorcet 算法提出具有低复杂度的融合算法。文献[28]基于数据

项的相似性，使用 CombSUM 和 CombMNZ 算法进行结果汇聚。

文献[10]提出 GA-Fusion，采用遗传算法在权重空间寻找各个提交结果的最优权值，实验结果表明，GA-Fusion 的性能优于前面所提的数据融合算法，包括 CombSUM、CombMNZ、Z-score、LC、LC2、多元线性回归、MixModel、ClustFuse、LambdaMerge 等。文献[11]采用基于差分进化算法寻找最优权值提高汇聚结果性能。文献[29]提出面向多样性任务基于性能、差异性和互补性的线性加权融合算法，实验结果表明所提算法并不优于 DE 和 GA 算法。

文献[10-11]采用基于演化学习的算法，通过不断地迭代优化权值提高汇聚结果的性能，然而，这种基于演化学习算法的不足在于求解时需要在解空间进行大规模的探索，运行时间很长，不能满足一些实时性要求高的应用场景的要求。本文研究众包排序任务实时结果汇聚问题，能够既快又好地获得汇聚结果。

3 系统模型和框架

LC 算法按各 Worker 提交结果的贡献度赋予权重，将 2 个或多个排序列表进行线性组合融合在一起，生成一个单一排序列表^[28]。因此，如果一个提交结果具有良好的性能，或者对最终汇聚结果性能的提升起到了重要作用，那么在结果汇聚时被赋予较大的权重；反之，则被赋予较小的权重。

假设有 m 个 Worker 节点，分别用 w_1, w_2, \dots, w_m 表示，对于给定任务 q ，各 Worker 节点 w_i 提交一个结果列表 τ_i^q ，对所有提交结果进行汇聚得到汇聚结果 π^q 。由于各个 Worker 排序具有不确定性，各个 Worker 的排序质量有好有差，因此引入衡量多 Worker 排序质量的权重 x_i ($1 \leq i \leq m$) 表示数据项的排序结果更确定的得分。在多个 Worker 参与的众包排序任务 q 中，对数据项 d_j^q ($1 \leq j \leq m_q$) 的融合排序应以多个 Worker 对数据项 d_j^q 的排序得分进行加权考虑，从而得到更确定的排序结果。因此数据项 d_j^q 融合排序得分可表示为

$$g(d_j^q) = \sum_{i=1}^m x_i s_i(d_j^q) \quad (1)$$

其中， $s_i(d_j^q)$ 表示数据项 d_j^q 在 τ_i^q 中的得分， $g(d_j^q)$ 表示数据项 d_j^q 的融合得分， π^q 中所有的数据项按照 $g(d_j^q)$ 降序排序。

以所有任务汇聚结果 $\pi = \{\pi^q | 1 \leq q \leq Q\}$ 的平均精度均值 (MAP, mean average precision) 性能最优原则进行融合排序优化, 建立目标优化函数为

$$\max_{\mathbf{X}} \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{M_q} \sum_{k=1}^{m_q} \frac{\sum_{t=1}^k I\{l_{\text{rk}(\mathbf{X}, D^q, t)} = 1\} I\{l_{\text{rk}(\mathbf{X}, D^q, k)} = 1\}}{m_q} \quad (2)$$

s.t. $\sum_{i=1}^m x_i = 1, x_i \in [0, 1], i = 1, 2, \dots, m$

其中, $\mathbf{X} = [x_1, x_2, \dots, x_m]^T$ 表示各个 Worker 的权重向量; $D = \{\tau_i | i \in [1, m]\}$ 表示各个 Worker 的提交结果集合, D^q 表示所有 Worker 第 q 个子任务的提交结果集合, $\text{rk}(\mathbf{X}, D^q, t)$ 函数返回汇聚结果排序为 t 的数据项, 汇聚结果排序以式(1)由 \mathbf{X} 对 D^q 得分加权融合得到; l_{d_i} 表示数据项 d_i 标注得分, 正确为 1, 否则为 0; $I\{\cdot\}$ 表示指示函数, $I\{\text{true}\}=1, I\{\text{false}\}=0$; M_q 表示任务 q 汇聚结果中相关数据项总数, m_q 表示该任务汇聚结果数据项总数, $|Q|$ 表示子任务 q 的数量。

式(2)的任务是找到权重向量 \mathbf{X}^* 使汇聚结果 π 的 MAP 性能最优, 由于 π 根据权重向量 \mathbf{X} 加权得分降序排序, 而优化目标使用 π 中数据项的位置信息, 因此该优化函数是一个非凸、非线性、多元优化问题, 无法直接对 \mathbf{X} 求导, 也不能使用基于梯度下降的相关算法求最优值, 因此, 采用基于差分进化算法 (DE, differential evolution) 和 Top- k 排序的结果汇聚算法计算式(2)的最优值, 将优化目标映射为 DE 算法的适应度函数 $f(\mathbf{X})$ 的最大值, 即

$$f(\mathbf{X}) = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{M_q} \sum_{k=1}^{m_q} \frac{\sum_{t=1}^k I\{l_{\text{rk}(\mathbf{X}, D^q, t)} = 1\} I\{l_{\text{rk}(\mathbf{X}, D^q, k)} = 1\}}{m_q} \quad (3)$$

s.t. $\sum_{i=1}^m x_i = 1, x_i \in [0, 1], i = 1, 2, \dots, m$

其中, m_q 的取值需依据 Top- k 中的 k 值设定。利用 DE 良好的寻优性能求解最优权重向量 \mathbf{X}^* 和最优汇聚结果 π 。

利用上述模型以及已有 m 个 Worker 在 $|Q|$ 个任务的提交结果, 按照式(2)的优化模型训练出各个 Worker 的权重, 根据式(1)得分降序排序获得汇聚结果。基于 DE 和 Top- k 排序的结果汇聚框架如图 1 所示。请求者将训练排序任务嵌入真实排序任务发布到众包平台, 平台将任务分配给多个 Worker 独立完成, Worker 努力工作后向平台提交结果。平台

收到所有提交结果后通过训练任务学习优化的 Worker 权重向量 \mathbf{X} , 将 \mathbf{X} 用于真实任务的结果汇聚, 得到优化的汇聚结果。平台向请求者提交汇聚结果后, 请求者向各个 Worker 支付相应报酬。

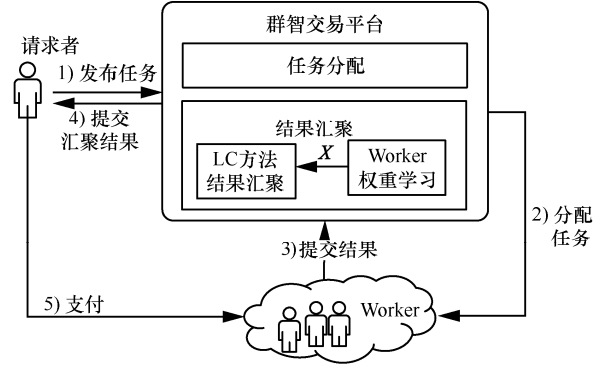


图 1 基于 DE 和 Top- k 排序的结果汇聚框架

4 算法设计

训练任务中使用差分进化算法基于 Top- k 排序相比全排序进行 Worker 权重学习优点明显, 其潜在假设是: 1) Top- k 排序上的训练和全排序上的训练一样好; 2) Top- k 排序上的运行速度相比全排序上的运行速度有显著提高。因此, 为了验证假设的正确性, 对 Top- k 排序和全排序的基于差分进化的结果汇聚算法进行理论和实验分析。在分析之前, 首先介绍基于 Worker 权重差分进化与 Top- k 排序的结果汇聚算法。

DE 是一种基于向量的适者生存、优胜劣汰算法^[30], 由 NP (种群规模) 个 m (参与汇聚的 Worker 人数) 维参数个体 \mathbf{X} (权重向量) 在搜索空间进行并行搜索, 基本操作包括变异、交叉和选择。假设工作者 $w_t (1 \leq t \leq m)$ 的提交结果为 τ_t , 则 τ_t 的 Top- k 排序表示为 $\tau_t^k = \langle d_{t,1}, d_{t,2}, \dots, d_{t,k} \rangle$ 。基于 Worker 权重差分进化和 Top- k 排序的结果汇聚算法 (简称 DE- k 算法) 如算法 1 所示。

算法 1 DE- k 算法

定义 $g = 0$, 给定 $\tau_t (1 \leq t \leq m)$ 和 k 。初始化 NP、 F (缩放因子)、CR (交叉概率)、 G (迭代次数)、 $P_0 = \{\mathbf{X}_{i,g} = (x_{i,1}, x_{i,2}, \dots, x_{i,m}) | i \in [1, \text{NP}], \sum_{j=1}^m x_{i,j} = 1 \wedge x_{i,j} = \text{rand}()\}$ (初始种群)

- 1) for ($i=0; i < \text{NP}; i++$)
- 2) 根据 $\mathbf{X}_{i,g}$ 和式(1)对 τ_t^k 进行结果汇聚, 得到汇聚结果 π_i
- 3) 根据式(3)计算 π_i 的适应度值, 即 $f(\mathbf{X}_{i,g})$

- 4) end for
- 5) $\mathbf{X}^* = \arg \max_{1 \leq i \leq \text{NP}} f(\mathbf{X}_{i,g})$
- 6) for ($g=1; g \leq G; g++$)
- 7) for ($\text{int } i=0; i < \text{NP}; i++$)
- 8) 随机选择 $r_1, r_2, r_3 \in [1, \text{NP}]$, 且 $r_1 \neq r_2 \neq r_3 \neq i$
- 9) $\mathbf{V}_{i,g+1} = \mathbf{X}_{r_1,g} + F(\mathbf{X}_{r_2,g} - \mathbf{X}_{r_3,g})$
- 10) for ($\text{int } j=1; j < m; j++$)
- 11) 随机选择 $j_{\text{rand}} \in [1, m]$
- 12) if ($\text{rand}() \leq \text{CR}$ or $j = j_{\text{rand}}$)
- 13) $\mathbf{U}_{i,j,g+1} = \mathbf{V}_{i,j,g+1}$
- 14) end if
- 15) end for
- 16) $\mathbf{X}_{i,g+1} = (f(\mathbf{U}_{i,g+1}) \geq f(\mathbf{X}_{i,g})) ? \mathbf{U}_{i,g+1} : \mathbf{X}_{i,g}$
- 17) 使用 Sum-to-1 算法归一化 $\mathbf{X}_{i,g+1}$ 中元素
- 18) $\mathbf{X}^* = (f(\mathbf{X}_{i,g+1}) \geq f(\mathbf{X}^*)) ? \mathbf{X}_{i,g+1} : \mathbf{X}^*$
- 19) end for
- 20) end for
- 21) 将 \mathbf{X}^* 代入式(1)对所有 $\tau_i (1 \leq i \leq m)$ 进行结果汇聚, 得到汇聚结果 π
- 22) 输出 \mathbf{X}^* 和 π

算法种群初始化操作中, 种群个体 $\mathbf{X}_{i,g}$ 在可行解空间内随机取值, 由于 LC 算法赋给各个 Worker 的权重区间为 $[0,1]$, 且参与汇聚的所有 Worker 的权重和为 1, 因此各 Worker 的权重初始取值为

$$\sum_{j=1}^m x_{i,j} = 1 \wedge x_{i,j} = \text{rand}() \quad (4)$$

其中, $x_{i,j}$ 表示个体 $\mathbf{X}_{i,g}$ 中第 j 个 Worker 的权重, $\text{rand}()$ 函数返回 $[0,1]$ 的随机数。步骤 1)~步骤 5) 根据随机生成的初始种群进行结果汇聚, 生成最优权重向量初值。步骤 8)~步骤 15) 对上一代种群中的每个个体执行差分变异和交叉操作。步骤 16) 为选择操作, 选择原个体向量 $\mathbf{X}_{i,g}$ 和实验个体向量 $\mathbf{U}_{i,g+1}$ 中适应度值大的个体进入下一代种群, 从而保证经过一次迭代后种群总体性能得到提升。为了保证解的有效性, 即在进化过程中保证每个个体中的元素满足 $\sum_{j=1}^m x_{i,j} = 1$ 。步骤 17) 对 $\mathbf{X}_{i,g+1}$ 中元素使用 Sum-to-1^[31] 算法进行归一化处理, 即

$$x_{i,j} = \frac{x_{i,j}}{\sum_{i=1}^m x_{i,j}} \quad (5)$$

步骤 18) 更新最优权重向量 \mathbf{X}^* 。步骤 21) 利用最优权重向量 \mathbf{X}^* 使用 LC 算法得到最优汇聚结果。

5 性能定性分析

5.1 汇聚结果性能分析

Top-k 学习在信息检索、信息过滤、物联网、信息安全、众包^[32]等领域均有广泛的应用。文献[33]从理论和实验 2 个方面验证基于 Top-k 排序能够获得和全排序同样的训练效果, 并指出随着 k 的增长, 测试算法的性能迅速增加到一个稳定的值, 例如, 当 $k=10$ 时, Ranking SVM、RankNet 和 ListMLE 这 3 种算法性能达到稳定; 当 $k=20$ 时, RankBoost 性能达到稳定。

文献[10]通过实验表明, Top-k 包含了用户最关注的信息, 仅用 Top-k 排序作为基于遗传算法的权重学习的训练集合, 与全排序作为训练集合相比性能并没有显著降低。此外, Pal 等^[34]发现, 如果每个任务仅选择 Top-k 数据项作为实现 TREC 评价池的依据, 那么评估的质量不会受到影响, 即排在 Top-k 的数据项含有丰富的信息, 较好地代表了长列表形式结果的性能。当使用 MAP 等位置相关的评价指标进行评估时, 这一点会更加突出, 并且, MAP 仍然是评价排序问题中重要的性能指标。

LC 算法按各个 Worker 提交结果的重要性赋予该 Worker 相应的权重, 并未更合理地考虑各个权重的进一步优化。所提模型考虑 Worker 完成排序任务存在不确定性和差异性, 以目标函数和约束条件中 Worker 完成任务的不确定性和差异性影响为基础, 建立基于 DE 算法和 Top-k 的 Worker 权重优化模型。该模型综合权衡各个 Worker 的提交结果, 利用 DE-k 算法迭代求解, 确定各个 Worker 的优化权重, 提高汇聚结果的性能, 使融合结果 MAP 性能最优, 因此 DE-k 算法求解后确定的优化权重向量能达到优化的汇聚结果。

综上所述, 通过 Top-k 排序包含的丰富信息和 DE 算法良好的寻优性能, DE-k 算法可实现优化的汇聚结果性能。

5.2 时间耗费与汇聚性能权衡分析

基于 DE-k 的结果汇聚算法中, 用 m 表示参与结果汇聚的 Worker 人数, k 表示提交结果列表的长度, NP 表示种群规模, G 表示迭代次数。该

算法的时间复杂度为 $O(mkNP)$ ，其与变量 m 、 k 、 NP 、 G 都成正比关系。因此，降低 m 、 k 、 NP 、 G 能够降低结果汇聚的运行时间，但是参与汇聚的 Worker 人数 m 和种群规模 NP 选定之后不再发生变化，一般采用对 DE 算法进行剪枝和降低 G 的迭代次数来减少结果汇聚的运行时间，而忽略了降低 k 减少权重向量的学习时间。全排序优化将花费大量的训练时间，而基于 Top- k 选择适当的 k 可加快速度。例如，Worker 提交结果 τ 的长度为 1 000，取 $k=10$ 参与权重的训练，则理论上运行速度提升约 $1\,000/10=100$ 倍，而汇聚结果性能接近高性能；若原提交结果列表越长，则结果汇聚运行时间降低越多，即 DE- k 算法可提升结果汇聚速度。

5.1 节和 5.2 节的定性分析证明了 DE- k 算法在获得优化的汇聚结果性能的同时可提升结果汇聚的速度。

6 实验结果分析

6.1 数据集及对比较算法

实验数据集为 TREC-10 和 TREC-11 的 Routing filtering Task 的提交结果集。TREC-10 Routing filtering (简称 TREC-10) 共有 84 个 Topic。TREC-11 Routing filtering 共有 100 个 Topic (前 50 个和后 50 个 Topic)，分别由专业和非专业评估人员评估，为了更准确地描述汇聚后结果的性能，将该结果集分成 2 个结果集，标记为 TREC-11A 和 TREC-11B，每个结果集分别有 50 个 Topic。

仿真数据集包括 TREC-10 的 13 个提交结果和 TREC-11 的 8 个提交结果。详细信息如表 1~表 3 所示，按照提交结果的 MAP 值降序排列。评价指标为 MAP、P@10 和召回率 (RP, recall precision)。

表 1 TREC-11A 提交结果性能

提交结果名称	MAP	P@10	RP
Iritsigr	0.369	0.626	0.412 2
msPUMb	0.355	0.64	0.374 8
apl11Frm	0.33	0.594	0.353 7
ICTRouFT11Ub	0.25	0.526	0.302
ICTRouFT11Ua	0.243	0.522	0.294
msPUMs	0.239	0.438	0.263 9
Visa1T11	0.236	0.514	0.280 1
apl11Frsvm	0.218	0.462	0.255 6

表 2 TREC-11B 提交结果性能

提交结果名称	MAP	P@10	RP
msPUMb	0.045	0.122	0.080 4
iritsigr	0.044	0.124	0.071 7
apl11Frsvm	0.043	0.12	0.077
apl11Frm	0.042	0.102	0.076 5
Visa1T11	0.028	0.098	0.051 5
ICTRouFT11Ub	0.025	0.074	0.047 2
ICTRouFT11Ua	0.024	0.072	0.047 2
msPUMs	0.006	0.018	0.011 3

表 3 TREC-10 提交结果性能

提交结果名称	MAP	P@10	RP
DLewis01rFa	0.168 2	0.875	0.198 5
DLewis01rUa	0.166 3	0.871 4	0.197 4
KUNr2	0.136 9	0.878 6	0.166 2
KUNr1	0.135 6	0.809 5	0.171 1
apl10frm	0.121	0.833 3	0.156 1
clT10rtb	0.104 8	0.772 6	0.144 4
oraRO082801	0.104	0.785 7	0.143 1
apl10frs	0.103 8	0.761 9	0.141
mer10r1	0.092	0.727 4	0.126 3
clT10rta	0.077 8	0.719	0.117
jscbtafr2	0.046 6	0.609 5	0.083 1
jscbtafr1	0.042	0.654 8	0.076 5
VisaSent1T10	0.011 8	0.377 4	0.0174

对比较算法为 CombSUM^[14]、CombMNZ^[14]、基于 DE 的结果汇聚算法^[11]、基于粒子群优化的结果汇聚算法、基于粒子群优化和 Top- k 排序的结果汇聚算法、基于遗传算法的结果汇聚算法^[10]、基于遗传算法和 Top- k 排序的结果汇聚算法^[10]。

6.2 实验结果和分析

下面，分别在 TREC-11A、TREC-11B 和 TREC-10 数据集对不同算法的汇聚结果进行分析和比较，并从 3 个方面验证所提算法在数据集上综合性能最优：相同迭代次数下不同算法结果汇聚的性能和运行时间的分析与对比、不同迭代次数下不同算法结果汇聚性能和运行时间的分析与对比，以及高性能系统的比较。

6.2.1 相同迭代次数下不同算法结果汇聚的性能和运行时间的分析与对比

本节实验中，训练在每个子任务的提交结果 Top- k 分别为 10、25、50、100 和 1 000 (全排序) 处进行，而测试在汇聚结果排序深度为 1 000 (全排序) 处进行。使用 5 折交叉验证，迭代次数都为

200。PSO- k 和 GA- k ($k \in \{10, 25, 50, 100\}$) 分别表示使用基于粒子群优化算法和 Top- k 排序的结果汇聚算法以及基于遗传算法和 Top- k 排序的结果汇聚算法。DE、PSO 和 GA 表示使用提交结果的全排序作为训练集, 分别基于差分进化算法、粒子群优化算法和遗传算法的结果汇聚算法。BR 表示所有 Worker 中 MAP 性能最优的提交结果。CombSUM 和 CombMNZ 分别表示使用 CombSUM 和 CombMNZ 算法进行结果汇聚。除了 BR、CombSUM 和 CombMNZ, 表 4 中的数据都是 8 次单独实验结果的平均值, 其中加粗数据表示该列最优的 3 个值。

表 4 数据表明, 以 MAP 为性能评价指标, 所有结果汇聚算法的性能都优于最优 Worker 的提交结果 (BR); DE- k 和 PSO- k 性能相似, 优于 GA、GA- k 、CombMNZ、CombSUM; DE- k 、PSO- k 和 GA- k 算法分别与 DE、PSO 和 GA 算法相比, 性能损失分别为 [0.98%, 4.01%]、[0.15%, 3.42%] 和 [0, 9.65%], 这些数据表明, 基于 Top- k 的结果汇聚相比全排序性能损失不明显; DE- k 性能优于 PSO- k 、GA- k , 具有较好的稳定性, 且在其他性能指标 P@10 和 RP 下也具有上述相似的结论。

运行时间包括数据装载、权重学习和结果汇聚的运行时间。表 5 中数据为表 4 中不同算法对应的

运行时间。以运行时间为评价指标, 全排序下, DE、PSO 和 GA 完成一次结果汇聚至少需要 3 h 以上, 且数据集越大, 运行时间越长, 不能满足实时结果汇聚的要求; DE 算法和 GA 算法运行时间相似, 远低于 PSO 算法的运行时间; CombSUM 算法和 CombMNZ 算法不需要权重学习, 运行时间最短。当 $k=10$ 时, 在 TREC11-A 和 TREC11-B 数据集中, DE-10、PSO-10 和 GA-10 在 51~64 s 内完成结果汇聚, 相比全排序速度提升 191~380 倍; 在 TREC-10 数据集中, 3 种算法分别在 16.4 min、19.7 min 和 17.6 min 完成结果汇聚, 相比全排序速度分别提升 20 倍、38 倍和 18 倍, 偏离预计的 100 倍左右 (1 000/10)。通过对 CombSUM 和 CombMNZ 算法运行时间分析发现, 这 2 种算法的运行时间分别是 3.1 min 和 6.5 min, 原因在于较大的数据集数据装载耗费时间较多。当 k 分别为 25、50 和 100 时, 具有相似的结论。实验结果表明, DE- k 运行时间相比 PSO- k 、GA- k 运行时间少; 随着 k 值的减小, 运行时间显著减少, 当 $k=10$ 时, DE-10 的运行时间在 3 个数据集上相比全排序分别降低了 98.6%、99.6%、95%, 大幅降低了运行时间。

上述实验结果表明, 基于 Top- k 排序的优化模型求解算法, 在合适的 k 值下可快速得到性能好的

表 4 汇聚结果的性能

算法	TREC-11A			TREC-11B			TREC-10		
	MAP	P@10	RP	MAP	P@10	RP	MAP	P@10	RP
BR	0.369	0.626	0.412 2	0.045	0.122	0.080 4	0.168 2	0.875	0.198 5
CombSUM	0.421 2	0.378	0.435 5	0.056 1	0.156	0.094 1	0.265 5	0.864 3	0.192 8
CombMNZ	0.425 4	0.373	0.436 7	0.058 9	0.154	0.097 9	0.277 8	0.865 5	0.198 9
GA	0.462 6	0.7	0.463	0.059 7	0.146 7	0.094 5	0.303 9	0.877 4	0.212 3
GA-100	0.460 6	0.701	0.461 4	0.057 3	0.144	0.092 8	0.302 2	0.873 8	0.211 9
GA-50	0.456 6	0.687 2	0.458	0.056 6	0.144 6	0.090 7	0.297 4	0.869	0.209 1
GA-25	0.462 6	0.698 6	0.462 5	0.055 2	0.145 8	0.088 6	0.300 6	0.882 4	0.210 5
GA-10	0.455 5	0.694 9	0.456 9	0.054 4	0.135	0.085 7	0.294 3	0.872 6	0.206 8
PSO	0.480 3	0.722	0.474 9	0.065 4	0.168	0.103 1	0.310 3	0.885 1	0.216 8
PSO-100	0.479 1	0.707	0.475 1	0.063 7	0.172	0.102 1	0.302 3	0.878 9	0.211 8
PSO-50	0.472 4	0.696	0.473 3	0.065 3	0.164	0.100 6	0.306 3	0.882	0.214 3
PSO-25	0.476 2	0.704	0.473 7	0.064 9	0.162	0.101 2	0.302 7	0.876 8	0.211 3
PSO-10	0.469 9	0.699 4	0.471 5	0.063 3	0.176	0.097 2	0.3	0.870 9	0.209 8
DE	0.481 9	0.720 7	0.474 8	0.066 3	0.173 3	0.101 5	0.312 2	0.886 1	0.217 8
DE-100	0.477 1	0.707 7	0.474	0.063 8	0.166 5	0.102 5	0.309 2	0.885 1	0.216 1
DE-50	0.474 5	0.705	0.474	0.064 4	0.164	0.099 4	0.307 6	0.876 2	0.215 1
DE-25	0.476	0.706	0.475 5	0.064 9	0.164	0.100 4	0.306 6	0.879 5	0.214 7
DE-10	0.472 4	0.711 2	0.474 4	0.063 7	0.172 7	0.095 7	0.306 3	0.883 7	0.214 3

汇聚结果，但是性能略有损失，原因在于使用较少的基本事实进行训练评估。其中，DE-*k* 算法从性能和运行时间两方面在 3 个数据集上具有显著的优势，综合性能最优。

6.2.2 不同迭代次数下不同算法结果汇聚的性能和运行时间的分析与对比

表 4 和表 5 中数据表明，基于 Top-*k* 排序是加快结果汇聚运行时间的有效算法，但是该算法的运行时间仍远高于不需要权重训练的 CombSUM 和 CombMNZ 算法的运行时间。

表 5 结果汇聚的运行时间

算法	TREC-11A	TREC-11B	TREC-10
CombSUM	1.14 s	1.49 s	3.1 min
CombMNZ	2.04 s	3.3 s	6.5 min
GA	3.1 h	3.53 h	5.35 min
GA-100	854 s	866 s	52.8 min
GA-50	380 s	400 s	35.2 min
GA-25	158 s	189 s	28 min
GA-10	58 s	58 s	17.6 min
PSO	3.4 h	6.75 h	12.4 h
PSO-100	717 s	796 s	41.6 min
PSO-50	322 s	370 s	35.1 min
PSO-25	170 s	172 s	19.9 min
PSO-10	53 s	64 s	19.7 min
DE	3 h	4.02 h	5.51 h
DE-100	678 s	862 s	37.9 min
DE-50	317 s	395 s	25.8 min
DE-25	147 s	184 s	19.8 min
DE-10	51 s	63 s	16.4 min

图 2~图 4 和表 6 中数据都是采用 5 折交叉验证、8 次实验的平均值，迭代次数分别为 25、50、100、150 和 200，观察迭代次数对不种算法的汇聚结果的性能和运行时间的影响。

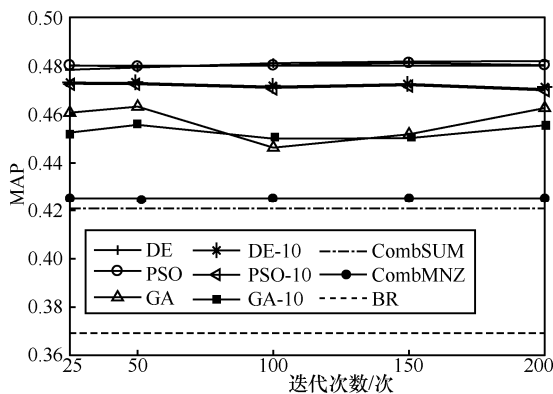


图 2 TREC-11A 中不同算法在不同迭代次数下汇聚结果性能

图 2~图 4 表明，①不同迭代次数下，以 MAP 为性能评价指标，DE、PSO、DE-10 和 PSO-10 始终优于 GA、GA-10、BR、CombSUM 和 CombMNZ；DE-10 和 DE 稳定性最好，随着迭代次数的增加，性能越来越好；当迭代次数达到 100 时，DE 超过所有算法的性能，DE-10 优于 PSO-10、GA-10、BR、CombSUM 和 CombMNZ。②不同算法的性能折线波动不明显，说明迭代次数从 200 次减少到 25 次的过程中，性能变化不大，而 DE-10 变化最小；DE-10 在 3 个数据集、迭代次数为 25 次的汇聚结果的性能分别是 200 次迭代汇聚结果性能的 99.98%、99.93%和 99.33%。其他算法也具有相似的结论。上述结果表明，随着迭代次数的减少，DE-10 的性能下降不显著。

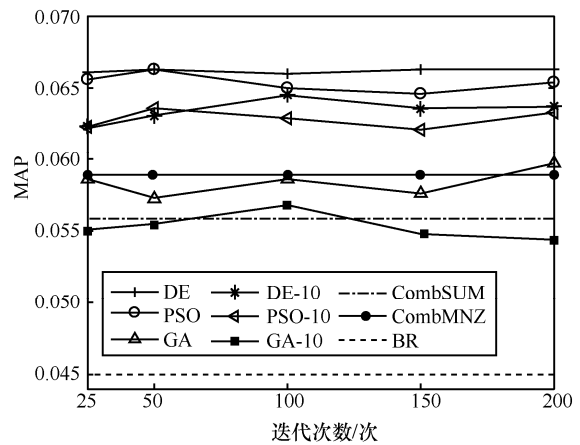


图 3 TREC-11B 中不同算法在不同迭代次数下汇聚结果性能

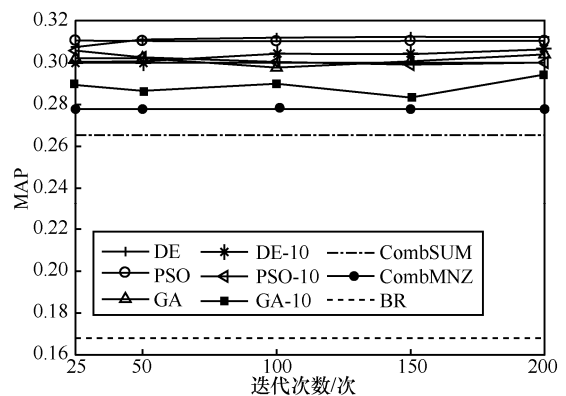


图 4 TREC-10 中不同算法在不同迭代次数下汇聚结果性能

表 6 数据表明，所有算法的运行时间和迭代次数呈线性关系，运行时间随着迭代次数的减少而线性减少。基于 Top-*k* 的 DE-10、PSO-10 和 GA-10 进行 25 次迭代的运行时间在 TREC-11A 和 TREC-11B 数据集中为 9~12 s，在 TREC-10 数据

表 6 不同迭代次数下不同算法结果汇聚的运行时间

算法	TREC-11A					TREC-11B					TREC-10				
	25	50	100	150	200	25	50	100	150	200	25	50	100	150	200
DE	27.8 min	51.5 min	1.54 h	2.27 h	3.02 h	30.7 min	58.2 min	1.93 h	3.01 h	4.02 h	56 min	1.56 h	2.83 h	4.08 h	5.35 h
PSO	28.3 min	57.9 min	1.96 h	2.76 h	3.45 h	47.3 min	1.58 h	3.22 h	5.18 h	6.75 h	1.58 h	2.93 h	6.02 h	9.21 h	12.4 h
GA	30.3 h	53.3 min	1.65 h	2.39 h	3.08 h	30.4 min	57 min	1.78 h	2.69 h	3.53 h	55.7 min	1.57 h	2.8 h	4.08 h	5.49 h
DE-10	10 s	17 s	29 s	40 s	51 s	12 s	18 s	34 s	48 s	63 s	12 min	13.6 min	16.9 min	19.1 min	19.6 min
PSO-10	9 s	16 s	28 s	43 s	53 s	12 s	21 s	36 s	49 s	62 s	11.7 min	11.7 min	13.3 min	16.1 min	19.9 min
GA-10	10 s	17 s	31 s	44 s	58 s	10 s	19 s	34 s	50 s	62 s	9.6 min	10.5 min	13 min	15.5 min	17.6 min

集中为 9.6~12 min，相比 200 次迭代的运行时间得到显著下降，满足任务结果汇聚实时性的要求。

综上所述，DE-10 通过降低迭代次数，以性能轻微损失为代价，显著降低运行时间。相比 PSO-10 和 GA-10，DE-10 在保证高质量的汇聚结果的前提下达到了实时结果汇聚的要求。

6.2.3 和高质量提交结果的性能分析与对比

自 2002 年 TREC-11 会议以来，信息过滤技术飞速发展，目前查到的最新关于使用 TREC-11 数据集进行实验的文献是 2015 年 Yang 等^[35]发表的。该文献使用最大匹配模式主题模型 MPBTM 大幅提高系统的 MAP 性能。将 MPBTM 作为高质量专家提交结果，并与本文算法进行对比，如图 5 所示。

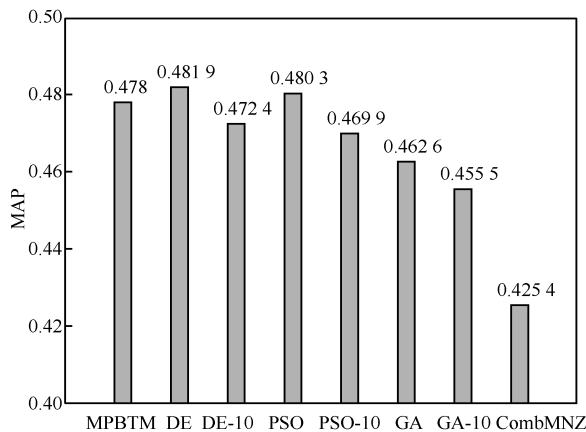


图 5 高性能系统和文中算法性能对比

以 MAP 为性能评价指标，MPBTM 的性能是 0.478，优于 BR 的 0.369，相比 BR 提高 29.54%，也优于 CombSUM、CombMNZ、GA、GA-10 和 PSO-10 的性能，低于 DE 和 PSO 的性能，略高于 DE-10 的性能，即 DE-10 相比 MPBTM 性能下降不显著。但是 DE 和 PSO 耗时过长，而基于 Top-k 训练的 DE-10 却能够实现实时结果汇聚，即通过汇聚

多个低质量的提交结果能够达到高质量专家提交结果的性能。

综上，本节验证了 DE-k 能够实现对多个低质量提交结果进行结果汇聚得到高质量的提交结果，同时满足实时性要求，相比其他算法，综合性能最优。

7 结束语

本文针对众包 Worker 完成任务存在一定的不确定性和差异性，并考虑在提升结果汇聚质量的同时又具有较快的汇聚速度，提出了基于 Worker 权重的差分进化和 Top-k 排序的结果汇聚算法。该算法首先对训练任务所有提交结果的 Top-k 排序通过基于差分进化算法的优化模型学习 Worker 优化的权重向量，然后基于优化的权重向量使用 LC 算法对真实任务进行结果汇聚。该算法在权重学习阶段，仅在信息足够丰富且更确定的 Top-k 数据项集合中进行，相比全排序，可以减少训练噪声，大幅减少运算量，提升权重学习的速度。结果汇聚阶段利用优化的权重向量进行加权汇聚，可避免 Worker 的差异性而提升结果汇聚质量。所提算法在获得优化的汇聚结果性能的同时，能够提升结果汇聚速度。定性分析证明了所提算法的正确性，仿真实验结果也验证了所提算法效果，与相关算法对比，所提算法的综合性能最优。

参考文献：

- [1] JIN Y, CARMAN M, ZHU Y, et al. A technical survey on statistical modelling and design methods for crowdsourcing quality control[J]. Artificial Intelligence, 2020, 287: 103351.
- [2] WU O, YOU Q, XIA F, et al. Listwise learning to rank from crowds[J]. ACM Transactions on Knowledge Discovery From Data, 2016, 11(1): 1-39.
- [3] CHANG Y, CHEN J, CHO M H, et al. Clustering from multiple uncertain experts[C]//Proceedings of the 34th International Conference

- on Machine Learning. New York: ACM Press, 2017: 674-683.
- [4] LI G, CHAI C, FAN J, et al. CDB: a crowd-powered database system[J]. Proceedings of the VLDB Endowment, 2018, 11(12): 1926-1929.
- [5] PARAMESWARAN A, BOYD S, GARCIA-MOLINA H, et al. Optimal crowd-powered rating and filtering algorithms[J]. Proceedings of the VLDB Endowment, 2014, 7(9): 685-696.
- [6] WANG S G, XIAO X K, LEE C H. Crowd-based deduplication: an adaptive approach[C]/International Conference on Management of Data. New York: ACM Press, 2015: 1263-1277.
- [7] LI K Y, ZHANG X H, LI G L, et al. A rating-ranking method for crowdsourced top- k computation[C]/International Conference on Management of Data. New York: ACM Press, 2018: 975-990.
- [8] VUURENS J, VRIES A P, EICKHOFF C. How much spam can you take? An analysis of crowdsourcing results to increase accuracy[C]/SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval. New York: ACM Press, 2011: 21-26.
- [9] ZHANG Y, DER SCHAAR M V. Reputation-based incentive protocols in crowdsourcing applications[C]/2012 Proceedings IEEE INFOCOM. Piscataway: IEEE Press, 2012: 2140-2148.
- [10] GHOSH K, PARUI S K, MAJUMDER P. Learning combination weights in data fusion using genetic algorithms[J]. Information Processing & Management, 2015, 51(3): 306-328.
- [11] XU C L, HUANG C L, WU S L. Differential evolution-based fusion for results diversification of Web search[C]/International Conference on Web-age Information Management. Berlin: Springer, 2016: 429-440.
- [12] WANG W, ZHANG M. Tensor deep learning model for heterogeneous data fusion in Internet of things[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2020, 4(1): 32-41.
- [13] CHATTERJEE S, MUKHOPADHYAY A, BHATTACHARYYA M. A weighted rank aggregation approach towards crowd opinion analysis[J]. Knowledge-Based Systems, 2018(149): 47-60.
- [14] THOMA S, THALHAMMER A, HARTH A, et al. FusE: entity-centric data fusion on linked data[J]. ACM Transactions on the Web, 2019, 13(2): 1-36.
- [15] FOX E A, SHAW J A. Combination of multiple searches[C]/The 2nd Text Retrieval Conference. [S.n.:s.l.], 1994: 243-252.
- [16] VOGT C C, COTTRELL G W. Fusion via a linear combination of scores[J]. Information Retrieval, 1999, 1(3): 151-173.
- [17] THOMPSON P. Description of the PRC CEO algorithms for TREC[C]/The First Text Retrieval Conference.[S.n.:s.l.], 1992: 337-342.
- [18] BARTELL B T, COTTRELL G W, BELEW R K. Automatic combination of multiple ranked retrieval systems[C]/17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York: ACM Press, 1994: 173-181.
- [19] VOGT C C, COTTRELL G W. Predicting the performance of linearly combined IR systems[C]/21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 1998: 190-196.
- [20] WU S L. Linear combination of component results in information retrieval[J]. Data and Knowledge Engineering, 2012, 71(1): 114-126.
- [21] LILLIS D, TOOLAN F, COLLIER R, et al. Probfuse: a probabilistic approach to data fusion[C]/29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2006: 139-146.
- [22] WU S L, BI Y X, ZENG X Q, et al. Assigning appropriate weights for the linear combination data fusion method in information retrieval[J]. Information Processing and Management, 2009, 45(4): 413-426.
- [23] SHELDON D, SHOKOUI M, SZUMMER M, et al. Lambdamerge: merging the results of query reformulations[C]/The Fourth ACM International Conference on Web Search and Data Mining. New York: ACM Press, 2011: 795-804.
- [24] KOZOROVITSKY A K, KURLAND O. Cluster-based fusion of retrieved lists[C]/The 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2011: 893-902.
- [25] HONG D, SI L. Mixture model with multiple centralized retrieval algorithms for result merging in federated search[C]/The 35th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2012: 821-830.
- [26] LEE C J, AI Q Y, CROFT W B, et al. An optimization framework for merging multiple result lists[C]/The 24th ACM International on Conference on Information and Knowledge Management, 2015: 303-312.
- [27] MOURAO A, MAGALHAES J. Low-complexity supervised rank fusion models[C]/Proceedings of the 27th ACM International Conference on Information and Knowledge Management. New York: ACM Press, 2018: 1691-1694.
- [28] LIANG S S, MARKOV I, REN Z C, et al. Manifold learning for rank aggregation[C]/World Wide Web Conference. [S.n.:s.l.], 2018: 1735-1744.
- [29] WU S L, HUANG C L, et al. Fusion-based methods for result diversification in Web search[J]. Information Fusion, 2019, 45: 16-26.
- [30] 郝晓辰, 王立元, 刘金硕, 等. WSN 中基于双群体差分进化的资源分配优化算法[J]. 通信学报, 2018, 39(4): 68-75.
- HAO X C, WANG L Y, LIU J S, et al. Resource allocation optimization algorithm based on double populations differential evolution in WSN[J]. Journal on Communications, 2018, 39(4): 68-75.
- [31] MONTAGUE M, ASLAM J. Relevance score normalization for meta-search[C]/The Tenth International Conference on Information and Knowledge Management. New York: ACM Press, 2001: 427-433.
- [32] FRANKLIN M J, KOSSMANN D, KRASKA T, et al. CrowdDB: answering queries with crowdsourcing[C]/The ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2011: 61-72.
- [33] LAN Y Y, NIU S Z, GUO J F, et al. Is top- k sufficient for ranking?[C]/The 22nd ACM International Conference on Information & Knowledge Management. New York: ACM Press, 2013: 1261-1270.
- [34] PAL S, MITRA M, KAMPS J. Evaluation effort, reliability and reusability in XML retrieval[J]. Journal of the Association for Information Science and Technology, 2011, 62(2): 375-394.
- [35] GAO Y, XU Y, LI Y F. Pattern-based topics for document modelling in information filtering[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(6): 1629-1642.

[作者简介]



邢玉萍 (1979-), 女, 江苏盐城人, 江苏大学博士生, 主要研究方向为众包、数据融合、人工智能、区块链等。

詹永照 (1962-), 男, 福建尤溪人, 博士, 江苏大学教授, 主要研究方向为人工智能和模式识别。