文章编号:1000-5862(2018)06-0616-05

基于决策树桩的元特征提取

曾子林[1],陈建军[2]
(1. 解放军陆军步兵学院,江西 南昌 330103;2. 上饶职业技术学院,江西 上饶 334109)

摘要: "No Free Lunch"定理表明:若无任何先验假设,则没有理由认为一种算法优于另一种算法. 算法的性能与问题的元特征密切相关. 目前的元特征提取方法只关注从数据集中提取元特征,而忽略了候选算法元特征的提取. 为此,在原有元特征集合的基础上提出基于决策树桩的元特征提取方法,将候选算法信息纳入新的元特征集合中. 实验表明:在传统元特征集合中加入基于决策树桩的元特征后,算法排序的预测准确率能够得到显著提高.

关键词: 元特征;算法性能;算法排序;决策树桩
中图分类号: TP 391 **文献标志码:** A **DOI:** 10.16357/j.cnki.issn1000-5862.2018.06.12

0 引言

随着人工智能技术的飞速发展,数据挖掘和机器学习领域内涌现出大量的学习算法. 而根据 D. H. Wolpert 等[1]提出的"没有免费的午餐"(No Free Lunch,NFL)定理,若无任何先验假设,则没有理由认为一种算法优于另一种算法. 如何从大量的学习算法中选择适合问题元特征的算法是目前迫切需要解决的问题. 该问题实际上是典型的算法选择问题. 早在 20 世纪 90 年代初期,就有学者意识到算法选择实际上是一种学习任务[2]. 之后,随着算法选择研究的不断深入,逐渐形成了元学习领域[3-6]. 元学习的目标是通过建立问题元特征和算法性能之间的映射关系,为用户提供算法选择的建议,从而有效减少算法选择所耗费的时间,而元学习中最核心的步骤是提取能够有效反映算法性能差别的元特征.

高质量的元特征能够有效反映学习问题对算法性能的影响. 目前,具有代表性的元特征可分为 3 类[7]:基于统计和信息论的元特征[8-9],基于基准思想的元特征[10],基于模型的元特征[11]. 然而,上述类型的元特征主要局限于对数据集(问题)进行元特征提取,却没有考虑候选算法的元特征. 众所周知,每种算法都有其相应的适用范围以及局限性. 如决策树算法和朴素贝叶斯算法不适用于特征具有关联关系的数据集,KNN 算法不适用于样本量大的数据集,支持向量机对缺失数据敏感等. 虽然,也有一些学者对算法的元特征进行了研究[12-13],但由于算法的定性元特征主观性较强,而算法的定量元特征实验过程比较复杂且难以度量,因此,鲜有文献将算法的元特征纳入到元特征集合中. 元特征的选择直接关系到元学习的预测准确率. 为了更好地预测候选算法的排序,提高元学习的准确率,在传统元特征的基础上,提出基于决策树桩的元特征提取方法,将候选算法的元特征纳入到元特征集合中,使元特征集合中包含候选算法的信息,从而提高候选算法排序的准确率.

1 标签排序数据集

从用户的角度出发,算法选择的目标是以最小开销减少可供选择算法的个数,从而达到节省时间的目的. 因此,算法推荐方法不需要准确地预测算法在某个数据集上的真实性能,而只需预测出算法的排序,即算法的相对性能. 与选出单个性能最优算法相比,算法排序能够给用户提供更多的信息.

为了使用机器学习的方法来解决算法排序问题,有必要以数据集的形式来描述候选算法的性能

收稿日期:2018-04-26
基金项目:国家自然科学基金(11501281),装备军内科研课题(面向作战任务的分队战斗体能数据分析评估系统建设)和江西省社科"十二五"规划课题(15GL44)资助项目.
通信作者:曾子林(1981-),女,江西鄱阳人,讲师,博士,主要从事元学习、特征选择方面的研究. E-Mail:zzljxnu@163.com

以及数据集的元特征,这里统称为元数据,其形式如表 1 所示.

表 1　元数据集

	f_1	f_2	f_3	a_1	a_2	a_3
d_1	-0.32	0.36	150	0.96	0.85	0.74
d_2	-0.75	1.27	200	0.64	0.73	0.52
d_3	-0.83	1.45	300	0.75	0.81	0.90

在表 1 中,d_1,d_2,d_3 分别代表 3 个数据集,f_1,f_2,f_3 表示数据集的元特征,a_1,a_2,a_3 表示候选算法在数据集上的性能值. 根据候选算法的性能值,可以得到候选算法的排序(也称为目标排序),其形式如表 2 所示. 可以看出,转化后的元数据集的类标签为排序形式,因此称之为标签排序数据集.

表 2　标签排序数据集

	f_1	f_2	f_3	a_1	a_2	a_3
d_1	-0.32	0.36	150	1	2	3
d_2	-0.75	1.27	200	2	1	3
d_3	-0.83	1.45	300	3	2	1

2　基于决策树桩的元特征生成方法

决策树桩是指分裂次数只有一次、只有单个节点的决策树,由于其结构简单,计算复杂度低,因而经常作为集成算法中的基分类器来使用. 为此,先通过决策树桩来生成元规则,再根据元规则形成可以反映候选算法信息的元特征.

给定一个大小为 $n \times m$ 的元数据集,其中 n 是数据集的个数,$m = m_f + m_a$,m_f 是数据集元特征的个数,m_a 是候选算法的个数. 基于决策树桩的元规则生成方法分为 2 步.

Step 1　将该元数据集转化为 $C_{m_a}^2$ 个两两算法性能比较的分类问题. 如表 2 中有 3 个候选算法,对每对算法(a_1,a_2),(a_1,a_3)和(a_2,a_3)分别建立一个分类模型,可得到 C_3^2 个 2 元分类模型. 这 3 个分类模型的训练数据相同,都为 $n \times m_f$ 矩阵,且每个 2 元分类模型有 2 个类标签$\{a_i, a_j\}$. 当类标签为 a_i 时,算法 a_i 的性能优于算法 a_j; 当类标签为 a_j 时,算法 a_j 的性能优于算法 a_i. 就表 2 而言,转化后的 3 个 2 元分类问题如表 3 所示(表 3 中的每列分别表示一个 2 元分类元数据集).

Step 2　对于每个 2 元分类元数据集,可以通过决策树桩算法建立一个基于规则的分类模型,并生成元规则. 如表 3 中的第 1 个 2 元分类元数据集(表 3 第 1 列),通过决策树桩算法可生成 2 条元规则:

(i) 若 $f_1 > -0.535$,则算法 a_1 的性能优于算法 a_2;

(ii) 若 $f_1 \leqslant -0.535$,则算法 a_2 的性能优于算法 a_1.

将上述 2 条元规则转化为一个布尔型的元特征,为了与传统的元特征区分,将新生成的元特征称为基于决策树桩的元特征(Decision-Stump Based Meta-Feature,DSMF),而传统的元特征称为基元特征(Base Meta-Feature,BMF). 对于一个新的数据集,基于决策树桩的元特征的值可以通过基元特征的值来确定. 以上述生成的元规则为例,若数据集的基元特征 f_1 满足规则(i)的条件,则新的元特征的值为 1; 若数据集的基元特征 f_1 满足规则(ii)的条件,则新的元特征的值为 0. 按照这种方法,可以得到 $C_{m_a}^2$(m_a 为候选算法的个数)个基于决策树桩的元特征.

表 3　3 个 2 元分类元数据集

(a_1,a_2)	(a_1,a_3)	(a_2,a_3)
$\{-0.32, 0.36, 150, a_1\}$	$\{-0.32, 0.36, 150, a_1\}$	$\{-0.32, 0.36, 150, a_2\}$
$\{-0.75, 1.27, 200, a_2\}$	$\{-0.75, 1.27, 200, a_1\}$	$\{-0.75, 1.27, 200, a_2\}$
$\{-0.83, 1.45, 300, a_2\}$	$\{-0.83, 1.45, 300, a_3\}$	$\{-0.83, 1.45, 300, a_3\}$

3　实验比较

为了验证 DSMF 的有效性,实验比较了单独使用 BMF 和加入 DSMF 后的算法排序预测的准确率.

3.1　实验设计

实验采用的元算法为基于实例的排序算法——k-近邻算法. 给定一个数据集的算法排序问题,k-近邻算法首先根据距离函数选择与该数据集元特征最接近的 k 个数据集,然后根据这 k 个数据集的算法性能排序结果生成新数据集的算法排序结果. 实验采用的基元特征如表 4 所示.

算法的准确率和运行时间是用户比较关心的性能测度. 实验采用的综合性能测度类似于 P. Brazdil 等[14] 提出的多准则评估测度 ARR,该测度结合了算法准确率和总执行时间(包括训练集和测试集的运行时间)的信息,定义为

$$P_{a_p,D_i} = A_{a_p,D_i} / (1 + \alpha \log(T_{a_p,D_i}))$$,

其中 A_{a_p,D_i} 表示算法 a_p 在数据集 D_i 上的准确率,T_{a_p,D_i} 表示算法 a_p 在数据集 D_i 上运行的时间,α 表示用户自定义的时间与准确率之间的相对重要性.

表 4 实验采用的基元特征列表

元特征类型	元特征
基于统计、信息论	属性个数、样本大小、离散属性比例、连续属性比例、缺失值比例、均值、方差、平均偏度、平均峰度、类熵、平均互信息
基于基准思想	朴素贝叶斯、线性判别、决策树树桩、1-近邻
基于模型	决策树的高度、决策树的宽度、决策树的节点数目、决策树的叶子数目

实验比较了 50 个数据集在 10 个候选算法上的性能排序,其中 50 个公用数据集来源于 UCI 数据库,10 个学习算法来自于目前较流行的数据挖掘软件 WEKA(见表 5.)

表 5 候选算法列表

算法类型	算法
基于贝叶斯理论	Naïve Bayes、Bayes Net
基于树结构	J48、Random Forest
基于规则	Decision Table、Jrip、PART
基于函数	Logistic、SMO
基于实例	IBK

参数的设置会影响到算法的性能,为了公平起见,这 10 种算法的参数均采用 WEKA 中默认的参数. 这些候选算法在 50 个公用数据集上的性能通过取 10 次 10 层交叉验证的平均值得到,并在每个数据集上形成算法排序. 分别提取 50 个数据集的基元特征和基于决策树桩的元特征,与 50 个数据集在 10 个候选算法上的性能排序,形成样本量为 50 的标签排序数据集(元数据集). 为了验证基于决策树桩的元特征能否提高算法排序的准确率,需要确定排序准确率的评估标准,经常采用的评估标准有 Spearman 秩相关系数[15] 和 Kendall 秩关联系数[16]:

设 $T = [T_1, T_2, \cdots, T_m]$, $P = [P_1, P_2, \cdots, P_m]$ 分别是 m 个候选算法的目标排序和预测排序,则 Spearman 秩相关系数定义为

$$\rho = 1 - 6\sum_{i=1}^{m} d_i^2 / (m(m^2 - 1))$$,

其中 $d_i^2 = (T_i - P_i)^2$.

Kendall 秩关联系数定义为

$$\tau = 1 - 4d_K(T,P) / (m(m-1))$$,

其中 $d_K(T,P) = \#\{(i,j): i<j, (T_i - T_j)(P_i - P_j) < 0\}$.

显然,ρ 和 τ 的取值范围在 $[-1,1]$ 内,当 2 个排序完全相同时,ρ 和 τ 的值都为 1;当 2 个排序完全相反时,ρ 和 τ 的值都为 -1.

3.2 BMF 与 BMF + DSMF 的比较

为了检验新元特征的有效性,将实验分成 2 组:一组实验使用的元特征仅为基元特征(BMF);另一组实验采用的元特征包括基元特征和基于决策树桩的元特征(BMF + DSMF). k-近邻算法的参数取值为 2,选取 90% 的元数据集样本作为训练数据,10% 的样本作为测试数据,最后取 10 次运行结果的平均值. 为了进一步说明提取元特征的重要性,实验还给出了元数据集默认排序(Default Ranking,DR)的运行结果. 默认排序是指将训练样本的平均排序作为测试样本的排序,因此每个测试样本的预测排序都是相同的. 默认排序经常作为参照排序来衡量元特征或元算法的有效性. 10 次运行结果的排序性能比较如表 6 和表 7 所示.

表 6 运行 10 次的 Spearman 秩相关系数的比较

Run	DR	BMF	BMF + DSMF
1	0.551 5	0.780 4	0.956 0
2	0.533 3	0.826 4	0.937 5
3	0.793 9	0.960 5	0.980 2
4	0.509 1	0.925 4	0.930 1
5	0.609 1	0.843 1	0.966 6
6	0.672 7	0.717 6	0.952 0
7	0.472 7	0.976 5	0.861 7
8	0.509 1	0.844 7	0.938 9
9	0.506 1	0.881 9	0.948 2
10	0.569 9	0.926 1	0.978 7
Avg.	0.572 7	0.868 2	0.945 0
Std.	0.097 1	0.081 9	0.033 8
p 值	1.34063E-07	0.033 276	

注: Avg. 和 Std. 分别表示均值和标准方差,下同.

表 7 运行 10 次的 Kendall 秩关联系数的比较

Run	DR	BMF	BMF + DSMF
1	0.455 6	0.705 1	0.899 5
2	0.444 4	0.733 6	0.821 7
3	0.622 2	0.932 5	0.938 2
4	0.433 3	0.858 6	0.865 1
5	0.511 1	0.741 2	0.898 8
6	0.533 3	0.643 8	0.881 2
7	0.411 1	0.924 4	0.837 2
8	0.422 2	0.760 2	0.871 4
9	0.433 3	0.787 1	0.863 8
10	0.494 4	0.837 6	0.932 2
Avg.	0.476 1	0.792 4	0.880 9
Std.	0.065 3	0.094 5	0.037 5
p 值	4.3804E-10	0.017 346	

从表 6 和表 7 可以看出,以 BMF + DSMF 为元特征的算法排序性能最高. 除此之外,还对实验结果进行了统计显著性检验,检验 BMF + DSMF 与 DR、BMF 的性能相比差异是否有统计学意义. 采取的统计显著性检验方法为单尾成对 t 检验,置信度设为 95%,对应的 p 值见表 6 和表 7 的最后一行. 最后一行显示对应的统计 p 值均远远小于 0.05,这表明以 BMF + DSMF 为元特征的算法排序性能明显优于 DR 和 BMF.

k-近邻算法在 2 组元数据集中的 ρ 和 τ 的平均值比较如图 1 所示.

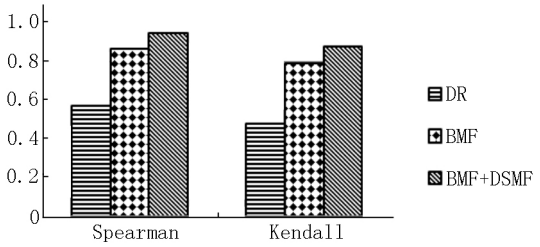

图 1　DR,BMF 和 BMF + DSMF 的性能比较

从图 1 可以看出,使用元特征能够大大提高算法的排序性能. 如当不使用元特征时,默认排序(DR)的平均 Spearman 秩相关系数为 0.572 7,而使用基元特征后的平均 Spearman 秩相关系数为 0.868 2,排序性能提高了 51.60%. 这说明元特征提取是非常有必要的. 从图 1 中还可以发现,将基元特征和基于决策树桩的元特征结合在一起使用时,算法排序预测性能最高. 例如,当单独使用基元特征时,平均 Kendall 秩关联系数为 0.792 4,而当结合基于元规则的元特征一起使用时,平均 Kendall 秩关联系数为 0.880 9,排序性能提高了 11.17%. 这表明在元特征集合中加入候选算法的排序信息能够有效提高算法排序的准确率.

为了进一步说明 BMF + DSMF 的优越性,对 k-近邻算法取不同 k 值(k 从 1 取到 10)进行了实验,图 2 和图 3 分别显示了 DR,BMF 和 BMF + DSMF 在不同 k 值上的排序性能比较.

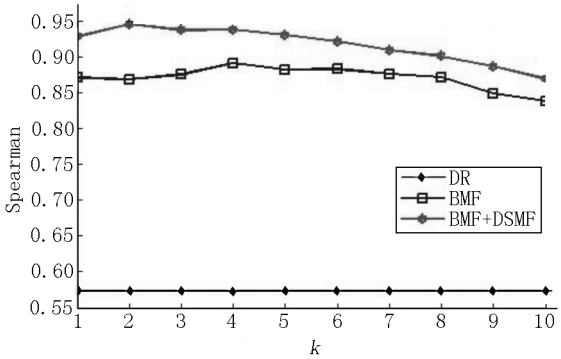

图 2　DR,BMF 和 BMF + DSMF 在不同 k 值上的平均 Spearman 秩相关系数比较

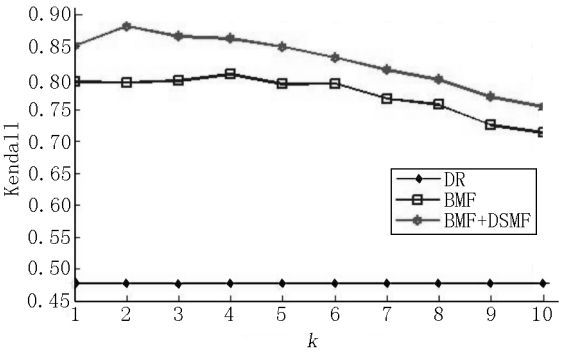

图 3　DR,BMF 和 BMF + DSMF 在不同 k 值上的平均 Kendall 秩关联系数比较

从图 2 和图 3 可以看出,不管 k 取什么值,BMF + DSMF 的排序预测性能总是优于 BMF 和 DR. 对于一个样本规模为 50 的元数据集来说,k-近邻算法的参数取值范围在 [2,4] 内效果较好. 当 $k > 4$ 时,随着 k 的增大,排序预测性能呈递减趋势.

4　结论与展望

元特征提取一直是元学习领域研究者关注的热点. 但是,目前的元特征提取方法大部分只关注从数据集中提取元特征,而忽略了对候选算法元特征的定量描述. 针对现有元特征类型不能反映候选算法信息的问题,提出了基于决策树桩的元特征提取方法,并通过实验验证了在传统元特征集合中加入基于决策树桩的元特征后,算法排序预测的准确率能够得到显著提高.

元学习领域在国内研究甚少,虽然实验结果表明基于决策树桩的元特征能够有效提高算法排序的准确率,但仍有问题需进一步完善. 为了简化实验,本文在计算候选算法性能时,采用的都是候选算法的默认参数,但是不同的参数设置对算法性能是有一定影响的. 因此,下一步工作将会采用相同的优化算法寻求候选算法的最优参数,并按照最优参数的算法性能对候选算法进行排序.

5　参考文献

[1] Wolpert D H, Macready W G. No free lunch theorems for search [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67-82.

[2] Rendell L, Cho H. Empirical learning as a function of concept character [J]. Machine Learning, 1990, 5(3): 267-298.

[3] Cruz R M O, Sabourin R, Cavalcanti G D C. Meta-des. oracle: meta-learning and feature selection for dynamic en-

semble selection [J]. Information Fusion, 2017, 38: 84-103.

[4] Filchenkov A, Pendryak A. Datasets meta-feature description for recommending feature selection algorithm [C] // Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search Fruct Conference, IEEE, 2015: 11-18.

[5] Sousa A F M, Prudêncio R B C, Ludermir T B, et al. Active learning and data manipulation techniques for generating training examples in meta-learning [J]. Neurocomputing, 2016, 194: 45-55.

[6] Morais R F A B D, Miranda P B C, Silva R M A. A meta-learning method to select under-sampling algorithms for imbalanced data sets [C] // Brazilian Conference on Intelligent Systems, IEEE Computer Society, 2016: 385-390.

[7] 曾子林,张宏军,张睿,等. 基于元学习思想的算法选择问题综述 [J]. 控制与决策, 2014, 29(6): 961-968.

[8] Rossi A L D, Carvalho A C P L F, Soares C, et al. Meta stream: a meta-learning based method for periodic algorithm selection in time-changing data [J]. Neurocomputing, 2014, 127(3): 52-64.

[9] Song Qinbao, Wang Guangtao, Wang Chao. Automatic recommendation of classification algorithms based on data set characteristics [J]. Pattern Recognition, 2012, 45(7): 2672-2689.

[10] Pfahringer B, Bensusan H, Garrier C G. Meta-learning by landmarking various learning algorithms [C] // Proc of the 17th Int Conf on Machine Learning, San Francisco: Morgan Kaufmann, 2000: 743-750.

[11] Peng Y H, Flach P A, Soares C, et al. Improved dataset characterization for meta-learning [C] // Proc of Discovery Science 5th Int Conf Lubeck Germany, 2002: 141-152.

[12] Hilario M, Kalousis A. Fusion of meta-knowledge and meta-data for case-based model selection [C] // Lecture Notes in Computer Science, 2001, 2168: 180-191.

[13] Kalousis A, Hilario M. Building algorithm profiles for prior model selection in knowledge discovery systems [J]. International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, 2000, 8(2): 77-88.

[14] Brazdil P, Carrier C G, Soares C, et al. Meta learning: applications to data mining [M]. Berlin: Springer Science and Business Media, 2008.

[15] Spearman C. The proof and measurement of association between two things [J]. The American Journal of Psychology, 1904, 15(1): 72-101.

[16] Kendall M. Rank correlation methods [M]. London: Charles Griffin and Company Limited, 1948.

The Extraction of Meta-Feature Based on Decision Stump

ZENG Zilin[1], CHEN Jianjun[2]

(1. Army Infantry College of People's Liberation Army, Nanchang Jiangxi 330103, China;
2. Shangrao Vocational and Technical College, Shangrao Jiangxi 334109, China)

Abstract: The "No Free Lunch" theorem shows that there is no reason to think that one algorithm is superior to the other one without any prior assumptions. The performance of algorithm is closely related to the meta-feature of problem. The current meta-feature extraction method is only concerned with extracting meta-feature from the data set, while ignoring the meta-feature extraction of candidate algorithms. Therefore, an extraction method based on decision stump is proposed, which can effectively reflect the information of candidate algorithms. Experiments show that the new meta-feature sets significantly increase the prediction accuracy of algorithm ranking.

Key words: meta-feature; performance of algorithms; ranking of algorithms; decision stump

(责任编辑: 曾剑锋)