分类算法的算法自动选择和超参数调优

1190300321 郑晟赫

1. 课题来源及研究目的和意义

课题来自于本人在哈工大海量数据计算研究中心实习的课题，并进行了一定程度的拓展。希望通过本课题的研究提高目前在分类算法上的自动机器学习(AutoML)算法的准确性与算法选择速度。

为了将机器学习应用到新的任务中，需要合适选择算法和超参数，这一问题即算法选择和超参数优化(CASH)的组合问题。早期工作需要大量的人工来进行解决，可能会耗费大量时间或空间，而现有的方法大多是在具有较高时间或空间复杂性的 AutoML 背景下开发的。而本课题提出的算法不仅自动化了算法选择过程，同时还以数据驱动的方式加快了 CASH 解决速度。

2. 国内外相关研究现状分析

目前，关于自动机器学习已经有一些较为成熟的算法被提出，也在该学术领域和工业界被采纳，例如：Random Search[1]、Learning Curve-Based Prediction for Early Stopping[2]。以及一些较为成熟的自动机器学习系统也已提出并广泛应用于各种机器学习场合，例如：Auto-WEKA[3]、Auto-Sklearn[4]。而目前的算法主要从两方面来解决 CASH 问题：a)将算法自动选择问题也转化为超参数优化问题进行求解。b)运用 KNN、successive halving 等算法从某一算法集中进行自动选择算法。在此领域目前存在的问题有：a) 从数据集中提取的特征不能很好的表示数据集特性。b) 将候选算法进行分类时分类方法较为粗糙，导致算法推荐时搜索空间太大。c) 针对于一个算法性能指标衡量标准较为单一。

3. 主要研究内容

本课题提出的是一种基于强化学习的 Auto-Cash 算法。在 Auto-Cash 问题中有一个难点在于拟合数据集的分布，从而根据数据集的分布与已知数据集的相似程度为其推荐最优算法。数据集的相似度有较多方式估计，本课题引入 DQN[5]方式提取具有最优表示效果的元特征来表示数据集，两个数据集的相似度比较问题转化为两组元特征的相似度比较问题，降低了比较的维度。

在选择了最优元特征表示后，对于所有的数据集均以元特征的方式表示数据集，并对已知数据集进行最优算法选择。当用户输入待选择算法数据集时将其转化为元特征表示，利用经过已知数据集训练的随机森林对输入的待选择算法数据集元特征进行分类，选取对用户给定数据集而言较优算法。根据选择的算法和用户数据集，利用遗传算法进行超参数优化，最终得到推荐的算法与超参数。

4. 技术方案与详细设计

算法分为算法自动选择与超参数优化两部分，构建过程中分为 online 阶段与 offline 阶段。算法整体细节可见 alg1，整体方案见图 1。

Algorithm1：Auto-Cash

输入：所有训练数据集 $\boldsymbol{D_{train}}$ ，待选择算法集 $\boldsymbol{Alg}$，需要进行算法自动选择和超参数优化步骤的数据集 $\boldsymbol{D}$。

输出：对于 D 最优的算法 ***alg*** 和最优超参数配置 λ_{alg}。

1：在待选择算法集 ***Alg*** 中为每一个训练数据集 ***Dtrain*** 选择最优的算法；

2：根据训练数据集 D_{train}，使用 DQN 选择一个合适的待选择元特征列表；

3：将代表每个训练数据集的元特征向量和其对应的最优算法输入进随机森林模型；

4：训练随机森林模型；

5：利用随机森林为数据集 ***D*** 预测最优算法 ***alg***；

6：利用遗传算法搜索最优超参数配置 λ_{alg} ；

7：返回 ***alg*** 和 λ_{alg} ；

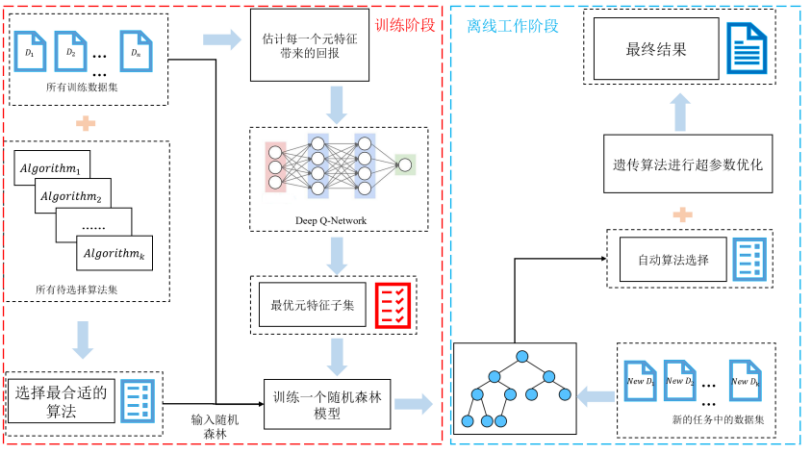

图 1 整体方案展示图

4.1 算法自动选择

4.1.1 数据集表示

通过之前的一些分类任务从中学习一些经验可以很大程度的加速当前的任务，因此对于训练集中的数据集 D，我们可以提取一个向量 ***Vd =<f1, f2 … fn >***来唯一的表示这个数据集，这样做可以提升训练的效率，同时降低运算所需要的算力需求。因此对于元特征的选择至关重要。我们选择的 5 种基本类型的元特征如下：a)类别信息熵；b)同一属性不同类别占比；c) 数值型属性平均值；d)数值型属性方差；e) 类别、属性、样本等数量

元特征的种类很多，但选择的元特征的数量多并不代表能带来更好的预测效果，如图 2 我们发现，当元特征数量较小的时候，随着选择的元特征数量的增加，对于预测数据集的精度会有明显的提升，但是当元特征数量达到一定程度之后，这一精度的上涨则明显减缓，甚至不再上涨，因此，我们引入了 DQN 算法来选择能带来相对最好的预测效果的元特征序列。

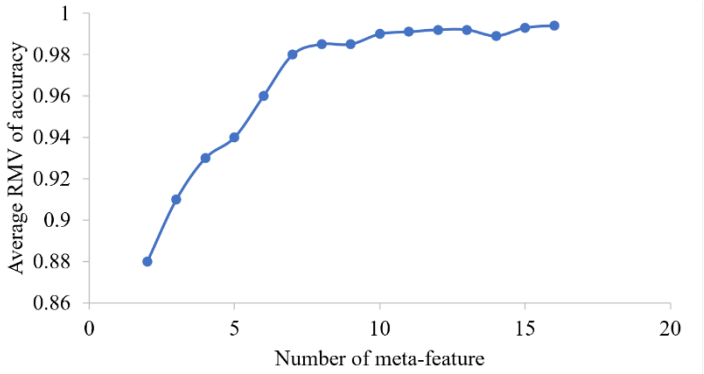

图 2 元特征数量对平均预测精度的影响

然而，DQN 主要用于解决自动连续决策问题。那么问题的关键转换为如何将元特征选

择问题转换为连续决策问题。这就需要对 DQN 中的状态进行特殊的定义实现。首先，我们构建状态集 S 和行为集 A，我们将某一个元特征是否选择用状态集中的 1/0 来表示，行为集表示对于某一个元特征的选择情况，同时，我们在 DQN 开始之前用随机森林估计每一个元特征的平均预测精度，详细数据见图 3。作为初始的奖励集 R，将 DQN 初始化为(S,A,R)，从而将元特征的选择问题转换为一个 01 串的选择问题。使用这种方式，就可以通过 DQN 的方式选择能带来最好表示精度的元特征序列。

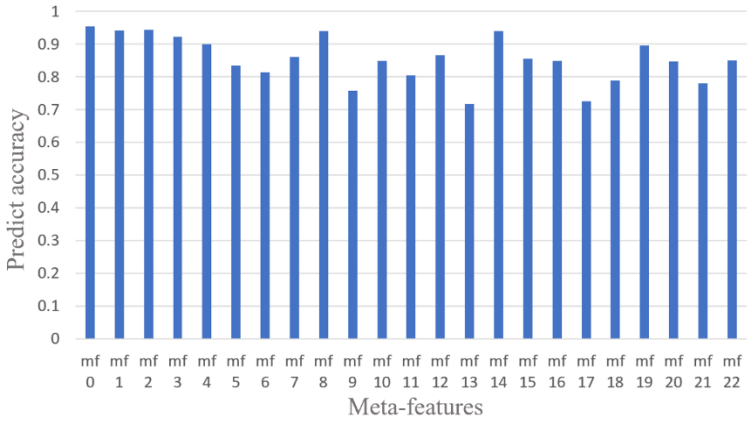

图 3 每个元特征性能估计回报值

而与传统的特征选择方法相比，DQN 最显著的优点是根据强化策略不断改进选择结果。一开始，经验的缺乏使得 DQN 的选择偏离了现实。随着训练的进行，DQN 会根据偏差调整学习率、折现率等参数，使下一次的选择更加合理，这种渐进式学习策略优化了选择的结果。

4.1.2 算法选择

此阶段的任务是建立数据集元特征表示 v 到最优算法 a 之间的映射关系。此阶段选用随机森林作为映射模型。

我们首先为每个数据集计算元特征向量，最终，将庞大规模的训练数据集转变为一个元特征表示的训练数据集。接下来我们使用遍历的方式为每一个数据集都分配最优的算法，从而有了一个基于已知数据集的 ***（v，s）*** 映射，记此映射为新数据集 ***B***，接下来就用训练集 ***B*** 训练随机森林分类器，得到一个训练好的随机森林分类器。

当用户输入一个数据集时，我们获取该数据集的元特征表示 ***a***，我们用得到的新数据集的元特征表示训练随机森里分类器，使得输入数据集的元特征向量，就可以通过随机森林去预测出最适合它的算法。当然实验中我们也尝试使用了其他的一些常见分类模型，比如 KNN，SVM，Adaboost 等算法，最终实验效果表明，由随机森林作为最后的分类器，可以最大的提升推荐出最好的算法的概率。

4.2 超参数调优

超参数调优部分选用目前超参数调优工作中效果较好且速度较快的遗传算法（Genetic Algorithm, GA）[5]进行。遗传算法详细流程见图 4。遗传算法是受到达尔文生物进化论的自然选择和遗传学机理的启发，被发展起来的一种全局迭代优化算法，是一种通过模拟自然进化过程来在搜索空间中搜索最优解的方法。经过初始化种群内部的交叉、变异和选择，选出优秀的下一代，然后经过同样的过程，经过世代延续，完成进化，达到最优性能，是一种成熟的基于种群的超参数调优算法。遗传算法最大的优点就是搜索速度快，可以用较短的时间搜索到全局最优解，避免陷入局部最优解。和目前性能同样优秀的贝叶斯优化相比较，搜索空间维度小，搜索时间短。开始时，我们使用二进制代码对超参数进行编码并进行初始化。

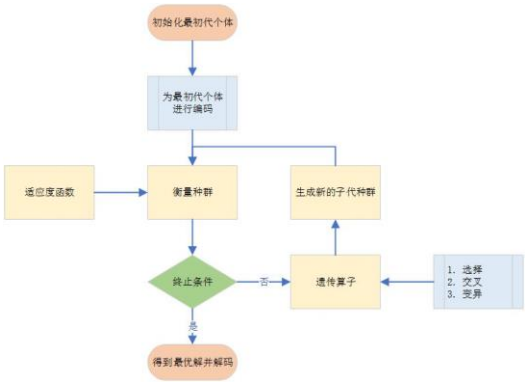

图 4 遗传算法流程图

我们选择最适合的个体批次，即具有特定超参数配置的算法性能，作为父本去生成子代。为了引入随机干扰，我们将交叉和变异作为遗传算子引入遗传过程中来，如图 5 所示。两个二进制序列（个体）在相同位置随机交换其子序列以表示交叉过程。个体的二进制数字因为突变而随机变化。实验结果表明，大多数情况下，个体的适应度将在 50 代内收敛到最佳值

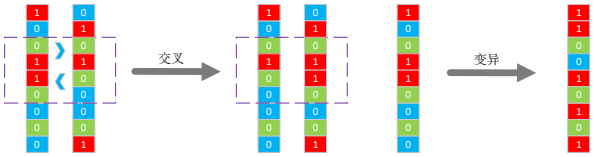

图 5 个体的交叉变异

4.3 衡量指标

目前的性能衡量指标大多是在测试集上的准确率和时间(CPU hour 和 GPU hour)，也有部分论文会考虑硬件问题，尽量去优化内存。诚然，算法在测试集上的准确率很重要，但是在一些情况下，单独看算法在测试集上的正确率并不能很好的反应算法性能。ROC 曲线是反映敏感性和特异性连续变量的综合指标[6]，ROC 曲线一个最大的优点就是不会随着样本数据分布的变化而变化。AUC（Area under the Curve of ROC 为 ROC 曲线下方的面积，它的意义也可以理解为随机抽取一个样本，分类器将其判断正确的概率。对于数据均衡与否，AUC 的值是不会发生变化的，因此 AUC 可以在绝大多数情况下来判断分类器分类性能的好坏，并且相对于准确率更有意义。同时优化 AUC 和正确率是一个多目标优化问题，一个经典的多目标优化问题的解决方法是设计权重最后加和，在我们的问题里被表示为

$$F_{score} = \omega 1 \cdot accuracy + \omega 2 \cdot AUC$$

但是分别优化正确率和 AUC 值并选择一个合适的权重分配会带来很多不必要的计算，所以为了降低计算复杂度，我们选择一种折中的方式去表示 Auto-CASH 中的性能衡量函数，如公式所示：

$$F_{score}(D,A)=accurate*AUC$$

其中 ***D*** 代表数据集，***A*** 代表要衡量的算法，accurate 代表该算法在该数据集分类正确率，AUC 代表该分类器的 AUC 面积。因为 accurate 和 AUC 都是（0,1）区间的值，并且只有当两者都比较大的时候，衡量指标才会比较大，这样同时考虑算法准确率和 AUC，更加均衡也更加有意义。

5. 实验分析与结论

5.1 样本集的构建

使用的所有数据集都是来自 UCI Machine Learning Repository 和 Kaggle 的真实多分类数据集。使用真实数据集的意义在于这样更能真实体现算法的性能，同时对于未来的优化有一定的指导意义。数据集对应最优算法信息如图 6 所示。

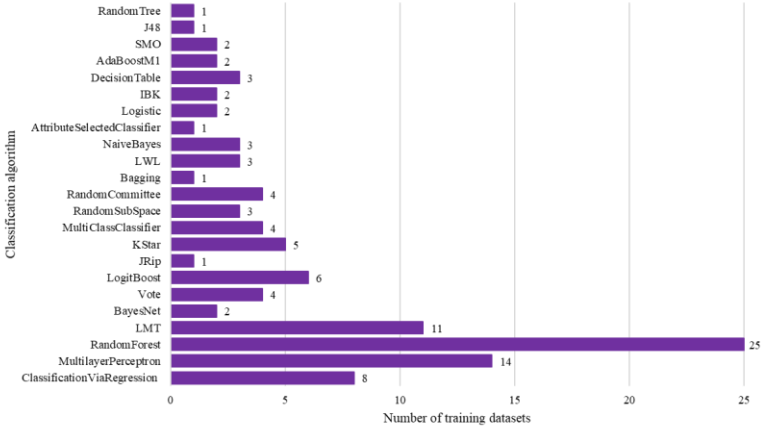

图 6 选用数据集相关信息

我们选择了 Auto-Cash 中 20 种较为典型且经试验较为有效的分类算法，详细信息如表 1 所示。

表 1 算法信息

算法	超参数个数	算法	超参数个数
AdaBoost	3	Bagging	3
AttributeSelectedClassifier	2	BayesNet	1
ClassifierViaRegression	2	IBK	4
DecisionTable	2	J48	8
JRip	4	KStar	2
Logistic	1	Logitboost	3
MultilayerPerception	5	MultiClass	3
RandomCommittee	2	NaiveBayes	2
RandomSubSpace	3	RandomForest	2
SMO	6	RandomTree	4
LWT	5	Vote	1
LWL	3		

5.2 实验与评价

我们评估了在 20 个分类数据集上的 Auto-CASH 的性能，同时评估了 Auto-WEKA 和 Auto-Model[6]的性能，详细的实验结果如表 2 所示（由于在某些数据集中 Auto-Model 无法输出具体的评估指标，在图中标记为 0)。可以发现，在大多数情况下 Auto-CASH 的性能与 Auto-WEKA 和 Auto-Model 相比更加优越。这得益于 DQN 选择元特征后随机森林的精准算法推荐与遗传算法的超参数优化过程的高效与精准。由此可见，本项目提出的方法优化了自动机器学习的准确程度，使得机器学习能在自动化程度更高的情况下优化机器学习的性能。

表 2 算法性能比较

数据集	**Auto-Cash**	**Auto-Model**	**Auto-WEKA**
D1	**0.998**	0.987	0.996
D2	**0.996**	0.942	0.947
D3	**0.408**	0.363	0.36
D4	0.925	**0.948**	0.711
D5	**0.845**	0.806	0.79
D6	**0.967**	0.965	1
D7	**0.882**	0	0.58
D8	**0.978**	0	0.886
D9	0.969	0.38	**0.974**
D10	**0.988**	0	0.976
D11	**0.563**	0.409	0.509
D12	**0.963**	0.591	0.11
D13	**1**	0.9	0.569
D14	0.961	**0.967**	0.952
D15	**0.979**	0.964	0.966
D16	0.677	**0.686**	0.633
D17	**0.986**	0	0.942
D18	0.951	0.951	0.951
D19	**0.952**	0.697	0.935
D20	**0.611**	0.569	0.478

6.参考文献

[1] Bergstra J, Bengio Y. Random search for hyper-parameter optimization[J]. Journal of machine learning research, 2012, 13(2).

[2] Li L, Jamieson K, Rostamizadeh A, et al. A system for massively parallel hyperparameter tuning[J]. arXiv preprint arXiv:1810.05934, 2018.

[3] Thornton C, Hutter F, Hoos H H, et al. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms[C]//Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013: 847-855.

[4] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J.T., Blum, M., & Hutter, F. (2015). Efficient and Robust Automated Machine Learning. NIPS.

[5] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

[6] Wang C, Wang H, Mu T, et al. Auto-model: utilizing research papers and HPO techniques to deal with the cash problem[C]//2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020: 1906-1909.

7.分工

在本实验中本人主要负责前期数据集选择、算法选择部分与一部分实验。