利用并行惯性权重 OOL-FA 的大数据分类

钟章生[1]，陈世炉[2]，陈志龙[3]

（1. 南昌理工学院 计算机信息工程学院，江西 南昌 330013；
2. 中国船舶总公司 第六三五四研究所，江西 九江 332000；
3. 南昌理工学院 电子与信息学院，江西 南昌 330013）

摘　要：针对现有大数据分类过程中特征选择算法精度较低，影响后续数据分类算法精度的问题，提出基于惯性权重正交反向学习（OOL）-萤火虫算法（FA）的大数据特征选择算法。借助 FA 的全局搜索能力，以及 OOL 分别在收敛速度、收敛精度方面的改进能力，实现数据特征的快速、精确选择，采用结构感知卷积神经网络对大数据特征进行精确分类。在包含 6600 万个样本和 2000 个属性的大数据集上进行实验，实验结果表明，所提算法在分类准确率上具有明显的优势。

关键词：大数据分类；惯性权重 OOL-FA 算法；结构感知神经网络；Spark 框架；特征选择；数据分类
中图法分类号：TP391　**文献标识号**：A　**文章编号**：1000-7024（2020）10-2818-07
doi：10.16208/j.issn1000-7024.2020.10.020

Inertia weight OOL-FA for big data classification

ZHONG Zhang-sheng[1], CHEN Shi-lu[2], CHEN Zhi-long[3]

(1. College of Computer Information Engineering, Nanchang Institute of Technology, Nanchang 330013, China;
2. 6354 Institute, China Shipping Corporation, Jiujiang 332000, China; 3. College of Electronics and Information, Nanchang Institute of Technology, Nanchang 330013, China)

Abstract: In view of the low accuracy of feature selection algorithm in the existing big data classification process, which affects the accuracy of the subsequent data classification algorithm, a big data feature selection algorithm based on the inertial weight orthogonal reverse learning (OOL)-firefly algorithm (FA) was proposed. With the help of the global search ability of FA and the improvement ability of OOL in convergence speed and accuracy, the data feature was selected quickly and accurately. The structure perception convolution neural network was used to classify the features of big data accurately. Experiments were carried out on a large dataset with 66 million samples and 2000 attributes. The results show that the proposed algorithm has obvious advantages in classification accuracy.

Key words: big data classification; inertia weight OOL-FA (IWOF) algorithm; structure-aware convolutional neural network (SACNN); Spark framework; feature selection; data classification

0　引　言

在数字化时代，深入挖掘海量数据内部蕴藏的有用信息来指导具体的工程问题，而在基于 MapReduce 范式的技术体系中，数据特征选择与数据分类是两项非常重要且复杂的工作[1-3]。

针对特征选择问题，部分文献提出针对高维数据的轻量级特征选择方法，采用加速粒子群优化对数据特征进行群搜索，在加快处理时间的同时提高了分析精度，但并未完全解决粒子群算法的局部最优问题[4-7]。针对数据分类问题，部分文献提出了基于在线打包集成的高效分类器，通过在训练实例上引入在线重采样机制和基于纠错输出码的鲁棒编码方法，减少了分类器之间相关性的影响，同时采用基于分类性能的动态更新模型减少不必要的更新操作，提高了分类效率。然而，并未解决分类精度与数据规模的矛盾，在分类精确性上仍需改进[8-11]。

收稿日期：2019-10-12；**修订日期**：2020-01-16
基金项目：国家自然科学基金项目（61663033）；江西省教育厅科学技术研究基金项目（GJJ180989）
作者简介：钟章生（1982-），男，江西赣州人，硕士，讲师，研究方向为大数据、计算机网络等；陈世炉（1966-），男，江西九江人，研究员，研究方向为计算机应用、惯性导航与测试技术；陈志龙（1969-），男，湖南永州人，博士，教授，研究方向为大数据等。
E-mail：z13576003643@126.com

为了有效提高数据选择与分类算法的速度与精度，在借鉴已有方法的基础上，提出基于惯性权重正交反向学习（orthogonal opposition learning，OOL）——萤火虫算法（firefly algorithm，FA）的数据特征选择算法：利用萤火虫算法实现数据特征的全局寻优，通过引入惯性权重来提高收敛速度，借助正交反向学习来提高选择精度，从而在特征选择过程的速度与精度上实现有效权衡。在此基础上，提出基于结构感知卷积神经网络（structure-aware convolutional neural network，SACNN）算法的数据分类方法，利用 SACNN 较强的非线性学习能力实现大数据的精准分类。在 Spark 框架下对所提方法进行实验分析，结果验证了所提方法的有效性和优越性。

1　利用 IWOF 算法的大数据特征选择

高维数据特征选择的目标是通过寻找特征最小子集来建立精确的数据预测模型。随着数据维数的指数级增长，现有的批量学习[12]和在线学习[13]方法已经很难满足特征选择对于快速性和可伸缩性的要求。为了解决这一问题，提出一种融合正交反向学习和萤火虫算法的新型数据特征选择算法，并利用惯性权重技术提升算法的收敛速度。首先，在映射阶段将原始大数据集分解为数据块；其次，基于正交反向学习和萤火虫算法选择大数据集特征；最后，将得到的部分结果合并到归约阶段的最终特征向量中。

1.1　问题描述

假设所研究的数据特征选择问题为二元分类中的特征选择问题。定义 $\{(x_t,y_t)\mid t=1,\cdots,T\}$ 为通过测试得到的输入数据样本，$x_t\in R^d$ 为 d 维向量，$y_t\in\{-1,+1\}$，t 代表实验序号。为了突出所提算法在大数据应用中的优势，假设 d 是一个很大的数值，因此从提高计算效率的角度来看，需要选择相对较少的特征进行数据分类。在每一个实验 t 中，通过设计分类器 $v_t\in R^d$ 并利用线性函数 $\mathrm{sgn}(v_t^Tv_t)$ 来对实例 x_t 进行分类。需要注意的是，并不是利用每一个特征进行分类，而是要求分类器 w_t 中包含的非零元素满足

$$\|v_t\|_0\leqslant N \tag{1}$$

式中：N 为预定义的常数，$\|\cdot\|_0$ 代表 0-范数。

为了在实现在线数据特征选择的同时尽可能地减小选择错误，考虑

$$\|x_t\|_2\leqslant 1,t=1,\cdots,T \tag{2}$$

式中：$\|\cdot\|_2$ 代表 2-范数。同时，需要保证算法获得每个训练实例的完整输入（即 $x_1,\cdots,x_T$）。

1.2　MapReduce 概述

MapReduce[14]是在大数据处理中应用最广泛的编程范式之一，是计算机集群化应用中的重要技术手段。MapReduce 分为两个阶段，映射和归约。映射阶段的作用是对输入数据集进行处理，得到一些中间结果，并对这些结果进行合并，以便在归约阶段生成最终的输出。

MapReduce 模式依赖于一个基本数据结构，其定义为

$$\langle k,v\rangle \tag{3}$$

首先，映射函数的所有应用程序都可以生成一个 $\langle k,v\rangle$ 对作为输入，并在映射阶段生成一组中间 $\langle k,v\rangle$ 对作为关键字，该过程表示为

$$\mathrm{map}(k_1,v_1)\rightarrow\{(k_2,v_2),\cdots,(k_n,v_n)\} \tag{4}$$

其次，映射归约库将每个中间数据对 $\langle k,v\rangle$ 和一个键聚在一起。

最后，归约函数获得聚合对，并生成一个新的 $\langle k,v\rangle$ 对作为输出，该过程表示为

$$\mathrm{reduce}(k_2,v_2)\rightarrow(k_2,v_3) \tag{5}$$

图 1 描述了 MapReduce 的流程图。

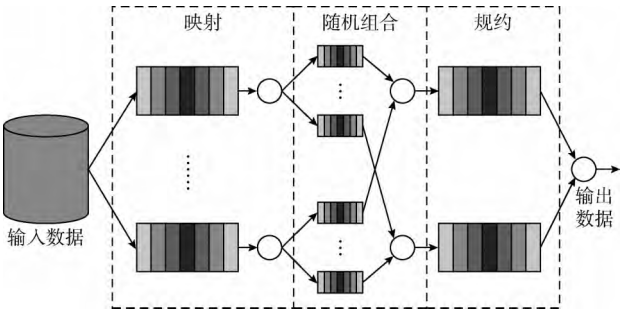

图 1　映射归约 MapReduce 模式流程

1.3　基于惯性权重 OOL-FA 算法的特征选择

萤火虫算法（FA）全局搜索能力强，可用于求解多目标优化问题[15]。为了弥补 FA 算法在局部搜索能力和收敛速度上的不足，将正交反向学习（orthogonal opposition learning，OOL）引入 FA，以深入挖掘并保存个体和反向个体中的有用信息。由此形成了一种新的启发式特征选择算法，即混合多目标 OOL-FA 算法，即 IWOF。

在具体介绍 IWOF 算法之前，需要先对经映射、规约后的数据进行编码和初始化，以形成可供 IWOF 算法使用的输入数据。

1.3.1　编码与初始化

编码方法：反映大数据集样本矩阵特征的编码方法可以充分保证启发式算法的性能。采用的编码技术由两部分组成，首先是原始大数据集样本矩阵的映射。Map_{ij} 矩阵表示大数据集样本矩阵 Map 的对角线。$TempMap$ 表示临时大数据集样本矩阵，$TempMap_{ij}$ 表示矩阵对角线，通过该矩阵的每一行显示与该行对应的机器上的数据集样本序列，而矩阵的每一列都显示大数据集样本分类或预测过程中特征的总和。

初始化：大数据集样本矩阵 Map 对角线上的每个单元格取 100，即 $Map_{ij}=100$，表示每个数据中心都在内部维护其 100%的特性选择。

1.3.2 基于萤火虫算法的最优特征集选择

在上述编码方案的基础上，利用小邻域结构和大邻域结构两种不同的邻域搜索结构来寻找最优特征集。在建立第一个搜索结构的邻域的同时，利用萤火虫算法的移动算子选择下一个最优特征集。此外，为了建立第二个搜索结构的邻域，在特征选择问题中引入交换、插入、逆 3 个常用运算符。通过左移操作符，将特性集中的临时特性 F_{temp} 移动到当前特性集中 F_{current}。在萤火虫算法中，通过后续关系确定最优特征 i 向最具吸引力（或更亮）的另一个特征 j 的方向运动，其变化过程描述为

$$x_i = x_i + \alpha_0 e^{-\lambda \cdot d_{ij}^2}(x_j - x_i) + \beta(\varepsilon - 0.5) \tag{6}$$

式中：$\alpha e^{-\lambda \cdot d_{ij}^2}$ 是吸引力函数，其值随两个萤火虫之间的距离 r_{ij} 的增加而减小；α_0 为 $d_{ij}^2 = 0$ 时的吸引力；λ 为环境中的固定光吸收系数；$\varepsilon \in [0, 1]$ 为均匀分布的随机数；β 为随机化参数。

利用笛卡尔距离计算方法，可以得到两个萤火虫 i 和 j 之间的距离

$$d_{ij} = \| x_i - x_j \| = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} \tag{7}$$

式中：x_{ik} 为第 i 个萤火虫的第 k 个组成部分。

通过计算两个特征矩阵之间的距离，即可得到 $TempMap$，然后就可以利用分类指标（查全率、精确度、准确度）计算新的适应度值。一般来说，准确度是指正样例和负样例的总量占总数据量的比例，适应度值的计算方法为

$$\eta = \frac{TN + TP}{TN + TP + FN + FP} \tag{8}$$

式中：TP 为正样例，FN 为漏报，TN 为负样例，FP 为误报，η 为适应度值。

更新过程为：如果新的适应度值低于当前适应度值，则固定新的位置，且初始 F_{current} 被更新为与当前特征矩阵 F_{temp} 等价；否则（即 $F_{\text{current}} \geqslant F_{\text{temp}}$，$F \in [0,1]$），在区间 $[0, 1]$ 内产生随机数 r，当 $r < F$ 时，新位置是一致的，从而使当前的特征矩阵是最新的。

为了建立第二搜索结构的邻域，对经典的插入算子和逆算子进行了改进。在这些运算符中，首先，随机选择一个特征矩阵，并使用相应的数据集矩阵；然后，在相同的概率下，考虑数据集样本，并在选定的特征集上实现算子。在交换运算符的情况下，随机选择两个位置，并相对于所有数据集样本，将两个现有数据集样本特征矩阵的位置互换；对于插入运算符，随机选择一个特征和一个位置，并将所选特征插入整个数据集矩阵中所选的位置；在逆算子的情况下，随机选择序列的两个点，并将这两个特征之间的数据中心位置反演到整个数据集矩阵中；如果 $r \geqslant F$，则否定该解。在每个迭代过程中，算法从数据集样本矩阵、映射中任意选择一行和一列，并临时将选择的单元格设置为零（如果对角线上没有相等的单元格）。

1.3.3 萤火虫算法的改进

为了提高萤火虫算法的收敛速度，引入惯性权重算法。此外，为了解决萤火虫算法的收敛精度，引入正交反向学习算法。通过上述算法的改进，即实现了萤火虫算法在寻优、速度和精度上的综合优化。

为了改善萤火虫算法的收敛速度，需要在萤火虫位置更新公式中引入惯性权重，其表达式为

$$x_i = w x_i + \alpha_0 e^{-\lambda \cdot d_{ij}^2}(x_j - x_i) + \beta(\varepsilon - 0.5) \tag{9}$$

式中：w 为惯性权重。

为了避免陷入局部最优，需要对 w 进行如下设计

$$w = \begin{cases} 0.85 + 0.25\varepsilon, & l < 0.5L \\ 0.35 + 0.25\varepsilon, & l > 0.5L \end{cases} \tag{10}$$

式中：l 为实时迭代次数，L 为迭代次数最大值。由上式可知，当 $l < 0.5L$ 时，w 取值较大，由于 ε 为均匀分布的随机数，因此 w 服从较大的均匀分布；反之，w 服从较大的均匀分布。

反向学习的核心思想是同时评估当前点和其反向点，择优使用，以此来提高搜索精度，反向学习的基本定义见文献 [15]。

为了充分利用群体搜索信息，需要借助重心反向，并以群体重心为参考点计算反向点，具体定义如下。

定义 1 设 $d_i \in R$ 是带有单位质量的点，$i = 1, \cdots, K$，则 K 个点的重心 G_j 定义为

$$G_j = \frac{1}{K} \sum_{i=1}^{K} d_{ij} \tag{11}$$

反向点 $\breve{d}_i$ 的表达式为

$$\begin{cases} \breve{d}_i = 2G - d_i \\ G = \sum_{i=1}^{K} G_i \end{cases} \tag{12}$$

G 为反向点重心，基于正交表的正交反向学习算法的具体算法流程见文献 [15]，此处不再赘述。

1.3.4 并行化 IWOF 算法

利用 MapReduce 模型实现 IWOF 算法的并行化。假设 T 是一个训练集，m 是映射任务的数量。首先，映射归约分割方法将 T 分割成 m 个不相交的实例子集。其次，每个子集 T_i，$i \in \{1, \cdots, m\}$ 由等价映射函数处理，由于这个分区是依次执行的，每个子集的实例数量大致相同，因此 T 文件的随机化保证了类的平衡。

特征选择算法包含每个 T_i 的映射阶段，因此二元向量 $s_i = \{s_{i1}, \cdots,\}$ 表示 IWOF 算法选择了哪些特征。每一个二进制向量在归约阶段取平均值，得到一个式（13）所定义的向量 x，其中 x_j 被称为在线特征选择应用程序在其结果中包含特征 j 的比率，该向量被称为完整在线特征选择学习的结果，用于构造用于附加机器学习原则的缩减数据集

$$x_{ij}=\frac{1}{m}\sum_{i=1}^{m}s_{ij},j\in\{1,\cdots,N\} \tag{13}$$

式中：N 为特征数量。

为了在计算特征向量 x 时以可伸缩的方式尽快从原始数据集中消除不重要的特征，需要采用映射归约模式。通过阈值 θ 实现矢量 x 的二值化，即

$$S=\{s_1,\cdots,s_N\} \tag{14}$$

$$s_j=\begin{cases}1, & x_j\geqslant\theta\\ 0, & x_j<\theta\end{cases} \tag{15}$$

式中：S 为简化数据集选择的特性向量。

对于较高的阈值，所选特征的数量 $N'=\sum_{j=1}^{N}S_j$ 可以得到控制，且只允许选择少量的特征，而较小的阈值则允许选择更多的特征。数据集约简的映射归约模式以这种方式工作，即每个映射处理一个实例，并生成一个新的实例，该实例仅包含 S 中选择的特性。最后，为了创建最终的简化数据集，使用简化阶段将生成的实例连接起来。

1.3.5　IWOF 算法流程

（1）输入特征 F 从原始数据集 $X=(x_{1j},\cdots,x_{nj})$；

（2）计算适应度函数 $f(x)$，其等于分类精度

$$TempMap=(x_{1j},\cdots,x_{nj})^T$$

（3）从数据集样本中生成萤火虫初始种群

$$x_i,(i=1,2,\cdots,n)(n=100)$$

（4）利用适应度函数 $f(x)$ 确定 x_i 处的光强 l_i；

（5）定义光吸收系数 γ：

（6）While $t<100$

（7）For $i=1$：n

（8）For $j=1$：i

（9）If $(I_j>I_i)$ && $(F_{\text{current}}\geqslant F_{\text{temp}})$

萤火虫 I 在 d 维方向上沿 j 方向前进；

吸引力随距离 r 通过 $e^{-\gamma r}$ 发散；

（10）Else　执行交换、插入和逆运算符；

（11）转到步骤（9）；

（12）End For j；

（13）End For i；

（14）对萤火虫进行分类并定位当前的适应度值；

（15）如果某些特性直到 t 小于最大进化代数时才被选中

执行正交反向学习

（16）End While；

（17）处理结果和可视化。

2　利用 SACNN 的数据分类

传统的分类算法难以处理大量的数据。因此，采用 IWOF 算法进行在线特征选择，然后选择分类器对所得特征进行分类。卷积神经网络（convolutional neural network，CNN）分类精度高，是一种应用十分广泛的深度神经网络。它的权值共享网络结构使之更类似于生物神经网络，降低了网络模型的复杂度，减少了权值的数量。

经典的 CNN 算法以较小的计算量对数据进行学习，具有稳定的学习效果，但其非线性处理能力较弱，因此在处理复杂数据时能力稍显不足。

针对 CNN 存在的问题，选择图 2 所示的结构感知卷积神经网络[18]算法实现特征数据的精确分类。图中：$x\in R^{n\times c}$ 为输入，$y\in R^n$ 为输出，$f(\cdot)$ 为功能滤波器，r_{ji} 为第 j 个顶点与第 i 个顶点之间的关系值，$M\in R^{c\times c}$ 为衡量局部顶点之间关系的矢量矩阵，$T(\cdot)$ 为 tanh 函数，x_j、x_i 代表输入 x 的第 j 行、第 i 行的行向量。

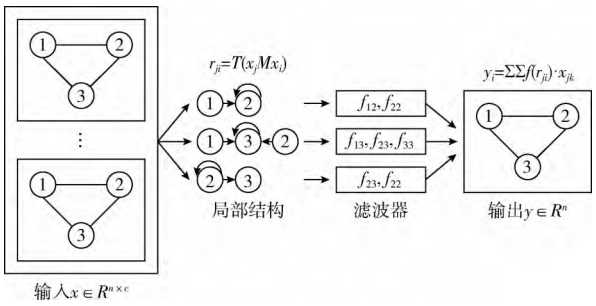

图 2　SACNN 结构

SACNN 的输出为

$$\begin{cases}y_i=\sum\limits_{\varepsilon_{ji}\in E}f(r_{ji})\cdot x_j\\ f(r_{ji})=\sum\limits_{k=1}^{K}v_k\cdot h_k(r_{ji})\\ r_{ji}=T(x_j^TMx_i)\end{cases} \tag{16}$$

式中：ε_{ji} 代表第 i 个顶点的第 j 个临近点；E 为这些点组成的集合；$h_k(\cdot)$ 为切比雪夫多项式；v_k 为多项式中的系数；K 为多项式的阶数。

通过将传统 CNN 中的卷积运算替换为结构识别卷积运算，使得 SACNN 具有非常高的模型学习能力。当将其应用到数据分类中时，可以充分发挥其精确建模能力。图 2 给出了 SACNN 应用的关键步骤及对应的算法，即：将数据集输入 SACNN 后，先计算这些数据之间的关系值，然后再对这些关系值进行滤波处理，最后再利用输入数据和滤波后的关系值计算最终的输出值。

3　实验结果与分析

在 Spark 框架下，为了检验 IWOF 算法的有效性，使用二分类数据集进行数据特征选择与分类实验。数据集选为包含 2000 个数值特征的，数据量为 50 万个样本集合的 Epsilon。此外，包含 631 个特征，数据量为 6600 万个样本的 ECBDL14 数据集。表 1 概述了这些数据集的主要特征。除了属性的数量外，用于训练和测试集的样本数量也在验证 IWOF 算法时进行了描述。在数据样本中，75%的样本

用于培训，25%的样本用于测试。

表 1 数据集概述

数据集	训练实例	测试实例	特征	拆分实例
Epsilon	500 000	100 000	2000	780
ECBDL14	65 003 913	2 897 917	631	1984

在应用映射归约模式支持的 IWOF 算法后，在上述数据集上，分别使用 Spark 中实现的 SACNN、文献［9］和文献［10］这 3 种不同的分类算法对数据集进行分类实验验证。

3.1 评估指标

在处理分类问题时，必须将一个类标记为正类，另一个类标记为负类，分别考虑 p 个阳性样本和 n 个阴性样本的测试集。任何分类器的任务都是为每个样本分配一个类，此外，某些任务可能是不正确的。为了评估分类器的性能，在正样例、负样例、误报和漏报样品的基础上，统计并设计了一个融合矩阵，见表 2。

表 2 融合矩阵

		正	负
分类器输出	正	TP	FP
	负	FN	TN
	行和	p	n

利用表 2 可以推导出用于不平衡学习的性能指标，包括：精度 P、查全率 R、测度 FM 和几何平均 GM，如下所示：

（1）精度 P 定义为检索到的相关实例的百分比，其表达式为

$$P=\frac{TP}{TP+FP}$$

（2）查全率 R 定义为检索到的相关实例的比例，其表达式为

$$R=\frac{TP}{TP+FN}$$

（3）测度 FM 定义为准确度与查全率相结合的测度，即准确度与查全率的调和平均值，其表达式为

$$FM=\frac{2PR}{P+R}$$

（4）几何平均 GM 用来评估不平衡数据集上的分类器，几何平均指定了主流和少数类的分类性能之间的平衡，该指标考虑了敏感性和特异性，敏感性即为查全率，特异性 SP 和几何平均的表达式分别为

$$SP=100-\frac{TP}{TN}$$

$$GM=\sqrt{SP\times R}$$

3.2 与其它方法的对比和分析

图 3 给出了 3 种分类算法在实验数据集分类实验中的精度和查全率结果。与文献［9］和文献［10］中的两类分类算法相比，提出的基于 IWOF 算法的 SACNN 分类算法的分类精度是最高的，其准确率约为 94%，与文献［9］和文献［10］相比，提出的 SACNN 分类算法的准确率分别提高了 5%和 8%。由此可见，提出的算法很好地解决了高维数据集问题，这表明所采用的映射归约模式解决了并行处理的需求。然而，在查全率方面，提出的基于 IWOF 的 SACNN 分类算法虽然高于文献［9］，但明显低于文献［10］，这表明所提算法很难保证数据的学习覆盖率，即其在算法的通用性方面略差于文献［10］。

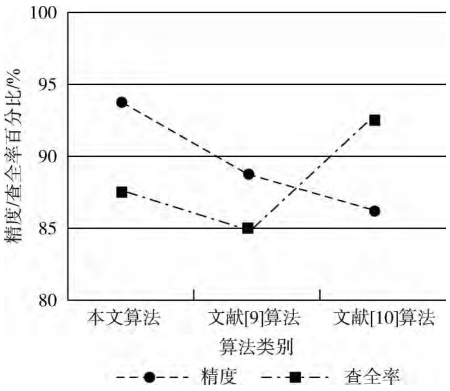

图 3 3 种分类算法的精度与查全率对比

图 4 给出了 3 种分类算法在实验数据集分类实验中的测度和几何平均结果。与文献［9］和文献［10］中的两类分类算法相比，提出的基于 IWOF 算法的 SACNN 分类算法的测度和几何平均结果均是最大的，这表明所提算法在大数据分类中的平均化性能指标方面具有明显的优势，且所提算法在收敛精度上高于文献［9］，但略低于文献［10］。

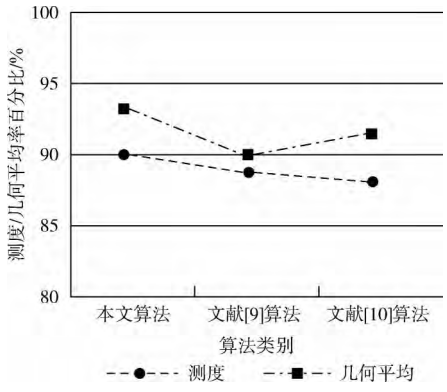

图 4 3 种分类算法的测度与几何平均对比

图 5 给出了 3 种分类算法在实验数据集分类实验中的准确率和错误率结果。与文献［9］和文献［10］中的两类分类算法相比，提出的基于 IWOF 算法的 SACNN 分类算

法的精度是最高的，相应的其错误率则是三者中最低的。由此可见，提出的分类算法的分类准确率是可以得到充分保证的。

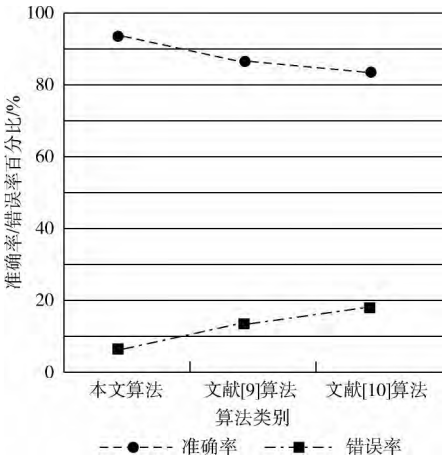

图 5　3 种分类算法的准确率和错误率对比

表 3 给出了以上 3 个实验的具体指标。

表 3　不同分类器的评价指标%

算法	精度	查全率	测度	几何平均	准确性
本文算法	94	87	90	93	92
文献[9]	89	85	88	90	86
文献[10]	86	93	87	91	82

为了进一步验证所提算法与文献［9］、文献［10］中的分类算法在大数据分类速度上的对比结果，图 6 给出了大数据集的训练运行时间，训练运行时间定义为用于训练或构造分类器的时间（以秒为单位）。

图 6 显示了针对 3 个不同分类器绘制的训练运行时间比较结果。文献［9］和文献［10］中的两类分类算法相比，提出的基于 IWOF 算法的 SACNN 分类算法的训练运行时间并不是最低的，即提出的 IWOF 算法的收敛速度介于文献［9］、文献［10］中的算法之间。由此可见，本文所提算法是以牺牲一部分快速性来换取分类准确率的。

4　结束语

高维数据的特征选择与分类对于很多实际的工程问题来说非常重要，提出一种基于惯性权重 OOL-FA 算法的大数据特征选择算法，在此基础上利用 SACNN 算法实现了大数据的精确分类。在实际数据集上进行的实验结果表明了所提算法在准确率、测度、几何平均等方面的优越性。但实验结果也表明，所提大数据分类算法在动态性能和查全率上并不是最优的，这说明所提算法还有一定的提升空间。下一步的研究应该在保证分类精度的基础上，进一步提升分类算法的快速性。

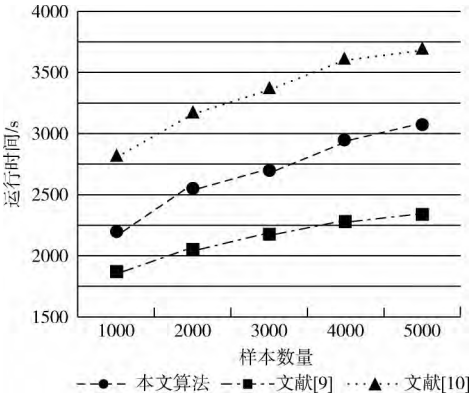

图 6　3 种分类算法的训练运行时间对比

参考文献：

[1] YAN Yi，XU Su. Parallel entropy based FIUT algorithm mining in Hadoop environment［J］. Computer Engineering and Design，2019，40（3）：92-97（in Chinese）.［晏依，徐苏. Hadoop 环境下基于并行熵的 FIUT 算法挖掘［J］. 计算机工程与设计，2019，40（3）：92-97.］

[2] WANG Fayu，LIU Zhiqiang. Optimization method of distributed K-means algorithm in Spark framework［J］. Computer Engineering and Design，2019，40（6）：1595-1600（in Chinese）.［王法玉，刘志强. Spark 框架下分布式 K-means 算法优化方法［J］. 计算机工程与设计，2019，40（6）：1595-1600.］

[3] ZHANG Yuanming，JIANG Jianbo，LU Jiawei，et al. An iterative data equalization partition strategy for MapReduce［J］. Journal of Computer Science，2019，42（8）：1873-1885（in Chinese）.［张元鸣，蒋建波，陆佳炜，等. 面向 MapReduce 的迭代式数据均衡分区策略［J］. 计算机学报，2019，42（8）：1873-1885.］

[4] Fong S，Wong R，Vasilakos A. Accelerated PSO swarm search feature selection for data stream mining big data［J］. IEEE Transactions on Services Computing，2016，9（1）：33-45.

[5] Barbu A，She Y Y，Ding L J，et al. Feature selection with annealing for computer vision and big data learning［J］. IEEE Transactions on Pattern Analysis and Machine Intelligence，2017，39（2）：272-286.

[6] Gallego S R，Talín H M，Rego D M，et al. An information theory-based feature selection framework for big data under apache spark［J］. IEEE Transactions on Systems，Man，and Cybernetics：Systems，2018，48（9）：1441-1453.

[7] Zhao L，Chen Z K，Hu Y M，et al. Distributed feature selection for efficient economic big data analysis［J］. IEEE Transactions on Big Data，2018，4（2）：164-176.

[8] Lv Y X，Peng S C，Yuan Y，et al. A classifier using online

bagging ensemble method for big data stream learning [J]. Tsinghua Science and Technology, 2019, 24 (4): 379-388.

[9] Gallego S R, Krawczyk B, García S, et al. Nearest neighbor classification for high-speed big data streams using spark [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017, 47 (10): 2727-2739.

[10] Duan M X, Li K L, Liao X K, et al. A parallel multiclassification algorithm for big data using an extreme learning machine [J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29 (6): 2337-2351.

[11] ZHANG Yu, BAO Yanke, SHAO Liangshan, et al. Multivariate decision tree for distributed data flow big data classification [J]. Journal of Automation, 2018, 44 (6): 1115-1127 (in Chinese). [张宇, 包研科, 邵良杉, 等. 面向分布式数据流大数据分类的多变量决策树 [J]. 自动化学报, 2018, 44 (6): 1115-1127.]

[12] MAO Guojun, HU Dianjun, XIE Songyan. Big data classification model and algorithm based on distributed data flow [J]. Journal of Computer Science, 2017, 40 (1): 163-177 (in Chinese). [毛国君, 胡殿军, 谢松燕. 基于分布式数据流的大数据分类模型和算法 [J]. 计算机学报, 2017, 40 (1): 163-177.]

[13] SUN Jingtao, ZHANG Qiuyu. Text feature gene extraction method under unbalanced big data set [J]. Journal of University of Electronic Science and Technology, 2018, 47 (1): 125-131 (in Chinese). [孙晶涛, 张秋余. 不均衡大数据集下的文本特征基因提取方法 [J]. 电子科技大学学报, 2018, 47 (1): 125-131.]

[14] Li F, Chen J Y, Wang Z Y. Wireless MapReduce distributed computing [J]. IEEE Transactions on Information Theory, 2019, 65 (10): 6101-6114.

[15] ZHOU Lingyun, DING Lixin, MA Maode, et al. Orthogonal opposition based firefly algorithm [J]. Journal of Electronics and Information, 2019, 41 (1): 202-209 (in Chinese). [周凌云, 丁立新, 马懋德, 等. 一种正交反向学习萤火虫算法 [J]. 电子与信息学报, 2019, 41 (1): 202-209.]

[16] LIU Huijun, SU Hongjun, ZHAO Bo. Multi feature optimization method of hyperspectral remote sensing based on improved firefly algorithm [J]. Remote Sensing Technology and Application, 2018, 33 (1): 110-118 (in Chinese). [刘慧珺, 苏红军, 赵波. 基于改进萤火虫算法的高光谱遥感多特征优化方法 [J]. 遥感技术与应用, 2018, 33 (1): 110-118.]

[17] Phan H, Andreotti F, Cooray N, et al. Joint classification and prediction CNN framework for automatic sleep stage classification [J]. IEEE Transactions on Biomedical Engineering, 2019, 66 (5): 1285-1296.

[18] Chang J L, Gu J, Wang L F, et al. Structure-aware convolutional neural networks [C] //Proceedings of the 32nd Conference on Neural Information Processing Systems, 2018: 1-10.