Journal of Computer Applications
计算机应用,2021,41(6):1679－1685
ISSN 1001-9081
CODEN JYIIDU
2021-06-10
http://www.joca.cn

文章编号:1001-9081(2021)06-1679-07
DOI:10.11772/j.issn.1001-9081.2020091436

基于随机子空间的扩展隔离林算法

谢　雨,蒋　瑜*,龙超奇
(成都信息工程大学 软件工程学院,成都 610225)
(*通信作者电子邮箱jiangyu@cuit.edu.cn)

摘　要:针对扩展隔离林(EIF)算法时间开销过大的问题,提出了一种基于随机子空间的扩展隔离林(RS-EIF)算法。首先,在原数据空间确定多个随机子空间;然后,在不同的随机子空间中通过计算每个节点的截距向量与斜率来构建扩展孤立树,并将多棵扩展孤立树集成为子空间扩展隔离林;最后,通过计算数据点在扩展隔离林中的平均遍历深度来确定数据点是否异常。在离群值检测数据库(ODDS)中的9个真实数据集与呈多元分布的7个人工数据集上的实验结果表明,所提RS-EIF算法对局部异常很敏感,相较EIF算法减少了约60%的时间开销;在样本数量较多的ODDS数据集上,该算法识别精度高出孤立森林(iForest)算法、轻型在线异常检测(LODA)算法和基于连接函数的异常检测(COPOD)算法2~12个百分点。RS-EIF算法在样本数量大的数据集中识别效率更高。

关键词:异常检测;随机子空间;扩展隔离林算法;扩展孤立树;平均遍历深度
中图分类号:TP181　　**文献标志码:**A

Extended isolation forest algorithm based on random subspace

XIE Yu, JIANG Yu*, LONG Chaoqi
(*School of Software Engineering, Chengdu University of Information Technology, Chengdu Sichuan* 610225, *China*)

Abstract: Aiming at the problem of excessive time overhead of the Extended Isolation Forest (EIF) algorithm, a new algorithm named Extended Isolation Forest based on Random Subspace (RS-EIF) was proposed. Firstly, multiple random subspaces were determined in the original data space. Then, in each random subspace, the extended isolated tree was constructed by calculating the intercept vector and slope of each node, and multiple extended isolated trees were integrated into a subspace extended isolation forest. Finally, the average traversal depth of data point in the extended isolation forest was calculated to determine whether the data point was abnormal. Experimental results on 9 real datasets in Outliter Detection DataSet (ODDS) and 7 synthetic datasets with multivariate distribution show that, the RS-EIF algorithm is sensitive to local anomalies and reduces the time overhead by about 60% compared with the EIF algorithm; on the ODDS datasets with many samples, its recognition accuracy is 2 percentage points to 12 percentage points higher than those of the isolation Forest (iForest) algorithm, Lightweight On-line Detection of Anomalies (LODA) algorithm and COPula-based Outlier Detection (COPOD) algorithm. The RS-EIF algorithm has the higher recognition efficiency in the dataset with a large number of samples.

Key words: anomaly detection; random subspace; Extended Isolation Forest (EIF) algorithm; extended isolated tree; average traversal depth

0　引言

大数据时代的到来,使得海量的数据被收集和存储在数据库中,从体量巨大的数据中挖掘可理解的知识显得更加有意义。作为数据挖掘工作中重要的一环,异常检测算法正在蓬勃发展[1]。越来越多的异常点检测算法正在进一步被运用于日常生活的方方面面[2]。

近年来,国内外学者研究提出了多种异常检测算法。文献[3]中对这些算法进行了分类总结。按照异常识别技术分为:以基于角度的离群值检测(Angle-Based Outlier Detection, ABOD)算法[4]与基于连接函数的异常检测(COPula-based Outlier Detection, COPOD)算法[5]为代表的基于统计的方法;以K最近邻(K-Nearest Neighbor, KNN)分类算法[6]为代表的基于距离的方法;以局部异常因子(Local Outlier Factor, LOF)算法[7]为代表的基于密度的方法;以及以自编码器(Autoencoder)[8]为代表的基于学习的方法等。除此之外,基于集成的方法同样也在不断地被研究,其中具有代表性的技术包括:Bagging[9]与Boosting[10]抽样等。文献[11]中提出了一种基于隔离思想[12]的集成学习算法[1]:孤立森林(isolation Forest, iForest)算法,该算法首先构建了一个由多棵孤立树组成的孤立森林,随后将待检测数据点在孤立森林中进行遍历,记录该数据点被完全隔离的平均路径长度并生成对应的异常分数。文献[13]中指出由于iForest算法的核心思想为隔离,

收稿日期:2020-09-15;**修回日期:**2020-11-27;**录用日期:**2020-11-30。
作者简介:谢雨(1996—),男,四川内江人,硕士研究生,主要研究方向:数据挖掘、智能计算、异常检测;　蒋瑜(1980—),男,四川邻水人,副教授,硕士,主要研究方向:入侵检测、粗糙集、数据挖掘、智能计算;　龙超奇(1996—),男,四川德阳人,硕士研究生,主要研究方向:数据挖掘、智能计算、小波聚类。

且每棵孤立树通过随机选择的特征值来进行隔离，所以该算法具有计算速度快、准确度高与内存占用低等优点。

传统方法在各个领域内被广泛应用，可以通过不同方法的结合产生性能更优的异常检测模型。文献[14]中结合LOF算法的最近邻思想与iForest算法的隔离思想，提出使用最近邻集合进行隔离(isolation using Nearest Neighbor Ensemble，iNNE)的异常检测算法，使用最近邻隔离超球体来隔离目标空间中的数据。该算法是一种基于隔离思想的高精度集成学习算法，具有精度高的优点，但在样本数量巨大的数据集中，时间开销太大。文献[15]中结合多粒度扫描机制，提出了基于多粒度随机超平面的孤立森林(Multi-dimensional Random Hyperplane iForest，MRHiForest)算法，该算法在多个小于原始数据空间的k维空间内分别构建孤立森林，通过多个森林集成投票机制，构成层次化集成学习的异常检测模型。该模型提高了在高维数据集上进行异常检测工作的稳定性与准确性，但存在精度下降、时间开销增加等缺点。

文献[16]中指出，iForest算法虽然效率与精度很高，但由于隔离条件通常为轴平行的，轴平行的隔离条件必然会导致隔离平面的交叉，进而产生呈规则分布的异常分数不准确区域。文献[17]中进一步指出，在高维数据空间中使用iForest算法将导致类似的异常分数不准确区域大量存在，因此将这种问题定义为“局部异常不敏感问题”，并在最后提出了扩展隔离林(Extended Isolation Forest，EIF)算法，该算法则完全解决了孤立森林算法对局部异常不敏感的问题。但由于EIF算法在扩展孤立树的每一个节点上进行隔离计算时，都需要进行一次向量点乘运算，所以在预测中高维数据时，其时间开销往往远大于iForest算法。

综上所述，iForest算法对局部异常不敏感的问题会导致部分异常点无法被准确检测，而EIF算法虽然解决了局部异常不敏感的问题，但由于该算法的时间开销太大，在高速发展的当下并不适用。

因此，本文结合子空间思想提出了基于随机子空间的扩展隔离林(Extended Isolation Forest based on Random Subspace，RS-EIF)算法。该算法从数据空间中随机选择维度构建子空间，并使用随机超平面来避免轴平行隔离条件下隔离条件重合区域的产生。本文算法在解决iForest算法对局部异常不敏感问题的同时，相较于EIF算法减少了计算开销。

1 相关工作

1.1 iForest算法

1.1.1 孤立森林的构建

设数据集$D=\{\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n\}$且$D\subset\mathbf{R}^u$，其中$\mathbf{R}^u$为实数集，$u$为最高维度，$n$为数据个数，$\boldsymbol{d}_i=\{x_{i1},x_{i2},\cdots,x_{iu}\}$，其中$i\in[0,n]$。

文献[12]定义孤立森林的构建过程为：首先，在数据集D中随机抽取η次，每次确定ψ个样本作为训练集；然后，在每次抽样完成后，以$\mathrm{lb}\psi$为高度限制构建一棵孤立树；最后，在构建η棵孤立树后，集成为一个孤立森林。

在每一个孤立森林中，包含的孤立树均为二叉树结构，下面给出孤立树的定义。

定义1 孤立树。从$[1,u]$中随机确定一个整数p作为随机选择的维度，统计维度p的范围$[p_{\min},p_{\max}]$，并在该范围下随机确定隔离条件q。当训练集ψ中的数据点在p维度的值小于q时，将该数据确定为节点的左子树节点；反之确定为右子树节点。递归构建一棵高度限制为$\mathrm{lb}\psi$的二叉搜索树。

完成构建孤立森林后，得到一个基于树形结构的集成学习模型[18]。

1.1.2 孤立森林的缺点

在孤立森林中，数据点的异常程度通常由数据点在每棵孤立树中遍历的深度决定。这里的深度指的是数据点在空间中被划分到无法继续划分的子空间时所需的划分次数h。使用数据$\boldsymbol{d}_k$在具有η棵孤立树的孤立森林中进行遍历，得到路径深度集合$H(\boldsymbol{d}_k)=\{h_1,h_2,\cdots,h_\eta\}$，并求得平均深度$E(h)$。最后，将平均深度值通过式(1)归一化处理为一个取值范围为0~1的异常分数$s(\boldsymbol{d}_k,n)$。

$$s(\boldsymbol{d}_k,n)=2^{-(E(h)/c(n))} \tag{1}$$

式中，$c(n)$为归一化因子，被定义为搜索失败的平均路径[12]。式(2)为$c(n)$的具体计算方式：

$$c(n)=2H(n-1)-2(n-1)/n \tag{2}$$

其中，$H(i)$为调和级数，可以通过$\ln(i)+0.577\,215\,664\,9$(欧拉常数)确定。

数据并不只是单一地分布在一个聚类中心，真实的数据往往具有多个聚类中心[19]。由于iForest算法的隔离条件是轴平行的，在多元数据集中，算法会因隔离条件的重叠使得局部异常难以被检测到，从而导致算法精度下降。针对该问题，提出了基于随机斜率构建超平面的EIF算法[17]。

1.2 EIF算法

1.2.1 EIF算法相关定义

EIF算法基于iForest算法进行改进，将轴平行的隔离条件替换为具有随机斜率[20]的超平面。

定义2 随机斜率。在二维空间中，随机超平面可以理解为线，而线段是具有斜率的，其斜率可以用平面内随机一个点到原点的向量表示。同理，在k维的高维空间中，其随机斜率可以表示为空间内的向量$\boldsymbol{j}=(i_1,i_2,\cdots,i_k)$，其中$i\in[0,1)$且随机生成。

本文将一个N维空间定义为一个具有N条坐标轴的空间。若存在一个超平面与N条坐标轴中的一条相交，则称该超平面的扩展程度为0。以二维空间为例，在二维空间中，当扩展等级为0时，隔离超平面可以理解为一条平行于坐标轴的直线。由于超平面需要由斜率与截距确定，下面给出自定义隔离等级的随机截距向量与隔离超平面的定义。

定义3 自定义隔离等级的随机截距向量。在维度为k的空间中，存在$\boldsymbol{s}=(n_1,n_2,\cdots,n_k)$，当隔离等级为$k-1$时，$n_1,n_2,\cdots,n_k$均为不为0的实数。

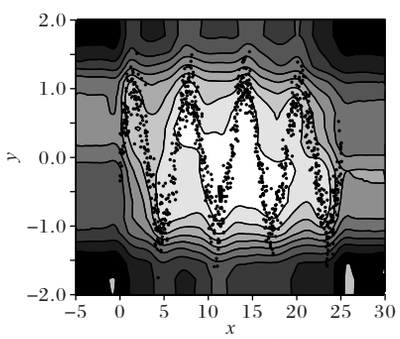
定义4 隔离超平面。在EIF算法中，通过随机生成超平面的随机斜率$\boldsymbol{j}$与具有自定义隔离等级的存在于超平面上的随机截距$\boldsymbol{s}$可以确定一个唯一的隔离超平面。

在确定隔离超平面后，隔离条件确定为：$(\boldsymbol{d}-\boldsymbol{s})\cdot\boldsymbol{j}$，其中$\boldsymbol{d}$表示来自数据集$D$中的一个待检测数据点。当隔离条件的值小于0时，将$\boldsymbol{d}$划分到当前根节点的左子树，反之划分到当前节点的右子树。

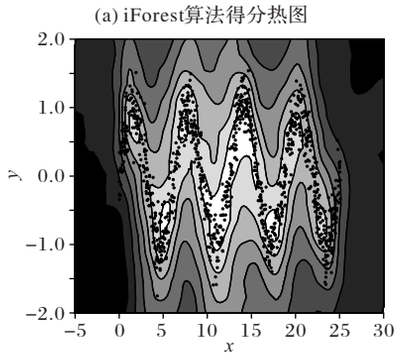
1.2.2　EIF 算法的优势

为了展示由于隔离平面轴平行导致的精度下降的问题,本节首先使用基于正弦函数分布的二维数据集来计算各个点的异常分数,并使用热图分别绘制 EIF 算法与 iForest 算法对该数据集进行分析后的得分分布。实验数据集为随机在正弦曲线两侧呈高斯分布的 1 000 个二维数据点。随后,对所得的数据分别使用 EIF 算法与 iForest 算法进行预测后,将所得到的异常分数划分为 10 个不同的等级,为每个等级标注边界,并使用热图绘制边界。两种算法的得分热图如图 1 所示。

iForest 算法得分热图如图 1(a)所示,而 EIF 算法得分热图则如图 1(b)所示。对比两个图可以发现,正弦函数的任意两个波峰或波谷之间,EIF 算法计算得到的异常分数相较于 iForest 算法异常分数更加具有层次感;而且在 EIF 算法的得分热图中,在数据分布外并没有存在矩形的得分异常区域。因此,EIF 算法可以更好地计算处在正弦函数任意两个波峰或波谷之间局部数据点的异常得分。



(a) iForest算法得分热图



(b) EIF算法得分热图

图 1　正弦分布数据上不同算法得分热图

Fig. 1　Score heat maps of different algorithms on sine distribution data

1.2.3　EIF 算法的劣势

EIF 算法核心为使用随机隔离超平面进行隔离。假设使用维度为 k 的数据集构建一个包含 η 棵孤立树的扩展隔离林,其中,每棵树均为平衡二叉树且包含有 256 个节点,每棵树的扩展等级均设定为 $k-1$。现存在随机生成的数据点 $\boldsymbol{m}=(x_1,x_2,\cdots,x_k)$,要想计算出该点的异常分数,在 iForest 森林中,最多只需要进行 8η 次、最少仅需要 η 次比较运算即可得出数据点 $\boldsymbol{m}$ 的遍历深度。

在 EIF 森林中,每个节点上需要进行 1 次向量减法与 1 次向量乘法运算(在 k 维数据集中,需要进行 k 次减法、k 次加法与 k 次乘法运算),即:最多需要 8η 次、最少 η 次向量运算。虽然在 EIF 算法中的计算时间总体是呈线性增长的[16],但相较于 iForest 算法,EIF 算法的时间成本过高。

2　RS-EIF 算法

首先,引入子空间思想,在介绍子空间思想在 EIF 算法中的应用后,结合集成学习的优点,使用随机子空间思想完成算法改进。然后,分析改进后算法 RS-EIF 的时间复杂度与空间复杂度。

2.1　随机子空间优化

根据多粒度思想[15],本文将子空间定义为维度小于目标空间并存在于目标空间中的一个空间,此处的目标空间被定义为数据集所在的高维空间[21]。由此可知,扩展等级为 $k-1$ 的 EIF 算法可以理解为在维度为 $k-1$ 的子空间中构建维度为 k 的随机斜率向量与随机截距向量,其中 k 为目标空间维度。因此,EIF 算法可以理解为在一个完整的数据空间中,以子空间思想对数据进行隔离。

在文献[17]中,随着 EIF 算法扩展等级的提高,算法对局部异常的敏感性也随之提升。因此,为了在实验中获得最高的精确度,扩展等级往往需要设置为 $k-1$。

然而,在最高扩展等级的子空间中构建扩展隔离森林虽然可以得到一个高精度的异常检测模型,但是随之而来的是大量上升的运算成本。因此考虑在更小的子空间中构建扩展隔离森林来提高运算速度。

EIF 算法本身是一种集成学习算法,并包含多个弱分类器。在 EIF 算法中,虽然一棵扩展隔离树的预测结果并不令人满意,但在组合多棵树构成森林后,其预测结果变得优于多数传统的异常检测算法。

因此,本文在随机子空间 l 中生成具有扩展等级的随机斜率 $\boldsymbol{j}'$ 与随机截距向量 $\boldsymbol{s}'$,其中 $l<k$。再将隔离条件优化为 $\boldsymbol{d}\cdot\boldsymbol{j}'<\boldsymbol{s}'\cdot\boldsymbol{j}'$ 来确定数据点是否存在于 l 维子空间中的隔离超平面内。由此方法构建的树结构命名为随机扩展隔离树 RS-Tree(Random Subspace Tree)。

由于 RS-Tree 的精度会低于原本的扩展隔离树,因此再通过构建多棵基于随机子空间的 RS-Tree 组成一个完整的模型来提高算法的精度。下面,给出构建 RS-EIF 森林的伪代码如算法 1 所示。

算法 1　rsForest($X,t,\psi,k,extendlevel$)。

输入　数据集 X,树的棵数 t,每棵树包含的节点个数 ψ,随机子空间维度 k,树的扩展等级 $extendlevel$;

输出　包含 t 棵 RS-Tree 的 RS-EIF 森林。

```
1)  初始化森林
2)  设置孤立树高度限制为 l = ceiling(lbψ)
3)  For i to t do
4)      X' ← sample(X,ψ)        #从数据集中随机抽取t个子样本
5)      rsForest ← rsForest ∪ rsTree(X', 0, l, k, extendlevel)
6)  End For
```

算法 1 需要构建 RS-EIF 森林,该过程与 EIF 算法中的扩展隔离树构建过程类似。而森林中是包含多棵树的,因此给出树的构建伪代码如算法 2 所示。

算法 2　rsTree($X,e,l,k,extendlevel$)。

输入　数据集 X,当前树的高度 e,树的高度限制 l,子空间维度 k,树的扩展等级 $extendlevel$;

输出　1 棵 RS-Tree。

1） 确定k个维度:$attrk$
2） If $e \geqslant l$ or sizeof$(X) \leqslant 1$ then
3） return $rs\_exNode\{Size \leftarrow \text{sizeof}(X)\}$ #若树达到高度限制或数据集样本不足,则返回叶子节点rs_exNode
4） Else
5） 将数据集X转化为只含有$attrk$属性的新数据
6） 根据扩展等级$extendlevel$随机生成维度为k的斜率向量$\boldsymbol{j}'$,其中每个特征值均符合高斯分布
7） 从训练集对应的k维的取值范围中,随机选择维度为k的截距向量$\boldsymbol{s}'$,其中每个特征值均存在于$attrk$的取值范围内
8） 确定隔离值$attr_value$
9） $X_l \leftarrow \text{filter}(X, X \cdot \boldsymbol{j}' < attr\_value)$
10） $X_r \leftarrow \text{filter}(X, X \cdot \boldsymbol{j}' \geqslant attr\_value)$
11） return $rs\_inNode(\text{rsTree}(X_l, e+1, l, k, extendlevel), \text{rsTree}(X_r, e+1, l, k, extendlevel), \boldsymbol{j}', \boldsymbol{s}', attrk, k)$
12） End If

在一个完整的RS-EIF森林中,要想实现对数据的决策判断,需要将该数据在森林中的每棵树进行遍历,计算其平均路径长度,最终给出判断结果。下面给出计算遍历深度的伪代码如算法3所示。

算法3 PathCount($\boldsymbol{d}$,$tree$,e,k)。
输入 数据点$\boldsymbol{d}$,RS-Tree的节点$tree$,当前路径长度e,随机子空间维度k;
输出 数据点$\boldsymbol{d}$的路径遍历长度。

1） If 节点的类型为$exNode$ then
2） return $e + c(tree.size)$
3） End If
4） 计算$\boldsymbol{d}'$ #$\boldsymbol{d}'$为数据点$\boldsymbol{d}$在不同子空间中的属性值
5） $\boldsymbol{j}' = tree \cdot \boldsymbol{j}'$ #取出对应节点的斜率向量
6） If $\boldsymbol{d}' \cdot \boldsymbol{j}' < tree.attr\_value$ then
7） return PathCount($\boldsymbol{d}$, $tree.left$, $e+1$, k)
8） Else If $\boldsymbol{d}' \cdot \boldsymbol{j}' \geqslant tree.attr\_value$ then
9） return PathCount($\boldsymbol{d}$, $tree.right$, $e+1$, k)
10） End If

最终,在计算出目标数据点的平均路径长度(平均深度)后,再使用式(1)处理为对应的异常分数。异常分数越高,则数据的异常程度就越高。

2.2 时间与空间开销分析

2.2.1 时间开销分析

假设在n个维度为K的数据集中,构建包含t棵RS-Tree的RS-EIF森林,每棵树包含ψ个节点,取子空间大小为k,扩展等级为$k-1$。其中,$0 < k < K$。

在训练阶段,需要进行t次随机抽样。由于t是常数,因此其时间复杂度为$O(1)$。在构建RS-Tree时,每棵树只需要进行一次随机的k维选择即可确定子空间,再在子空间中进行ψ次随机生成所需的两个向量并计算不同节点隔离值的工作。因此,一棵完全的RS-Tree只需要常数时间$O(\psi)$即可完成构建。对于包含t棵树的RS-EIF森林,其训练过程时间复杂度为$O(\psi t)$。

在测试阶段,一棵完全RS-Tree中,最多只需要8次向量乘法运算与比较运算,由于k维常量,可以近似地等于$O(1)$。则在t棵RS-Tree进行遍历,时间复杂度为$O(t)$。因此,训练阶段的时间复杂度取决于样本个数。对于n个样本,时间复杂度则等同于$O(tn)$。由于t是一个需要指定的参数,该参数通常为常数,因此时间复杂度可以理解为$O(n)$。

2.2.2 空间开销分析

由于RS-EIF算法的目的是生成一个强分类器用于判断输入信息是否是异常的,因此,算法只需要存储这个包含t棵RS-Tree的数据结构以及其包含的信息。因此空间复杂度为常数。

3 实验与结果分析

为了验证RS-EIF算法的有效性,本文使用3组实验来分别验证该算法的准确性与时间提升。

实验中,使用的人工数据集共有7个,这些人工数据集均使用Python自带的sklearn包生成,每个数据所包含的数据点都是呈高斯分布均匀地分布在4个聚类中心周围。其中,DS_One、DS_Two、DS_Three与DS_Four均为二维数据,其样本数量以2 000的步长递增;而DS_Five、DS_Six与DS_Seven则为包含4 000个样本的数据集,并且其维度以步长2增长。在这些数据集中,异常数据由均匀分布在其他聚类中心周围的10个数据代替,具体的信息如表1所示。

表1 实验使用的人工数据集

Tab. 1 Synthetic datasets used in experiments

数据集	样本数	维度	聚类中心	离群值占比/%
DS_One	2 000	2	4	0.50
DS_Two	4 000	2	4	0.25
DS_Three	6 000	2	4	0.17
DS_Four	8 000	2	4	0.12
DS_Five	4 000	4	4	0.25
DS_Six	4 000	6	4	0.25
DS_Seven	4 000	8	4	0.25

其次,实验使用的真实数据集来自离群值检测数据库(Outliter Detection DataSet, ODDS)[22],本文选择其中9个具有代表性的多维点数据集进行实验。选取的数据集信息如表2所示。这些数据包括低维到高维、低样本数量到高样本数量的数据,可以更好地展现本文算法的性能。

表2 实验使用的ODDS数据集

Tab. 2 ODDS datasets used in experiments

数据集	样本数	维度	离群值占比/%
Lympho	148	18	4.10
Glass	214	9	4.20
Arrhythmia	452	274	15.00
Satimage-2	5 803	36	1.20
Annthyroid	7 200	6	7.42
Mnist	7 603	100	9.20
Shuttle	49 097	9	7.00
ForestCover	286 048	10	0.90
Http	567 479	3	0.40

最后,所有实验使用Python3.8.3编程实现,运行在Windows 10操作系统,12 GB内存,Intel Core i5-4200H CPU @ 2.90 GHz的计算机平台上。由于实验中所涉及的5种算法中有3种均为基于树型结构的集成学习算法,因此为了更加清晰直接地展现对比结果,将3种算法的部分参数默认设置为训练100棵包含256个节点的树。

3.1　局部异常检测

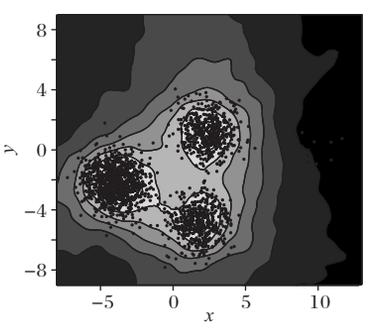
本节选取DS_One为实验数据集。在该数据集中,存在2个聚类中心的距离偏近,因此视觉上产生了类似于存在3个聚类中心的效果。而异常点则明显远离数据集中的大量数据。

根据前文分析,如果在该数据集上使用传统的iForest算法,则必定会产生至少一个矩形的得分异常区域。因此,本节使用与前文类似的方式绘制EIF算法与RS-EIF算法的异常得分热图来观察是否有效避免了矩形异常区域的产生。

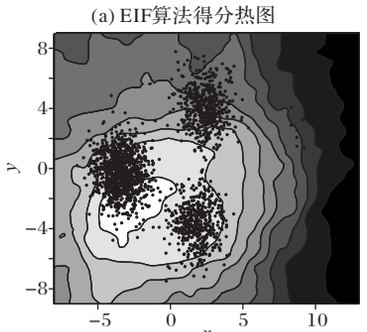
在参数设置方面,由于DS_One是一个二维数据集,为了避免RS-EIF算法在维度为1的子空间出现隔离条件失效的问题,本节将子空间的维度设置为2。除此之外,两种算法的扩展等级均设置为最高等级。实验结果如图2所示。

在图2(a)中清晰地展示出:EIF算法的异常分数热图中,并没有存在呈矩形、由隔离条件重叠所导致的分数异常区域。在图2(b)中,同样没有发现类似的矩形区域。然而,图2(b)的异常分数轮廓的平滑程度并没有达到图2(a)的程度。分析后发现,导致该现象出现的原因在于:本节并没有刻意地调整参数设置以达到最佳效果。

综上所述,不难发现,RS-EIF算法具有与EIF算法相似的局部异常检测能力。



(a) EIF算法得分热图



(b) RS_EIF算法得分热图

图2　多元分布数据上不同算法得分热图

Fig. 2　Score heat maps of different algorithms on multivariate distribution data

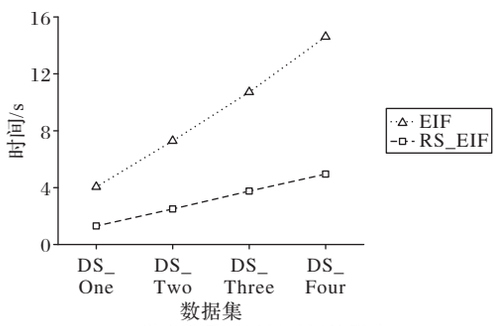
3.2　维度与样本数量对算法运行时间的影响

本节首先使用RS-EIF算法与EIF算法对DS_One、DS_Two、DS_Three与DS_Four这4个数据集进行预测。除了默认参数外,同样指定RS-EIF算法在维度为2的子空间进行训练,两种算法的扩展等级均设置为最高等级,并且采取10次实验求平均值的方法,得到在不同人工数据集的运行时间,实验结果如图3所示。

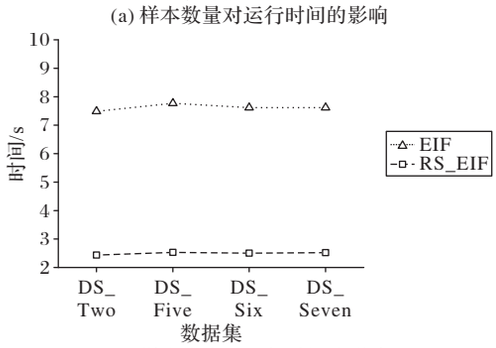
图3展示了在不同样本数据量的情况下,两种算法在运行时间方面的差异。在图3(a)中,所绘制的两条线分别代表两种算法在不同数据集的运行时间。不难看出,虽然两条折线都是近似线性增加的,但RS-EIF算法的运算时间明显少于EIF算法。出现该现象的原因是:RS-EIF算法在预测阶段,每个节点的遍历只需要计算1次随机斜率向量与数据点的点乘;而EIF算法在预测阶段的节点遍历则需要分别计算随机截距向量与数据点同随机斜率向量的点乘。

除此之外,本节还选取了DS_Two、DS_Five、DS_Six与DS_Seven数据比较数据维度对两种算法运行时间的影响。由于上述4个数据的维度是不同的,因此在两种算法中,扩展等级均设置为最高等级。而RS-EIF算法的子空间维度则分别设置为:2、3、5、7。

通过图3(b)发现,两种算法在不同维度的数据集中,运行时间均在一个固定范围内上下变化,但RS-EIF算法的运行时间同样远少于EIF算法。出现该现象的原因在于:所选取的数据集样本数量并不大,没有完全体现算法在不同维度数据集上的运行时间差异。



(a) 样本数量对运行时间的影响



(b) 样本维度对运行时间的影响

图3　数量与维度对运行时间的影响

Fig. 3　Impact of quantity and dimension on running time

综上所述,在相同维度的数据集中,虽然EIF算法与RS-EIF算法时间开销均表现为随样本数量的增加而线性增加;但在样本数量相同的数据集中,EIF算法的时间开销是远多于RS-EIF算法。

3.3　与其他异常检测算法的比较

本节将两个样本数量较多的数据集ForestCover与Http按照1:9划分为测试集与训练集;将两个样本数量过少的数据集Glass与Lympho采用10折交叉验证求均值的方法进行实验;其余数据则使用5折交叉验证取平均值的方法获得结果。

算法选择方面,将RS-EIF算法与同为集成模型的iForest算法、EIF算法与轻型在线异常检测(Lightweight On-line

Detection of Anomalies, LODA)算法[23]以及基于统计的COPOD算法[5]进行对比,这些模型均使用PyOD库[24]实现。参数设置方面,iForest算法使用默认参数;而EIF算法与RS-EIF算法的扩展等级则均设置为最高;除此之外,为了方便计算,RS-EIF算法的子空间维度设置为对应数据空间维度的一半;LODA算法的直方图数量与随机消减数则分别设置为文献[23]中提出的10与100。

表3给出了5种算法在时间开销与精确度方面的差异。其中:时间是指从开始训练到结束预测的时间;精确度(ACcuracy score, AC)则表示使用sklearn包中的相应函数计算5种算法预测结果的准确度。通过表3可以发现,在样本数量大于1 000的数据集中,RS-EIF算法、EIF算法的精确度与iForest算法、LODA算法、COPOD算法相比分别高出了2~12个百分点与3~15个百分点,但在时间开销上,两种算法则均劣于其他3种算法。除此之外,RS-EIF算法与EIF算法在这些数据集上的精确度均相差不超过5个百分点,而RS-EIF算法的运行时间却远少于EIF算法。RS-EIF算法与EIF算法的精确度远高于其他算法的原因是两种算法均使用随机超平面进行隔离,可以有效识别出绝大部分异常点。因此,RS-EIF算法相较于EIF算法的改进是行之有效的。而在两个样本数量过少的数据中,RS-EIF算法的精确度虽然低于EIF算法,但与其他3种算法对比可以发现,其精确度还是明显高于iForest算法的,导致出现该现象的原因则可能是样本数量过少。

综合分析实验结果可知,RS-EIF算法的时间开销在各类型的数据集上均少于EIF算法,具体约下降60%。在实验精确度上,RS-EIF算法与EIF算法相差不大,但在中大型数据集中却明显优于iForest算法、LODA算法与COPOD算法。除此之外,由于RS-EIF算法需要在子空间中进行训练与预测,而该过程包含大量的随机运算,因此,即使集成学习的方式提高了RS-EIF算法的精确度,但在少数数据集上,RS-EIF算法与EIF算法相比还是略有不足。

表3 RS-EIF算法与其他算法的耗时和AC比较结果

Tab. 3 Comparison results of time consumption and AC between RS-EIF algorithm and other algorithms

数据集	RS-EIF		EIF		iForest		LODA		COPOD	
	耗时/s	AC	耗时/s	AC	耗时/s	AC	耗时/s	AC	耗时/s	AC
Lympho	0.056	0.789	0.659	0.960	0.171	0.527	0.039	0.847	0.068	0.967
Glass	0.057	0.862	0.404	0.957	0.164	0.743	0.036	0.912	0.049	0.860
Arrhythmia	0.309	0.872	1.867	0.854	0.291	0.858	0.059	0.875	0.872	0.857
Satimage-2	1.067	0.952	4.874	0.987	0.422	0.829	0.151	0.908	0.573	0.916
Annthyroid	1.274	0.911	4.341	0.925	0.234	0.871	0.095	0.853	0.090	0.870
Mnist	1.423	0.882	6.715	0.901	1.144	0.808	0.165	0.865	1.354	0.848
Shuttle	9.128	0.991	32.853	0.928	0.854	0.952	0.635	0.876	0.738	0.969
ForestCover	27.004	0.931	94.442	0.979	3.838	0.853	4.979	0.901	8.772	0.901
Http	43.189	0.996	151.169	0.995	4.452	0.884	5.518	0.929	6.566	0.904

4 结语

本文从iForest算法的隔离条件轴平行导致算法对局部异常不敏感入手,借鉴EIF算法解决该问题的思想,提出在随机子空间中使用随机超平面进行隔离的RS-EIF算法。该算法从减少向量计算维度的角度入手,充分利用随机的不确定性与集成学习提高弱分类器分类效果的特质,使得性能更加稳定。

实验结果验证了RS-EIF算法存在精度高、时间开销呈线性增加并且明显少于EIF算法的特点。在真实的ODDS数据集上,将RS-EIF算法与其他4种算法分别对比了精确度与时间开销的差异,得出结论为:在样本数量更大的数据集中,RS-EIF算法的精确度与EIF算法相近,但相较iForest算法、LODA算法与COPOD算法来说,RS-EIF算法的精确度平均提升约为5个百分点;在时间开销方面,该算法较EIF算法减少约60%,但与其他3种算法相比,时间开销并不占优势。因此,本文所提RS-EIF算法是一种适用于中大型多元数据集的高精度异常检测算法。

由于本文算法在计算随机超平面时离不开向量的点乘计算,因此时间开销还是偏大。下一步,可以结合集成学习中,各个弱分类器之间可以不存在耦合关系的特点,将该算法部署在分布式系统上,通过合理的任务规划,提高本文算法的运行速度与精确度。

参考文献 (References)

[1] 陈斌,陈松灿,潘志松,等. 异常检测综述[J]. 山东大学学报(工学版),2009,39(6):13-23. (CHEN B, CHEN S C, PAN Z S, et al. Survey of outlier detection technologies [J]. Journal of Shandong University (Engineering Science), 2009, 39(6): 13-23.)

[2] CHANDOLA V, BANERJEE A, KUMAR V. Anomaly detection: a survey [J]. ACM Computing Surveys, 2009, 41(3): Article No. 15.

[3] WANG H, BAH M J, HAMMAD M. Progress in outlier detection techniques: a survey [J]. IEEE Access, 2019, 7: 107964-108000.

[4] KRIEGEL H P, SCHUBERT M, ZIMEK A. Angle-based outlier detection in high-dimensional data [C]// Proceedings of the 2008 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2008: 444-452.

[5] LI Z, ZHAO Y, BOTTA N, et al. COPOD: copula-based outlier detection [C]// Proceedings of the2020 IEEE International Conference on Data Mining. Piscataway: IEEE, 2020: 1118-1123.

[6] RAMASWAMY S, RASTOGI R, SHIM K. Efficient algorithms for mining outliers from large data sets [J]. ACM SIGMOD Record, 2000, 29(2): 427-438.

[7] BREUNIG M M, KRIEGEL H P, NG R T, et al. LOF: identifying density-based local outliers [C]// Proceedings of the 2000 ACM

SIGMOD International Conference on Management of Data. New York: ACM, 2000: 93-104.

[8] CHEN J, SATHE S, AGGARWAL C, et al. Outlier detection with autoencoder ensembles [C]// Proceedings of the 2017 SIAM International Conference on Data Mining. Philadelphia: SIAM, 2017: 90-98.

[9] LAZAREVIC A, KUMAR V. Feature bagging for outlier detection [C]// Proceedings of the 2005 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2005: 157-166.

[10] RAYANA S, AKOGLU L. Less is more: building selective anomaly ensembles [J]. ACM Transactions on Knowledge Discovery from Data, 2016, 10(4): Article No. 42.

[11] LIU F T, TING K M, ZHOU Z. Isolation forest [C]// Proceedings of the 2008 8th IEEE International Conference on Data Mining. Piscataway: IEEE, 2008: 413-422.

[12] LIU F T, TING K M, ZHOU Z. Isolation-based anomaly detection [J]. ACM Transactions on Knowledge Discovery from Data, 2012, 6(1): Article No. 3.

[13] 杨先圣,姜磊,彭雄,等. 基于大数据的异常检测方法研究[J]. 计算机工程与科学,2018,40(7):1180-1186. (YANG X S, JIANG L, PENG X, et al. A new outlier detection method based on large data [J]. Computer Engineering and Science, 2018, 40(7): 1180-1186.)

[14] BANDARAGODA T R, TING K M, ALBRECHT D, et al. Isolation-based anomaly detection using nearest-neighbor ensembles [J]. Computational Intelligence, 2018, 34(4): 968-998.

[15] 杨晓晖,张圣昌. 基于多粒度级联孤立森林算法的异常检测模型[J]. 通信学报,2019,40(8):133-142. (YANG X H, ZHANG S C. Anomaly detection model based on multi-grained cascade isolation forest algorithm [J]. Journal on Communications, 2019, 40(8): 133-142.)

[16] 王茹雪,张丽翠,刘姝岐. 基于瀑布型混合技术的异常检测算法[J]. 吉林大学学报(信息科学版),2017,35(5):544-550. (WANG R X, ZHANG L C, LIU S Q. Anomaly detection algorithm based on waterfall hybrid technology [J]. Journal of Jilin University (Information Science Edition), 2017, 35(5): 544-550.)

[17] HARIRI S, KIND M C, BRUNNER R J. Extended isolation forest [EB/OL]. [2020-09-01]. https://arxiv.org/pdf/1811.02141.pdf.

[18] 于玲,吴铁军. 集成学习:Boosting算法综述[J]. 模式识别与人工智能,2004,17(1):52-59. (YU L, WU T J. Assemble learning: a survey of Boosting algorithms [J]. Pattern Recognition and Artificial Intelligence, 2004, 17(1): 52-59.)

[19] 李建中,刘显敏. 大数据的一个重要方面:数据可用性[J]. 计算机研究与发展,2013,50(6):1147-1162. (LI J Z, LIU X M. An important aspect of big data: data usability [J]. Journal of Computer Research and Development, 2013, 50(6): 1147-1162.)

[20] HARMAN R, LACKO V. On decompositional algorithms for uniform sampling from n-spheres and n-balls [J]. Journal of Multivariate Analysis, 2011, 101(10): 2297-2304.

[21] 李倩,韩斌,汪旭祥. 基于模糊孤立森林算法的多维数据异常检测方法[J]. 计算机与数字工程,2020,48(4):862-866. (LI Q, HAN B, WANG X X. Multidimensional data anomaly detection method based on fuzzy isolated forest algorithm [J]. Computer and Digital Engineering, 2020, 48(4): 862-866.)

[22] RAYANA S. ODDS library [DS/OL]. [2020-09-01]. http://odds.cs.stonybrook.edu.

[23] PEVNÝ T. Loda: lightweight on-line detector of anomalies [J]. Machine Learning, 2016, 102(2): 275-304.

[24] ZHAO Y, NASRULLAH Z, LI Z. PyOD: a Python toolbox for scalable outlier detection [J]. Journal of Machine Learning Research, 2019, 20: 1-7.

XIE Yu, born in 1996, M. S. candidate. His research interests include data mining, intelligent computing, anomaly detection.

JIANG Yu, born in 1980, M. S., associate professor. His research interests include intrusion detection, rough sets, data mining, intelligent computing.

LONG Chaoqi, born in 1996, M. S. candidate. His research interests include data mining, intelligent computing, wavelet clustering.