

快速近似计算 Shapley 值的归因解释方法^①



余晓晗¹, 王从波¹, 谢媛媛², 张中辉¹, 马 荣³

¹(中国人民解放军陆军工程大学 指挥控制工程学院, 南京 210007)

²(中国人民解放军 32069 部队, 北京 100091)

³(中国人民解放军 32269 部队, 兰州 730030)

通信作者: 王从波, E-mail: m17863931803@163.com

摘 要: Shapley 值归因解释方法虽然能更准确量化解释结果, 但过高的计算复杂度严重影响了该方法的实用性. 本文引入 KD 树重新整理待解释模型的预测数据, 通过在 KD 树上插入虚节点, 使之满足 TreeSHAP 算法的使用条件, 在此基础上提出了 KDSHAP 方法. 该方法解除了 TreeSHAP 算法仅能解释树结构模型的限制, 将该算法计算 Shapley 值的高效性放宽到对所有的黑盒模型的解释中, 同时保证了计算准确度. 通过实验对比分析, KDSHAP 方法的可靠性, 以及在解释高维输入模型时的适用性.

关键词: 归因解释; Shapley 值; 可解释机器学习; KD 树

引用格式: 余晓晗, 王从波, 谢媛媛, 张中辉, 马荣. 快速近似计算 Shapley 值的归因解释方法. 计算机系统应用, 2022, 31(11): 290–295. <http://www.c-s-a.org.cn/1003-3254/8751.html>

Attribution Explanation Method for Fast Approximation of Shapley Values

YU Xiao-Han¹, WANG Cong-Bo¹, XIE Yuan-Yuan², ZHANG Zhong-Hui¹, MA Rong³

¹(College of Command & Control Systems, Army Engineering University of PLA, Nanjing 210007, China)

²(32069 Unit of PLA, Beijing 100091, China)

³(32269 Unit of PLA, Lanzhou 730030, China)

Abstract: Although the attribution explanation method based on Shapley value can quantify the interpretation results more accurately, the excessive computational complexity seriously affects the practicality of this method. In this study, we introduce the k-dimensional (KD) tree to reorganize the predicted data of the model to be explained, insert virtual nodes into the KD tree so that it meets the application conditions of the TreeSHAP algorithm, and then propose the KDSHAP method. This method lifts the restriction that the TreeSHAP algorithm can only explain tree models and broadens the efficiency of the algorithm in calculating Shapley value to the explanation of all black-box models without compromising calculation accuracy. The reliability of the KDSHAP method and its applicability in interpreting high-dimensional input models are analyzed through experimental comparisons.

Key words: attribution explanation; Shapley value; interpretable machine learning; KD tree

随着机器学习的不断发展, 机器学习在各个领域得到了广泛的应用, 如医药^[1], 生物学^[2,3] 以及天文学^[4] 等. 但是, 大部分机器学习模型, 特别是神经网络模型, 为缺少透明性的黑盒模型, 即模型预测的过程无法被人们所理解^[5]. 伴随着机器学习的深入研究和广泛

应用, 人们对黑盒模型的可信度提出要求, 对于黑盒模型的可解释研究也随之发展起来.

近年来, 可解释方法研究工作得到了不断推进, 对黑盒模型进行解释的技术手段逐渐丰富起来, 比如类激活图^[6], 梯度变化^[7,8] 等可视化的方法. 但是其中一些

① 收稿时间: 2022-01-28; 修改时间: 2022-02-24; 采用时间: 2022-03-09; csa 在线出版时间: 2022-06-30

方法是定性分析而非定量分析,解释结果看似能够对模型做出解释,实际上与模型预测不一定相关^[9].为使解释结果可信度更高,研究人员将博弈论中公平分配的 Shapley 值引入到黑盒模型可解释的研究工作中^[10,11],用 Shapley 值表示每个输入特征对模型预测结果的贡献程度.虽然在理论上基于 Shapley 值的解释方法有诸多良好的性质^[12],但最大的问题在于其过高的计算时间复杂度,为指数级,并不适用于高维输入的黑盒模型,因此需要一种准确快速的近似算法去计算 Shapley 值.

针对 XGBoost、随机森林、梯度增强树等树结构模型, Lundberg 等人^[13]提出了基于 TreeSHAP 算法解释树结构模型的方法, TreeSHAP 算法能够在多项式时间复杂度内计算出特征的 Shapley 值.本文在 TreeSHAP 算法的基础上提出一种名为 KDSHAP 的可解释方法,该方法引入 KD 树重新整理待解释数据集及其模型预测结果,继承了 TreeSHAP 算法能够在多项式时间内计算 Shapley 值的优点,同时能够解释除树结构模型之外的黑盒模型.

本文的主要工作包括:通过插入虚节点改造 KD 树,对待解释模型的预测数据进行了重新整理,使得利用 TreeSHAP 算法快速计算 Shapley 值成为可能;基于 KD 树及 TreeSHAP 算法提出了 KDSHAP 方法,该方法在继承了 TreeSHAP 算法多项式时间复杂度的同时,保证了 Shapley 值解释的准确度,也避免了高维输入模型中 Shapley 值计算的限制;通过 KDSHAP 方法与其它近似算法的对比实验,显示出 KDSHAP 方法在准确性方面的优势;另外,也用实验证实了本文提出方法对高维输入模型的适用性.

1 基础知识

近些年的可解释方法的研究中,归因解释方法得到了研究人员的广泛关注,归因解释方法探究样本特征对样本预测结果的贡献程度,其中 Shapley 值是解释黑盒模型一种重要的归因方法.

1.1 归因方法

归因方法 (attribution methods) 是一种基于特征的解释方法^[14-16],通过给样本中每个特征附上一个权值的方式解释黑盒模型.假设黑盒模型 M 的输入特征有 d 维: $X = \{X_1, X_2, \dots, X_d\}$, 一条样本数据 $\mathbf{x} = (x_1, x_2, \dots, x_d)$ 输入到黑盒模型 M 的输出结果记为 $M(\mathbf{x})$. 归因方法建立映射 $\vec{\phi}(M, \mathbf{x}) = (\phi_1, \phi_2, \dots, \phi_d)^T$, 其中 $\phi_i = \phi_i(M, \mathbf{x}) \in \mathbb{R}$

($\mathbb{R}$ 是实数域) 衡量了特征 X_i 对预测结果 $M(\mathbf{x})$ 的贡献程度.若 $M(\mathbf{x})$ 与 ϕ_i 的符号相同,特征 X_i 做出有利影响,反之做出不利影响. $|\phi_i|$ 则代表了影响的大小.

Lundberg 等人^[10]认为归因方法应具有以下属性:

(1) 局部准确性 (local accuracy) 要求下列等式成立:

$$\sum_{i=1}^d \phi_i = M(\mathbf{x}) - \phi_0 \quad (1)$$

其中, ϕ_0 是所有样本共有的权重.局部准确性确保了模型 M 的输出 $M(\mathbf{x})$ 等于 ϕ_0 与样本 $\mathbf{x}$ 在各个特征下的贡献 ϕ_i 之和;

(2) 缺失性 (missingness) 要求若样本 $\mathbf{x}$ 中的 x_i 缺失,那么特征 X_i 对模型 M 预测 $\mathbf{x}$ 的贡献程度为 0, 即 $\phi_i(M, \mathbf{x}) = 0$;

(3) 一致性 (consistency) 要求对于两个不同的黑盒模型 M 和 M' , 特征 X_i 对 $M(\mathbf{x})$ 的贡献程度大于对 $M'(\mathbf{x})$ 的贡献程度,那么归因方法在解释这两个模型时应有 $\phi_i(M, \mathbf{x})$ 大于 $\phi_i(M', \mathbf{x})$.

Lundberg 等人^[10]认为联盟博弈中的 Shapley 值能够满足上述的是 3 种属性,并提出了 SHAP (Shapley value attribution explanation) 的概念,即基于 Shapley 值的归因方法.

1.2 基于 Shapley 值的特征贡献程度计算

Shapley 值是联盟博弈中用于解决可信公平分配问题的最优解,由 Shapley 于 1953 年提出^[17],其计算过程如下所示:

$$\phi_i(M, X) = \sum_{S \subseteq X \setminus \{X_i\}} \frac{|S|!(|X| - |S| - 1)!}{|X|!} \times [\mu(S \cup \{X_i\}) - \mu(S)] \quad (2)$$

式 (2) 的计算结果是特征 X_i 的 Shapley 值, X 是由特征构成的集合,即 $X = \{X_1, X_2, \dots, X_d\}$, S 是集合 $X \setminus \{X_i\}$ 的子集, $|S|$ 为集合 S 中的特征个数, μ 是度量集合 S 联盟值的函数 $\mu: S \mapsto \mathbb{R}$, 特别地 $\mu(X) = M(\mathbf{x})$. Shapley 值满足归因方法要求的局部准确性、缺失性以及一致性,保证了解释结果公平合理.

根据黑盒模型 M 求解 $\mu(S)$ 是 Shapley 值计算的关键,文献 [10] 给出了近似计算 $\mu(S)$ 的方法:

$$\mu(S) = \mathbb{E}[M(\mathbf{x})|S] \approx \mathbb{E}_{\mathcal{S}^C}[M(\mathbf{x})] \quad (3)$$

其中, $\mathcal{S}^C = X \setminus S$ 表示缺失特征. $\mu(S)$ 由条件概率期望 $\mathbb{E}[M(\mathbf{x})|S]$ 得出,但实际情况下,由于输入空间的数据分布不可知,期望 $\mathbb{E}[M(\mathbf{x})|S]$ 的计算存在难度.为近似计算

$\mu(S)$, 需要做出 S 与 S^C 相互独立的假设, 那么 $\mu(S)$ 可近似为 $\mathbb{E}_{S^C}[M(\mathbf{x})]$.

对于实际问题, 具体做法 (如图 1 所示, 图中不同颜色代表不同的值) 如下: 存在黑盒模型 M , 数据集 $D_{H \times d}$ 及其中样本 $\mathbf{x} = (x_1, x_2, \dots, x_d)$, 其中 H 表示数据集中样本个数. 欲求解 $\mu(\{x_i, x_j\})$, 需将 $D_{H \times d}$ 中第 i 列和第 j 列特征值分别重新赋值为 x_i 和 x_j , 即 $D_{:, [i, j]} = [x_i, x_j]$ 得到 $D'_{H \times d}$. 将 $D'_{H \times d}$ 送入到模型 M 得到 $M(D'_{H \times d}) = (M(D'_{1,:}), M(D'_{2,:}), \dots, M(D'_{H,:}))^T$, 至此, $\mu(\{x_i, x_j\}) \approx \frac{1}{H} \sum_{c=1}^H M(D'_{c,:})$. 本文将使用上述思想计算 Shapley 值的算法称为 RealSHAP.

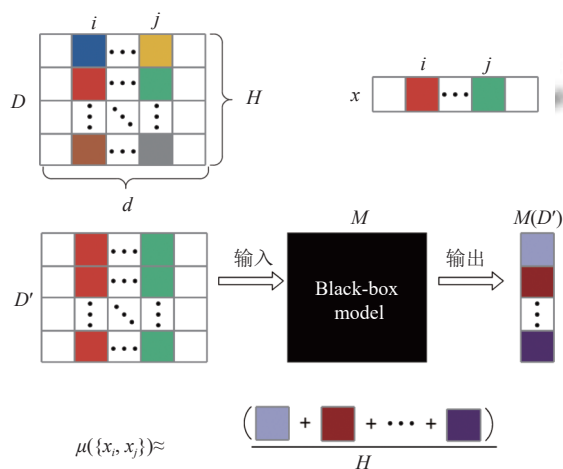


图 1 RealSHAP 算法计算联盟值

2 KDSHAP

RealSHAP 算法解释黑盒模型就必须考虑样本集 X 的 2^d 个子集 S , 该算法的时间复杂度至少为 $O(2^d)$, 在 d 比较大的高维输入特征问题中不适用. 所以需要近似算法来快速计算样本特征的 Shapley 值. 本节在 KD (K-dimensional) 树和 TreeSHAP 算法基础上, 提出一种新的快速计算样本特征 Shapley 值的方法, 称之为 KDSHAP.

2.1 KDSHAP 方法

KDSHAP 方法伪代码如算法 1 所示.

算法 1. KDSHAP

输入: 待解释样本 $\mathbf{x}$, 待解释样本集 D , 黑盒模型 M
输出: Shapley 值向量 $\phi = (\phi_0, \phi_1, \dots, \phi_d)^T$

1. Procedure $KDSHAP(\mathbf{x}, D, M)$
2. 初始化: $\phi = \mathbf{0}, T = \text{NULL}$
3. $\mathbf{y} = M(D)$
4. $T = \text{CreateKDTree}(T, D, \mathbf{y}, 0)$

5. $\phi_0 = \text{Expect}(T)$ // 计算样本公有贡献程度 ϕ_0

6. $\phi_{1:d+1} = \text{Treeshap}(\mathbf{x}, T)$ // 计算样本 $\mathbf{x}$ 特征的 Shapley 值

7. return ϕ

本文提出一种可解释方法 KDSHAP, 该方法能够快速计算 Shapley 值来解释黑盒模型. KDSHAP 方法所需的参数有待解释数据集 D , 数据集样本 $\mathbf{x}$ 以及黑盒模型 M , 其大致分为 3 个主要部分: 数据预测部分 (算法 1 中第 3 行), 该部分的作用是获得模型 M 对数据集 D 的预测结果 $M(D)$; 构建 KD 树部分 (算法 1 中第 4 行), 其目的是联合数据集 D 以及预测结果 $M(D)$ 构造出 KD 树; 模型解释部分 (算法 1 中第 5、6 行), 该部分会利用 TreeSHAP 算法以及 KD 树计算样本 $\mathbf{x}$ 中各个特征的 Shapley 值以及样本公用的贡献程度 ϕ_0 , 从而解释黑盒模型 M 对样本 $\mathbf{x}$ 的预测结果. 事实上, 算法 1 中第 2-5 行对不同样本是通用的, 因此只需一次构造 KD 树, 然后调用 TreeSHAP 算法对多个样本进行解释. 相较于 RealSHAP 算法的指数级时间复杂度, KDSHAP 能够在多项式时间内得到模型 M 对样本 $\mathbf{x}$ 预测的解释结果.

TreeSHAP 算法^[13,18]利用树结构模型的特点, 能够在多项式时间内计算出样本 $\mathbf{x}$ 中各个特征的 Shapley 值, 其时间复杂度为 $O(TLD^2)$, T 为树结构模型中树的个数, L 是模型中叶子节点的个数, D 为模型中所有树深度的最大值. 树结构模型根据树中间节点来选择左右分支, 直到叶子节点. 此外, 树结构模型在训练过程中保存了样本信息, 如中间节点上的特征阈值, 流经节点的样本数量等. 依靠树结构模型预测过程的特点, TreeSHAP 算法能够同时追踪多个特征子集, 并根据模型中保存的样本信息计算出多个特征子集的联盟值, 达到快速计算的目的. 虽然 TreeSHAP 算法在计算速度上有着巨大优势, 但是 TreeSHAP 算法仅能解释树结构模型并不适用于其它黑盒模型. 本文将 TreeSHAP 与 KD 树结合, 将其使用范围拓展到所有非树结构模型中.

2.2 KD 树引入及改造

为使 TreeSHAP 算法适用于所有的黑盒模型, 本文采用两步走的思路: 首先, 引入 KD 树将黑盒模型的输入/输出数据整理成树状结构; 然后利用 TreeSHAP 算法计算目标样本特征的 Shapley 值.

KD 树作为一种数据结构, 避免了决策树复杂的特征选择过程, 能够在短时间内快速地重新组织待解释数据集. KD 树是在 K 维欧几里得空间组织点的数据

结构,其中所有的非叶子节点都可看作是一个超平面,将当前空间划分成两个半空间,左子树上的点在超平面左边,超平面的右边的点在右子树上,该特点与决策树相似.相对于决策树,KD树在选择中间节点的特征时没有复杂的计算,能够更快速地得到数据集的树状结构.假设存在数据集 $\text{Dataset} = \{(1,7), (2,8), (3,9), (4,10), (5,11), (6,12)\}$ 采用依次选取特征并且使用选取特征的中值作为阈值,KD树结构如图2所示.

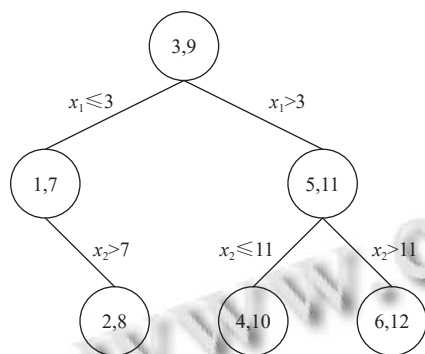


图2 KD树构造示例

虽然KD树能够重新组织数据集成树状结构,但是直接构造出的KD树不能满足TreeSHAP算法的适用条件.因为树结构模型中的关于样本预测结果的信息仅存在于叶子节点上,而KD树中的所有节点都保存有样本预测结果.为更好地利用TreeSHAP算法,需要对原始KD树结构进行改造.

相较于TreeSHAP算法所需的树结构,KD树存在的主要问题是树中所有的节点都包含样本预测结果信息.为解决适用性问题,本文通过在KD树中插入虚节点的方式改造KD树,生成过程如算法2.

算法2. CreateKDTree

输入: 待解释数据集 D , D 的模型预测值 y ,当前分支的特征索引 i
输出: KD树 T

```

1. Procedure CreateKDTree( $T, D, y, i = 0$ )
2.    $H = \text{row}(D)$ 
3.   if  $H = 1$  then
4.      $T' = \text{KDleaf}(T, D, y)$ 
       // 如果 $D$ 中仅含有一个样本,构造叶子节点.
5.   else
6.      $i' = i, i = (i+1) \% d$ 
7.     if  $H$ 是偶数 then  $h = H/2$  else  $h = (H-1)/2$ 
8.      $\widehat{D}^{(i')}, y' = \text{sort}(D, y, i')$  //按第 $i'$ 列对 $D$ 升序排列,相应的改变 $y$ 的顺序.
9.      $\tau = \widehat{D}_{h,i'}^{(i')}$ 
10.     $T' = \text{KDnode}(\tau, \text{CreateKDTree}(T, \widehat{D}_{1:h}^{(i')}, y'_{1:h}, i))$ 

```

```

11.   CreateKDTree( $T, \widehat{D}_{h+1:H}^{(i')}, y_{h+1:H}, i', h$ ) //构造中间节点.
12.   return  $T'$ 

```

改造后的KD树由两种不同的节点构成,分别是虚节点和样本节点.虚节点充当KD树的中间节点(算法2中第10,11行),其作用是选择左右分支以及保存流经该节点上的样本数量等信息.样本节点也是叶子节点(算法2中第4行),树中的每一个叶子节点都对应数据集中的一个样本,叶子节点中保存了样本数据以及模型对该样本的预测结果.经过上述算法改造,图2中的KD树结构改变为图3所示,满足了TreeSHAP算法的适用要求.

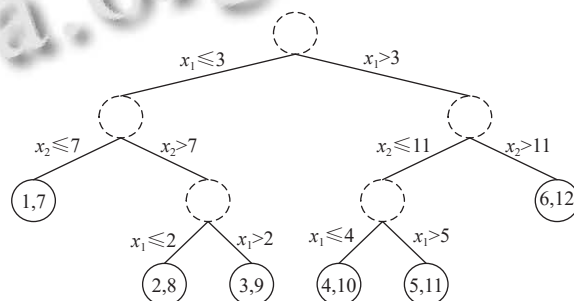


图3 经改造后的KD树

3 实验分析

本文设计了3个实验来探究KDSHAP的可行性以及准确性.实验1,比较KDSHAP方法的解释结果与其它Shapley值近似算法的准确性;实验2,研究不同样本数量构成的KD树对解释结果准确性的影响;实验3,使用KDSHAP方法解释高维输入模型,证明KDSHAP方法在高维数据上的可行性.需要说明的是,实验1与实验2都是使用RealSHAP算法的解释结果作为基准.

3.1 实验1. KDSHAP的准确性

实验1使用的数据集共有10000条数据,单个样本由3个特征组成,每个特征的数据类型都是浮点型.构造KD树时使用了整个数据集,即10000条数据,为了说明KDSHAP的准确性,对照组使用填充缺失特征的近似算法^[19].对于缺失特征 S^C ,填充缺失特征方法使用一个特殊样本填充对应缺失位置的特征值,常用的特殊样本有均值样本和零值样本等.本次对照实验使用的有均值样本,最小值样本以及最大值样本.实验1结果对比如表1.其中,本文方法的实验结果使用粗体表示,“*”表示在该指标上表现最好的方法.

表1 实验1结果对比

方法	L_1^{mean}	MSE	r_s^{mean}
均值填充	2.31	10.422 5	0.937 2
最大值填充	711.47	6.8×10^5	0.539 0
最小值填充	772.725	8.025×10^5	0.602 1
KDSHAP	0.652 5*	3.23*	0.968 8*

RealSHAP 算法的解释结果在实验中充当基准值. 假设近似算法的解释结果为 $(\hat{\phi}_1, \hat{\phi}_2, \hat{\phi}_3, \hat{\phi}_0)$, RealSHAP 算法的解释结果为 $(\phi_1, \phi_2, \phi_3, \phi_0)$, 其中 $\hat{\phi}_0$ 和 ϕ_0 为每个样本的共有权值, 分别由近似算法和 RealSHAP 算法计算得出. 该实验使用了3种指标来对比实验结果, 平均绝对值损失函数 (L_1^{mean}), 均方误差 (MSE) 以及平均斯皮尔曼秩相关系数 (r_s^{mean}). 下面所有的指标计算公式中, n 表示解释结果的个数, H 为数据集中样本数据的个数, 在本次实验中 $n=4$ 、 $H=10\,000$.

平均绝对值损失函数用来衡量近似算法与 RealSHAP 算法解释结果的平均 L_1 距离, 其计算公式如下:

$$L_1^{\text{mean}} = \frac{1}{nH} \sum_{k=1}^H \sum_{i=0}^{n-1} |\phi_{k,i} - \hat{\phi}_{k,i}| \quad (4)$$

均方误差指标是近似算法解释结果偏离 RealSHAP 算法解释结果差值的平方和的均值, 其计算过程如下所示:

$$MSE = \frac{1}{nH} \sum_{k=1}^H \sum_{i=0}^{n-1} (\phi_{k,i} - \hat{\phi}_{k,i})^2 \quad (5)$$

斯皮尔曼秩相关系数又称等级相关系数, 主要用来描述存在等级变量的两个变量间关联的程度, 其计算结果的绝对值越靠近1, 说明这两组数据的相关性越强, 绝对值越接近0, 等级相关性越弱. 本实验使用平均斯皮尔曼秩相关系数衡量近似算法与 RealSHAP 算法结果的相关程度, 首先计算出对单条样本使用的斯皮尔曼秩相关系数 (r_s), 最后求解数据集中所有样本相关系数的期望.

$$r_s = 1 - \frac{6 \times \sum_{i=0}^{n-1} (\phi_i - \hat{\phi}_i)^2}{n \times (n^2 - 1)} \quad (6)$$

$$r_s^{\text{mean}} = \frac{1}{H} \sum_{k=1}^H r_s^{(k)} \quad (7)$$

实验结果表明, KDSHAP 在3个指标上全部是最优的, 说明本文提出的方法与 RealSHAP 算法的解释结果最为接近. 此外, 还可以发现, 不同的填充样本对解释结果也有较大的影响.

3.2 实验2. 不同样本数量对 KDSHAP 准确性的影响

通过对 KD 树生成过程分析可知, 如果样本数量越多, KD 树的深度越深, 那么就可能存在同一特征在不同深度位置分裂多次的情况. 设计本次实验的目的是探究不同的样本数量对 KDSHAP 方法解释结果的影响.

本实验使用的数据集与实验1为同一数据集, 为达到实验目的, 从整个数据集中进行10次采样, 从1000个样本容量开始, 每次增加1000个样本. 根据不同容量的采样结果构建 KD 树, 计算不同样本容量下的解释结果与 RealSHAP 算法解释结果的均方误差 (MSE), 实验结果如图4所示.

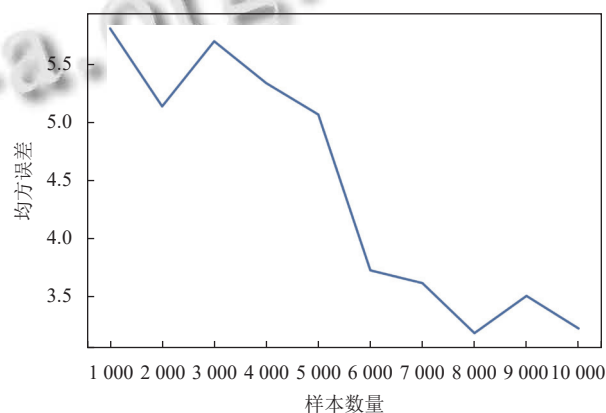


图4 均方误差随样本数量的变化情况

通过对比不同样本容量下均方误差的变化情况发现, 随着样本容量的增加, 均方误差值总体上呈现减小趋势. 这说明参与构建 KD 树的样本容量越大, KDSHAP 的解释结果误差越小越准确.

3.3 实验3. KDSHAP 对大规模数据集的适用性

该实验使用的数据集共有8000条数据, 每条数据有20维特征, 每个维度上的数据符合正态分布, 模型 M 拟合函数 $y = (\sum_{i=1}^{20} x_i)^3$, 实现回归任务. 欲对数据集中样本 $\mathbf{x} = [2.10, -0.82, 0.39, 0.76, 0.97, -0.45, -1.07, -0.84, 0.71, -2.22, 2.25, -1.44, 0.79, -0.06, 2.53, -0.61, 0.49, -0.06, -0.45, -0.28]$ 进行解释. KDSHAP 方法对该样本解释所花费的时间为4.467 8 s, 解释结果 ϕ 如表2.

表2中, 共有权重 $\phi_0 = 20.54$, 该样本的标签值为7.249, 除去计数保留法对解释结果精度的影响, KDSHAP 方法解释结果的累加和约等于样本的标签值, 即 $\sum_{i=0}^{20} \phi_i \approx 7.249$, 符合局部准确性.

而 RealSHAP 算法要考虑 $2^{|X|}$ 个子集, 其中 $|X| = 20$, 计算特征对 X 预测结果的贡献程度需要花费巨大的时间成本, 没有得到解释结果.

4 结论与展望

通过实验对比发现,本文提出的 KDSHAP 方法对黑盒模型的解释结果更接近真实 Shapley 值,而且在计算时间复杂度上相较于 RealSHAP 算法, KDSHAP 方法的计算代价更容易接受,更适用于高维输入的黑盒模型.现阶段本文只是在一些简单的表格数据及模型上做了一些尝试性工作,在未来可以应用在处理计算机视觉和自然语言处理等任务的模型上.此外, KDSHAP 还可以在 KD 树特征选择上做一些改进工作,使得 KD 树能更准确表达出待解释数据集中样本分布信息,达到更精准的效果.

表2 KDSHAP 对样本 x 的解释结果

解释维度	Shapley值	解释维度	Shapley值
ϕ_1	-1.56	ϕ_{11}	-0.21
ϕ_2	0.04	ϕ_{12}	-0.01
ϕ_3	-0.62	ϕ_{13}	-0.07
ϕ_4	-0.35	ϕ_{14}	-1.97
ϕ_5	-0.41	ϕ_{15}	-1.42
ϕ_6	0.36	ϕ_{16}	-0.22
ϕ_7	-2.55	ϕ_{17}	-0.11
ϕ_8	-2.68	ϕ_{18}	-0.25
ϕ_9	0.33	ϕ_{19}	-1.29
ϕ_{10}	0.09	ϕ_{20}	-0.98

参考文献

- Wainberg M, Merico D, DeLong A, *et al.* Deep learning in biomedicine. *Nature Biotechnology*, 2018, 36(9): 829–838. [doi: 10.1038/nbt.4233]
- Xiong HY, Alipanahi B, Lee LJ, *et al.* The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 2015, 347(6218): 1254806. [doi: 10.1126/science.1254806]
- Zhang S, Hu HL, Jiang T, *et al.* TITER: Predicting translation initiation sites by deep learning. *Bioinformatics*, 2017, 33(14): i234–i242. [doi: 10.1093/bioinformatics/btx247]
- Parks D, Prochaska JX, Dong S, *et al.* Deep learning of quasar spectra to discover and characterize damped Ly α systems. *Monthly Notices of the Royal Astronomical Society*, 2018, 476(1): 1151–1168. [doi: 10.1093/mnras/sty196]
- Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019, 1(5): 206–215. [doi: 10.1038/s42256-019-0048-x]
- Zhou BL, Khosla A, Lapedriza A, *et al.* Learning deep features for discriminative localization. *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas: IEEE, 2016. 2921–2929.
- Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. *Proceedings of the 34th International Conference on Machine Learning*. Sydney: PMLR, 2017. 3145–3153.
- Sundararajan M, Taly A, Yan QQ. Gradients of counterfactuals. *arXiv*: 1611.02639, 2016.
- Jacovi A, Goldberg Y. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020: 4198–4205.
- Lundberg SM, Lee SI. A unified approach to interpreting model predictions. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach: Curran Associates Inc., 2017. 4768–4777.
- Štrumbelj E, Kononenko I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 2014, 41(3): 647–665. [doi: 10.1007/s10115-013-0679-x]
- Shapley LS. A value for n -person games. *Contributions to the Theory of Games (AM-28)*, Volume II. In: Kuhn HE, Tucker AW, eds. Princeton: Princeton University Press, 2016. 307–318.
- Lundberg SM, Erion GG, Lee SI. Consistent individualized feature attribution for tree ensembles. *arXiv*: 1802.03888, 2018.
- Chen JB, Jordan M. LS-tree: Model interpretation when the data are linguistic. *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. New York: AAAI, 2020. 3454–3461.
- Arras L, Montavon G, Müller KR, *et al.* Explaining recurrent neural network predictions in sentiment analysis. *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen: Association for Computational Linguistics, 2017. 159–168.
- Bach S, Binder A, Montavon G, *et al.* On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One*, 2015, 10(7): e0130140. [doi: 10.1371/journal.pone.0130140]
- Shapley LS. A value for n -person games. *Contributions to the Theory of Games*, 1953: 307–317.
- Lundberg SM, Erion G, Chen H, *et al.* From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2020, 2(1): 56–67. [doi: 10.1038/s42256-019-0138-9]
- Chen H, Lundberg S, Lee SI. Understanding Shapley value explanation algorithms for trees. https://hughchen.github.io/its_blog/index.html. (2021-12-22).

(校对责编: 孙君艳)