doi:10.3772/j.issn.2095-915x.2019.04.001

# 基于机器学习的区块链智能合约脚本设计

张延华 王勇 杨兆鑫 杨睿哲

北京工业大学信息学部 北京 100124

开放科学（资源服务）标识码（OSID）

**摘要：** 区块链对上链的工业缺失数据一般不具有填充能力。本文基于机器学习框架，对前端数据经规范化处理并经机器学习模型训练，获得训练模型参数并被写入智能合约脚本，编译后被部署到区块链中。缺失数据通过调用智能合约进行缺失值的拟合填充并上传区块链。

**关键词：** 区块链；智能合约；数据拟合；机器学习
**中图分类号：** N99 G35

# Design of Block Chain Smart Contract Script Based on Machine Learning

ZHANG Yanhua WANG Yong YANG Zhaoxin YANG Ruizhe

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

**Abstract:** Block chains usually have no capacity of filling the industrial missing data in the upstream chains. Based on machine learning framework, the front-end data is normalized and trained by machine learning model. The parameters of training model are obtained and written into smart contract script. After compiling, they are deployed in block chain. Missing data are fitted and filled by calling intelligent contracts and uploaded to block chains.

**Keywords:** Block chain; smart contracts; data fitting; machine learning

**基金项目：** 国家自然科学基金资助项目（61571021，61901011）；北京市博士后工作经费资助项目 (2018ZZ029，ZZ2019-73)；中国博士后科学基金第 64 批面上项目 (2018M640032)；北京市朝阳区博士后科研经费资助 (2018ZZ017)；北京工业大学基础研究基金。
**作者简介：** 张延华（1960-），教授，研究方向：工业互联网、区块链及智慧无线通信网络研究；王勇（1995-），硕士研究生，研究方向：区块链技术与数据挖掘技术；杨兆鑫（1993-），博士研究生，研究方向：区块链技术与数据挖掘技术；杨睿哲（1982-），讲师，研究方向：无线通信技术与移动边缘计算技术等方面的研究，E-mail: yangruizhe@bjut.edu.cn。

## 引言

工业数据的完整性决定其应用场景的有效性。目前，各类自动数据采集设备及数据传输环节易于受到外界干扰，容易出现数据丢失现象。对缺失数据如果不能及时发现和处理，将对数据集的完整性和有效性产生严重影响。机器学习算法是拟合工业数据缺失值的一种有效方法，对前端数据经规范化处理，利用机器学习模型训练并获得训练模型参数，则数据缺失值可通过拟合进行预测填充。在工业场景下运用区块链技术，鉴于区块链本身的不可篡改特性，可以保证链上数据的完整性和有效性[1]。

区块链智能合约包括事务处理、保存机制以及一个完备的状态机[2]。事务及事件信息载入智能合约后，其资源状态将被更新并进而触发智能合约进行状态机判断。如果自动状态机中相应动作的触发条件满足，则由状态机根据预设信息选择合约动作自动执行[3]。基于智能合约对事物的处理机制，文献 [4] 设计了一种基于区块链的众筹智能合约，将众筹规则写入智能合约并实现了一个区块链众筹合约系统；文献 [5] 提出一种基于区块链的物联网可伸缩管理机制，在智能合约中写入接口使得用户可通过调用智能合约对物联网设备进行伸缩管理。本文基于机器学习框架，提出针对前端数据采用机器学习模型训练数据，将训练完成的模型存储到本地，经训练获得的训练模型参数写入智能合约脚本，利用规则对上链数据进行缺失数据的拟合预测，脚本编译后被部署到区块链并通过调用智能合约进行缺失值的拟合填充和上传区块链。

## 1 区块链智能合约脚本设计

区块链智能合约基于对事物的处理机制，合约脚本采用机器学习算法经训练获得的训练模型，利用规则对上链数据进行拟合预测。

### 1.1 功能架构

系统功能架构如图 1 所示，由物理层，机器学习层与区块链层组成。物理层负责数据采集，包括移动终端、各类工业传感器设备，数据上传到机器学习层。前端数据在机器学习层中对数据进行规范化处理，将其划分为训练数据集与测试数据集，它们分别用于机器学习模型的训练与性能的评估。经训练获得的训练模型参数被写入智能合约，编译后被部署到区块链中。测试数据即实际工作中需存入区块链中的数据通过智能合约置入的规则（拟合参数），进行模型预测和数据拟合，补全缺失数据。

### 1.2 机器学习回归算法脚本

#### 1.2.1 脚本模型

设因变量 $y$ 与 $p$ 个自变量 $x_1$，$x_2$，…，$x_p$ 相关，现采集有 $n$ 组数据（独立观测数据）

$$(x_{i1},x_{i2},...,x_{ip},y_i),\quad i=1,2,..,n$$

则多元线性回归模型为公式 (1)：

$$\begin{cases} y_i=\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\cdots++\beta_px_{ip}+\varepsilon_i,\quad i=1,2,\cdots,n \\ E(\varepsilon_i)=0 \\ Var(\varepsilon_i)=\sigma^2 \\ \mathrm{cov}(\varepsilon_i,\varepsilon_j)=0,\quad i\neq j,\quad i,j=1,2,\cdots,n \end{cases} \tag{1}$$

式中 $\beta_k,k=0,1,\cdots,p$ 是回归系数，$\varepsilon_i,i=0,1,\cdots,n$ 是误差项[6]。上式的矩阵描述为公式 (2)：

$$\begin{cases} Y=X\beta+\varepsilon \\ E(\varepsilon)=0 \\ Var(\varepsilon)=\sigma^2I_n \end{cases} \tag{2}$$

doi:10.3772/j.issn.2095-915x.2019.04.001

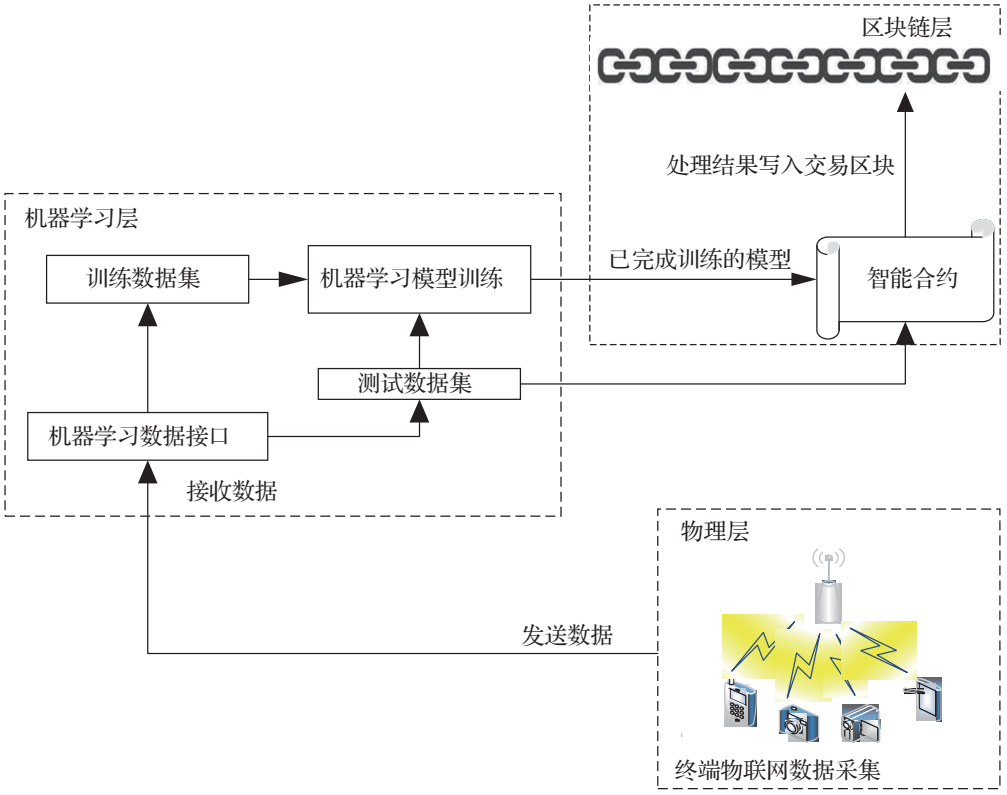


图 1 系统功能架构

### 1.2.2 参数估计

若记 $Q(\beta)=(Y-X\beta)'(Y-X\beta)$，则 β 的最小二乘估计 $\hat{\beta}$ 使 $Q(\beta)$ 达到最小。这里 $\hat{\beta}$ 是正规方程组 (3) 的解 [7]。

$$X'X\beta=X'Y \tag{3}$$

当 $X'X$ 的逆存在时，它的最小二乘估计为公式 (4)：

$$\hat{\beta}=(X'X)^{-1}X'Y \tag{4}$$

### 1.2.3 缺失值拟合

如果回归参数 $\beta$ 的估计为 $\hat{\beta}$，则估计的多元线性回归模型为公式 (5)，$\hat{Y}$的值就是拟合值 [8]。

$$\hat{Y}=X\hat{\beta} \tag{5}$$

当 n 组观测数据中发生数据缺失时，若用一组预测值填补缺失值，对于给定的 $x=x_0=(x_{01},x_{02},...,x_{0p},y_i)'$，$y_0$ 的无偏估计是公式 (6)：

$$\hat{y_0}=\hat{\beta}_0+\hat{\beta}_1x_{01}+\hat{\beta}_2x_{02}+\cdots+\hat{\beta}_px_{0p} \tag{6}$$

它的 100(1–α)% 置信区间为公式 (7)：

$$x_0'\hat{\beta}\pm t_{n-p-1}\left(\frac{\alpha}{2}\right)\sqrt{(x_0'(X'X)^{-1}x_0)s^2} \tag{7}$$

其中 $t_{n-p-1}(\alpha/2)$ 为自由度 $n$-$p$-1 的 $t$ 分布上 100(α/2) 百分位数 [9]。

在机器学习中实现上述拟合过程的脚本文件的伪代码如下：

Data 为完整标准化数据，$\hat{\beta}$ 为最佳估计

输入：完整的训练数据集

输出：线性回归最佳估计

1. Input Data
2. 划分为数据 xArr 和目标值 yArr
3. X ← xArr，Y ← yArr
4. 计算 $X^TX$=X.T*X

5. IF $X^TX == 0$
6. 后台打印“数据矩阵不可逆，无法求解”
7. Else $\hat{\beta}$ = $X^TX$ 的逆矩阵乘以 X 的转置乘以 Y
8. End

### 1.3 智能合约脚本

智能合约通常具有值和状态两个属性。合约触发条件被满足，则状态机根据预设信息选择合约动作自动执行。合约脚本模型经训练获得一个拟合函数，即训练模型，利用规则对上链数据的缺失值进行拟合及补全[10]。

智能合约调用流程如图 2 所示，首先，任意节点如需访问区块链，将含缺失值的数据 X 输入机器学习器，合约脚本被执行得到参数估计值 $\hat{\beta}$，$\hat{\beta}$ 值被实时刷新并被编译进智能合约；其次将 X 与 $\hat{Y}$ 作为完整数据存入区块链，触发编译部署的智能合约，通过置入的规则（拟合参数），在数据接收端进行模型预测和数据拟合，得到响应 $\hat{Y}$，$\hat{Y}$ 中就包含了缺失值的拟合填充值。

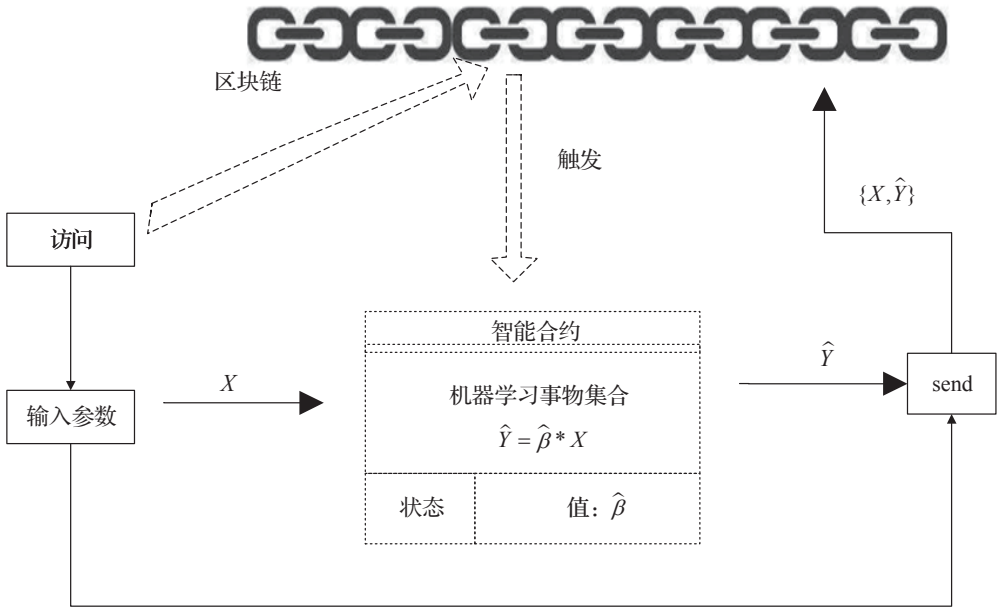


图 2 智能合约调用流程

数据预处理合约算法伪代码如下：

输入：待处理数据

输出：如果处理成功，完整数据存入区块链；如果处理失败，返回错误信息

1. 编译线性回归模型智能合约
2. 部署智能合约并调用 start（）；
3. 解锁已有账户
4. 获取输入数据
5. 与最佳估计脚本交互
6. If 交互出错
7. Return error
8. Else 获得最佳估计 $\hat{\beta}$
9. If $\hat{\beta}$ 与智能合约脚本不符
10. Return error
11. Else 异步执行：实例化智能合约
12. Then putParams（最佳估计系数，{gas:

doi:10.3772/j.issn.2095-915x.2019.04.001

140000, from: accountAddress} )

13. For：数据逐行输入

14. If：完整数据，直接存入区块链

15. else: y = regression( 输入单行数据 )

16. 将输入单行数据与 $\hat{Y}$ 打包成完整数据存入区块链

17. End

本文使用线性回归模型，将机器学习与区块链智能合约相结合完成数据的预处理。公式 (5) 中 X 与 $\hat{Y}$ 共同组成了完整的每一组数据，在数据收集的过程中缺失了 $\hat{Y}$ 数据，根据线性回归的特性确定两种或两种以上变量间相互依赖的定量关系，通过数据训练得到的最佳估计 $\hat{\beta}$ 的转置与剩余数据 X 矩阵相乘，便可以补全缺失值 $\hat{Y}$。本文训练模型的部分在机器学习节点中完成，将实际情况中涉及的公式 $\hat{Y} = \hat{\beta} * X$ 通过 Solidity 语言写入智能合约脚本，其中 $\hat{\beta}$ 为训练出的最佳估计，X 是自变量，$\hat{Y}$ 是预测出的因变量。

应用时通过预制触发条件调用指定的智能合约，输入自变量 X 矩阵，预制响应结果为根据 X 矩阵与最佳估计 $\hat{\beta}$ 矩阵的转置相乘得到预测结果 $\hat{Y}$，最后将 X 与 $\hat{Y}$ 同时存入区块链，存入区块链的数据一经上传便不可篡改，完成基于机器学习的智能合约的部署与调用。

## 2 功能设计

以太坊（Ethereum）是一个基于区块链数据结构的、可实现智能合约的、开源的底层系统。在以太坊体系中，一个合约就是一个存在区块链里的程序。以太坊虚拟机（Ethereum Virtual Machine，EVM）提供了一种图灵完备的脚本语言（Ethereum Virtual Machine Code），可以执行编写的以太坊合约。本文基于以太坊区块链实验环境，实现了基于机器学习的区块链智能合约脚本。

合约脚本的实现分为四个部分：回归算法；合约编译；合约部署与合约调用。

（一）回归算法。数据预处理算法模型是基于线性回归算法设计的。通过 Python 语言编写，利用平方误差最小解出最佳估计值。

（二）合约编译。智能合约脚本是基于 Solidity 语言编写的，以 EVM 为运行环境。Solidity 被设计成了以编译的方式生成以太坊虚拟机代码如下所示：

```
contract Trace {

    uint attr_number = 1;


    struct ModelParams {
            uint one;
            uint two;
            uint three;
            uint four;
            uint five;
        }

    mapping (uint  => ModelParams) params;
    //存储模型参数
    function putParams(uint _one, uint _two, uint _three, uint _four, uint _five  ) public{
                ModelParams memory item = ModelParams(_one, _two,_three,_four,_five);
                params[attr_number] = item;
    }
    //计算回归模型预测结果
    function regression(uint _attr_number, uint _year, uint _numPieces, uint _newOrUsed, uint _original) returns(uint) {
        ModelParams memory item = params[_attr_number];
        return item.one - item.two*_year + item.three*_numPieces + item.four*_newOrUsed + item.five*_original;
    }
}
```

truffle 编译结果如下所示：


```
{
  "contract_name": "Trace",
  "abi": [
    {
      "constant": false,
      "inputs": [
        {
          "name": "_attr_number",
          "type": "uint256"
        },
        {
          "name": "_year",
          "type": "uint256"
        },
        {
          "name": "_numPieces",
          "type": "uint256"
        },
        {
          "name": "_newOrUsed",
          "type": "uint256"
        },
        {
          "name": "_original",
          "type": "uint256"
        }
      ],
      "name": "putParams",
      "outputs": [],
      "payable": false,
      "type": "function"
    }
  ],
  "unlinked_binary": "0x6060604052600160005534156100145760008fd5b5b6102018061002460000396000f30
0606060405263ffffffff7c0100000000000000000000000000000000000000000000000000000000600035041663
6d474f6d81146100485780638063cda227101461007c575b600080fd5b341561005357600080fd5b61006a60043560243
5604435606435608435610124565b005b60006100aa6101a5565b60008781526001602052604090819
0209060a090519081016040908152825482526001830154602083015260028301549082015260038201546060820
5260049091015460808201908152909150839051028482606001510286836040015102888460200151028451030
10191505b5095945050505050565b61012c6101a5565b60a060405190810160405280878152602001868152602001
858152602001848152602001838152509050806001600080548152602001908152602001600020600082015181556
0208201518160010155604082015181602015560608201518160030155608082015160049091015550505050
5050565b60a060405190810160405280600081526020016000815260200160008152602001600081526020016000
15250905600a165627a7a72305820a04a21d8bc138937fe0a8415a7bf038e42bf86798a4f58666cf8daf21b02ef2c
0029",
  "networks": {
    "12345": {
      "events": {},
      "links": {},
      "address": "0xba1c1e4ab0ec7423f1175ac2a7faf45fcf5f1598",
      "updated_at": 1560906994748
    }
  },
  "schema_version": "0.0.5",
  "updated_at": 1560906994748
}
```


（三）合约部署。经过 truffle 编译可验证智能合约语法正确性，并将其部署到区块链上。创建一个以太坊账户用于挖矿，在部署智能合约时对应的区块链网络必须要有正在挖矿的节点进行验证。如 3 图所示，以太坊收到部署请求并挖矿认证。本文在以太坊私有链上部署了基于机器学习的智能合约脚本用于对上链数据进行预处理。

（四）合约调用。合约的调用主要用于验证其功能是否完善，能否利用机器学习算法对数据进行拟合处理并上传存入区块链。本文采用乐高模型价格预测数据对智能合约脚本进行验证，数据如图 4 所示。

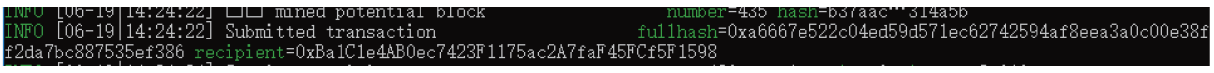


图 3 挖矿认证过程

doi:10.3772/j.issn.2095-915x.2019.04.001

```
[1.0000e+00 2.0060e+03 8.0000e+02 0.0000e+00 4.9990e+01]
[1.0000e+00 2.0060e+03 8.0000e+02 0.0000e+00 4.9990e+01]
[1.0000e+00 2.0060e+03 8.0000e+02 0.0000e+00 4.9990e+01]
[1.0000e+00 2.0020e+03 3.0960e+03 0.0000e+00 2.6999e+02]
[1.0000e+00 2.0020e+03 3.0960e+03 0.0000e+00 2.6999e+02]
[1.0000e+00 2.0020e+03 3.0960e+03 0.0000e+00 2.6999e+02]
[1.0000e+00 2.0020e+03 3.0960e+03 0.0000e+00 2.6999e+02]
[1.0000e+00 2.0020e+03 3.0960e+03 0.0000e+00 2.6999e+02]
[1.0000e+00 2.0020e+03 3.0960e+03 1.0000e+00 2.6999e+02]
[1.0000e+00 2.0020e+03 3.0960e+03 0.0000e+00 2.6999e+02]
```

图 4 乐高模型价格预测数据

根据线性回归算法求出的 $\hat{\beta}$ 为：

```
{"one": 86281.65653636074, "two": 42.98888487839099, "three": 0.00890952361462638, "four": 120.50201628989635, "five": 1.455913306084497}
```

由上述智能合约的调用结果可以看出，本文设计的智能合约脚本很好的实现了预期设计目标，并且验证了基于机器学习的区块链智能合约的有效性与可行性。本文提出的基于机器学习的智能合约脚本有部分体现在利用区块链技术和机器学习技术协同来完成在节点上的数据预处理。能够很好的将机器学习与区块链相结合，将数据存储、数据处理放在区块链中完成，利用区块链的不可篡改特点很好的提高了数据存储准确安全性以及数据处理的安全性。在区块链的智能合约中加入机器学习算法，利用机器学习算法提高区块链的数据拟合能力，将入链数据进行拟合处理。实现一个基于智能合约的工业数据预拟合的区块链应用。

## 3 结束语

本文基于机器学习算法，智能合约理论和以太坊测试环境实现了一个工业数据拟合的区块链应用，验证了应用的正确性与和可实施性。比特币的出现带动了区块链的发展，至今为止区块链的应用重心已经从数字货币向智能合约转移，智能合约是区块链未来的研究方向。智能合约技术在区块链的应用中也存在挑战，一旦部署便无法更改意味着合约要有着高容错率，意外情况需合理响应。下一步的研究工作将会完善应用功能，提高智能合约的强壮性与可用性。

## 参考文献


[1] Fedchenkov P. Reinventing Energy Consumption with Blockchain[J]. Transmission & Distribution World, 2019.

[2] 张帅 , 延安 , 贾敏智 . 基于区块链的众筹智能合约设计 [J]. 计算机工程与应用 , 2019, 55(8):220-225.

[3] 徐晓冰 , 戚枭宏 , 王建平 , 等 . 基于区块链的物联网可伸缩管理机制 [J/OL]. 计算机应用研究 , 2019(7):1-5. https://doi.org/10.19734/j.issn.1001-3695.2019.01.0022.

[4] 张彬 , 广晖 , 陈熹 . 一种基于智能合约的无线 Mesh 网络安全架构 [J/OL]. 计算机工程 , 2019(7):1-10. https://doi.org/10.19678/j.issn.1000-3428.0054940.

[5] 李松钊 , 李文敬 , 陆建波 . 物流服务交易区块链与蚁群智能合约算法研究 [J/OL]. 计算机工程与应用 , 2019(5):1-10.

[6] Zhang L F, Wang Y L, Li F Y, et al. A game-theoretic method based on Q-learning to invalidate criminal smart contracts[J]. Information Sciences, 2019, 498.

[7] Özdemir S, Olcay Arslan O. An alternative algorithm of the empirical likelihood estimation for the parameter of a linear regression model[J]. Communications in Statistics - Simulation and Computation, 2019, 48(7):1913-1921.

[8] Ciullan G, D'Amico A. Building energy performance forecasting: A multiple linear regression approach[J]. Applied Energy, 2019(253):113500.

[9] Acitas S, Senoglu B. Robust change point estimation in two-phase linear regression models: An application to metabolic pathway data[J]. Journal of Computational and Applied Mathematics, 2020, 363.

[10] 马春光 , 安婧 , 毕伟 , 等 . 区块链中的智能合约 [J]. 信息网络安全 , 2018(11):8-17.