Computer Science and Application 计算机科学与应用, 2020, 10(4), 710-720
Published Online April 2020 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2020.104074

Hans汉斯

TMS Data Error Correction Method Based Spark Distributed Support Vector Machine

Song Wang[1], Xueguang Zhou[2], Rui Chen[1]

[1]School of Economics and Management, Chuxiong Normal University, Chuxiong Yunnan
[2]Department of Information Security, Naval University of Engineering, Wuhan Hubei
Email: 36606469@qq.com, zxg196610@hotmail.com, chenrui@cxtc.edu.cn

Received: Apr. 1st, 2020; accepted: Apr. 16th, 2020; published: Apr. 23rd, 2020

Abstract

Massive date generated from TMS needs to be analyzed, so as to address the in-consistency between the financial data and real data, wrong data input, and data missing. This paper proposes a method to identify and correct abnormal data in data site maintenance times of TMS, which is based on support vector machine training algorithm running on the Hadoop distributed cluster-based framework and Spark distributed parallel computing platform. To this end, the writer takes a series of data represented by site type as the feature attribute and uses models which support vector machine algorithm to predicate and evaluate each site, thus identifying the abnormal sites needed to be further checked by relevant personnel. This method has been finally verified by experiment.

Keywords

Support Vector Machine, Transportation Management System, Spark-MLlib SVM, Data Error Correction

基于Spark分布式支持向量机的TMS数据纠错方法研究

王　松[1]，周学广[2]，陈　瑞[1]

[1]楚雄师范学院经济与管理学院，云南 楚雄
[2]海军工程大学信息安全系，湖北 武汉
Email: 36606469@qq.com, zxg196610@hotmail.com, chenrui@cxtc.edu.cn

收稿日期：2020年4月1日；录用日期：2020年4月16日；发布日期：2020年4月23日

文章引用: 王松, 周学广, 陈瑞. 基于Spark分布式支持向量机的TMS数据纠错方法研究[J]. 计算机科学与应用, 2020, 10(4): 710-720. DOI: 10.12677/csa.2020.104074

摘 要

在智能电网通信管理系统(TMS)中产生的大量数据信息有待分析总结，这些数据信息存在账务和实物不一致、数据录入错误以及缺失数据等问题。本文基于Hadoop分布式集群基础框架和Spark通用并行计算平台的分布式支持向量机训练算法，提出一种针对TMS系统数据站点检修次数中的异常数据纠察分析的解决方法。该方法以站点类型为代表的一系列数据为特征属性，使用支持向量机算法建立的模型，对各个站点进行预测和评级，纠察出异常站点，以供相关人员进行排查。最后该方法通过实验进行了验证。

关键词

支持向量机，TMS，Spark-MLlib SVM，数据纠错

1. 引言

在智能电网通信管理系统(TMS)中，对于录入的业务数据的分析是至关重要的。在当今数据量暴增的时代[1]，录入的数据不仅有着规模大的问题，还有包括普遍存在的账务和实物不一致，数据录入错误以及存在缺失数据等关键的应用问题，所以纠错作为其中必不可少的业务需求同样是一个关键[2]。传统的人工纠错在面对大规模数据集时，高开销低效率问题突出，这时结合大数据挖掘技术的算法纠错就显得格外重要。通过支持向量机等算法在文本分类上的成功应用，可以轻松取缔传统的纠错方式，实现自动发现账务与实物的不一致，发现已有数据的异常，对录入的错误数据分类划分出来，并做出预测推荐。

支持向量机作为当前广为人知的经典机器学习算法[3]，在统计分类和回归分析等领域被普遍地应用，在数据量较少，特征维度较高的分类问题中凭借其结构化风险最小的出色的泛化能力，而成为目前最为常用的分类器之一。但它也存在明显的缺陷，就是当面对大规模数据量的存储分析时显得力不从心，训练样本不断变大时，支持向量机算法训练的内存和时间消耗急剧增加，所以这种传统的模式在当今数据量大爆发的时代已经无法适应[4]。如今应对大规模数据集的常用方法是借助分布式云平台实现并行计算，比如 Hadoop 分布式集群基础框架和 Spark 通用并行计算平台[5]。为了解决支持向量机算法在大规模数据集中的应用效率低、开销大的常见问题，优化算法性能，将支持向量机和 Hadoop、Spark 结合实现分布式支持向量机有着必然的需求和广阔的前景[6]。通过并行的分布式支持向量机可以大幅降低串行支持向量机的内存开销和训练时间，具备现实实践需求和价值。

本文将基于 Hadoop 分布式集群基础框架和 Spark 通用并行计算平台上实现的分布式支持向量机算法应用到 TMS 系统数据纠错中，为人工纠察数据异常难，开销大的问题提供一种 AI 辅助解决方法。

2. 分布式支持向量机

2.1. 支持向量机

支持向量机(Support Vector Machine, SVM)是 Cortes 和 Vapnik 在 1995 年正式发表的一种基于统计学习的机器学习算法，凭借其在文本分类任务中的出色性能成为了机器学习的主流技术。如今经过多年研

究发展，支持向量机思想已经涉及分类、回归等多个领域，实现了线性、非线性以及分布式等多种支持向量机算法应用。

在给定的训练样本集后，简单二分类学习的最基本思想就是基于给定的训练样本空间中找到一个超平面，可以将不同类别的样本划分开来。其中距离超平面最近的几个训练样本点即为“支持向量”(support vector)，两个异类支持向量到超平面的距离之和即为“间隔”，具有最大间隔的超平面划分就是鲁棒性最高，泛化能力最强的最优划分。由此可推导出支持向量机的基本型及其本质就是如何高效求解一个凸二次规划问题，如公式(1)所示，其中公式中出现的 ω 为超平面的法向量，b 为超平面的位移项，(x_i, y_i) 为样本点。

$$\begin{aligned} &\min_{\omega,b} \frac{1}{2}\|\omega\|^2 \\ &\text{s.t. } y_i\left(\omega^{\mathrm{T}} x_i + b\right) \geq 1, i = 1, 2, \cdots, m \end{aligned} \tag{1}$$

对于求解一个凸二次规划问题有两种常用的方法，可以直接利用成熟的优化计算包求解，也可以引入拉格朗日乘子法，添加拉格朗日乘子 $\alpha_i \geq 0$，得到比原问题更容易求解的对偶问题，如公式(2)所示：

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^{\mathrm{T}} x_j \\ &\text{s.t. } \sum_{i=1}^{m} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, 2, \cdots, m \end{aligned} \tag{2}$$

最后通过 SMO 等方法高效地求解二次规划问题，得到判别模型，如公式(3)所示：

$$f(x) = \omega^{\mathrm{T}} x + b = \sum_{i=1}^{m} \alpha_i y_i x_i^{\mathrm{T}} x + b \tag{3}$$

在求解过程中我们不难发现所谓支持向量机，即是最终模型仅与支持向量有关，此种分类器的关键是如何找到合适的支持向量构建解模型，其复杂度也主要和支持向量的数目有关，这样可以帮助我们抓住关键样本，提出冗余属性，一定程度上避免了维数暴增难题，具有更好的鲁棒性。

在现实任务中，原始样本并非都是理想的线性可分形式，若在原始样本空间中，不存在一个能正确划分两类样本的超平面时，则需要引入核函数，借助核技巧，将样本从原始空间映射到一个更高维的特征空间，使得样本在这个新的特征空间内线性可分。同时在现实任务中，完全的线性可分太过理想，我们需要允许支持向量机在一些样本上有一定的容错率，可以出错，由此引入了“软间隔”的思想，加入了损失函数，即为常用的“软间隔支持向量机”。

支持向量机的最基本思想就是寻找到一个合适的超平面分割不同类别的数据，显然这是一个典型的解决二分类问题的思路。但在现实任务应用问题中，仅仅解决二分类问题是不够的，所以需要构建出适合的多类分类器。目前常用的解决方案主要从修改优化求解问题或者通过多个二分类器合并的思路实现了以下几种：

(1) 修改目标函数，把多个分类超平面的求解问题融合到一个最优化求解问题中。

(2) 针对 K 类问题，训练 K 个二分类器，每个分类器将把某一类归为一类，其它不属于该类的样本划为另一类，最终分类结果考虑 K 个分类器的分类函数值，选择其中最大的那一类为分类结果。

(3) 针对 K 类问题，选取任意两个类别的样本，从中设计一个二分类器，由此可得到 $k(k-1)/2$ 个二分类器。每当分类未知样本时，考虑所有分类器的结果，选择其中出现最多的一个类别作为最终的分类结果。由台湾大学林智仁教授等开发设计的经典支持向量机算法包 LibSVM [7]中的多分类解决方案就是依据这种方式实现的。

(4) 针对 K 类问题，参考决策数的多分类思想，先把所有类别划分成两个子类，由此训练得到一个

二分类器作为一个分枝节点，而后分别把两个子类的每一类内部再进一步划分为两个子类，再训练，如此迭代循环，直到最后得到原始 K 类问题中的一个独立类别为止，该类即为最终分类结果。

2.2. Spark-MLlib SVM 的分布式实现分析

MLlib 是 Spark 为了用户更加便捷地使用机器学习算法解决问题，而提供的机器学习拓展库(machine learning library)。用户可以利用拓展库中封装的部分经典算法，同时可以依据自己的需求做出相关更改，具有较强的兼容性和可拓展性，大大简化了机器学习在分布式环境下的移植工作。

对于 Spark2.2.0 系统中 Spark-MLlib 中封装的支持向量机算法，主要实现了随机梯度下降的线性二分类的支持向量机[8]，遵从线性二分类的最基本思路，寻找最优化划分超平面，即寻找最优的支持向量。目标函数为：$y=\omega^{\mathrm{T}}x+b$，ω 为超平面法向量，b 为超平面的位移项。考虑实际问题难以做到线性可分，需要一定的容错率，因此采用软间隔方式，引入损失函数，采用的是 hinge 损失函数：$\ell_{\text{hinge}}(z)=\max(0,1-z)$。在引入损失函数后可以添加正则化项，利用正则化方法防止模型过拟合，提高泛化能力。显然可以引入松弛变量 ξ_i，依此把待优化的二次规划问题重写为公式(4)：

$$\begin{aligned}&\min_{\omega,b,\xi}\frac{1}{2}\|\omega\|^2+C\sum_{i=1}^{m}\xi_i\\&\text{s.t. } y_i\left(\omega^{\mathrm{T}}x_i+b\right)\geq 1-\xi_i,\ \xi_i\geq 0,\ i=1,2,\cdots,m\end{aligned}\tag{4}$$

同样可以引入拉格朗日乘子，将上面的公式转化为对偶问题再求解。在求解过程中，我们需要求解合适的参数 ω，使得损失函数取值最小化，即要求在最大化间隔的同时，不满足约束的错误样本应该尽可能的少，这也被称为最优化的过程。最优化问题中最常采用的方法就是随机梯度下降优化方式。Spark-MLlib 也采用这种方式实现了随机梯度下降的线性支持向量机，即继承了 Generalized Linear Algorithm 的 SVMWithSGD 类，该类中定义了训练支持向量机分类模型的 train()方法，是创建线性支持向量机模型的入口。Train()方法将会创建 SVMWithSGD 对象，并使用 run()方法调用 optimitor：Gradient Descent (gradient, updater)优化得到模型的权值参数 weights，并调用 create Model (weights)方法创建一个 SVMModel，最后把 SVMModel 作为返回结果输出。

其中 GradientDescent 即为随机梯度下降算法封装的求解器，通过其下的成员方法 runMiniBatchSGD 实现模型参数的迭代运算，在运算过程中也通过集群下模型参数的共享同步和先求解分区上的梯度值再聚合求总梯度值的两方面计算方式体现分布式运算的思想。SVMWithSGD 的类图关系如图 1 所示。

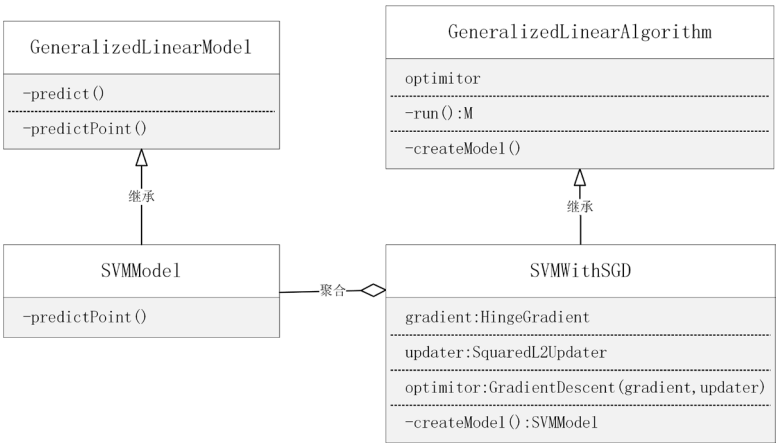

Figure 1. Class diagram of SVMWithSGD
图 1. SVMWithSGD 类图

仅从 Spark-MLlib 的开源代码中看，程序执行的都是基于弹性分布式数据集(Resilient Distributed Datasets, RDD)的操作。这是因为 Spark-MLlib 使用 RDD 抽象了所需要的分布式计算，将其转化为 RDD 上的转换(Transform)和动作(Action)等操作。RDD DAG 是用来描述 RDD 之间的依赖关系，Spark 依照此依赖关系利用 DAG 划分成不同的 Stage，Stage 间依据依赖关系先后执行。但每个 Stage 都会将对应的 Task 任务提交给 Executor，如此在一个 Stage 进行运算时 Executor 便可以并行计算 Task 任务。以 Spark 中 wordCount 为例，其在集群上分布式计算的流程示意图如图 2 和图 3 所示。

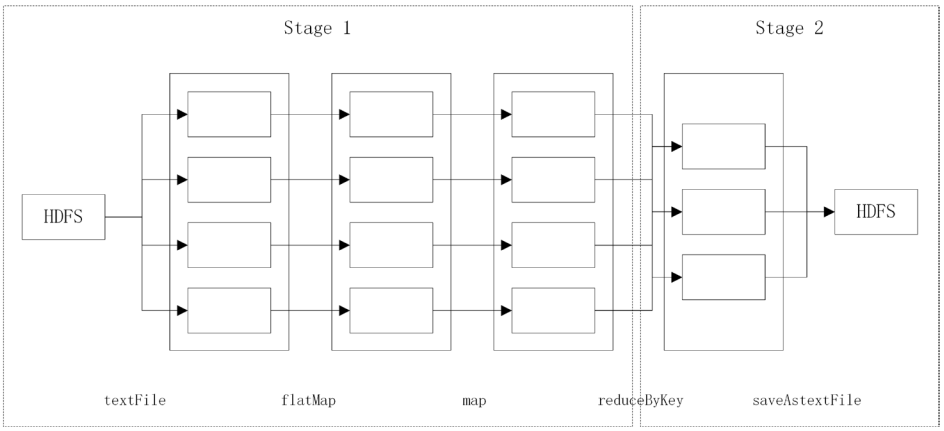

Figure 2. Distributed computing process of wordCount
图 2. wordCount 分布式计算流程

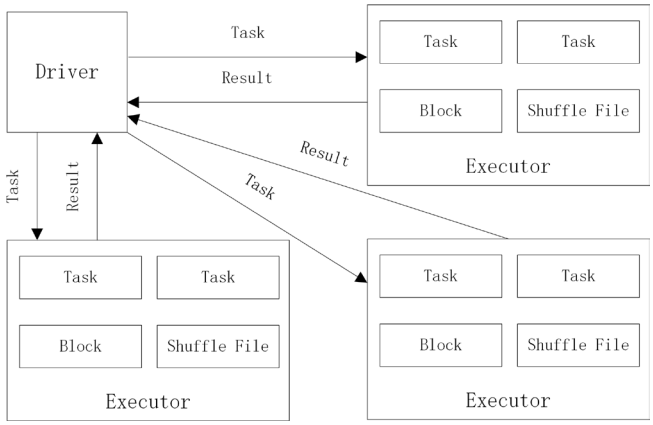

Figure 3. Distributed computing of Spark
图 3. Spark 分布式计算

使用 Spark-MLlib 中的分布式线性支持向量机和台湾大学林智仁教授开发的 LibSVM 工具包实现的单机支持向量机算法做简单的对比实验可以发现，分布式支持向量机在数据量较少时并不能体现出相比于单机的明显优势，但当数据集规模巨大超过十万数量级时，分布式支持向量机算法将在时间开销上大大优于单机支持向量机算法。正如文献[9]中记录的实验，实验数据采用来自 UCI/Covtype 的数据集 covtype，数据行数达到 581,012 条，特征数共 54 个 [1]。分别使用 LibSVM 实现的单机线性支持向量机分类算法和 Spark-MLlib 实现的分布式线性支持向量机分类算法对该数据集进行训练分类测试，最终实验结构表明：

[1] http://archive.ics.uci.edu/ml/datasets/Covertype.

在面对数据行数达到万条记录，特征数达到百位计数的同一份数据集，训练迭代次数都选用 20 次。此时分布式支持向量机算法和单机下 LibSVM 分类的准确率相差不大，但相比在时间效率上，分布式支持向量机算法有着明显的优势，如表 1 所示。

Table 1. Experimental comparison results
表 1. 实验对比结果

	covtype	
	准确率	执行时间(s)
Spark-MLlib	61.18%	96
LibSVM	77.12%	54320

3. 基于 Spark 分布式支持向量机的 TMS 数据纠错实现及实验分析

本文实验数据来自于依托某项目获得的电网 TMS 系统数据，实验环境的搭建主要包括：Hadoop 分布式系统采用 Hadoop2.9 版本，Spark 通用并行计算框架采用 Spark2.2.1 版本，采用 Standalone 集群部署方式，Java 使用的是 Java1.8.0 版本，python 使用 python2.7 版本，Pycharm 集成编译环境。分布式集群环境搭建在三台 Ubuntu 64-bit 虚拟机上，一台做为 Master 主节点，另外两台做为 Slave 节点。

依据一般的数据纠错分析方式，首先针对选取的目标数据利用相关性分析结合相应的专家知识，找到对目标数据影响较大的几个相关特征数据，通过多次特征分析实验选择最合适的几个相关数据作为目标纠错数据的特性属性，对特征进行规范化、标准化等数据预处理操作，然后训练模型，评估模型好坏，最后选择最合适的模型去完成目标数据的异常检测，发现异常错误，提取分析，以待进一步处理。具体流程如图 4 所示。

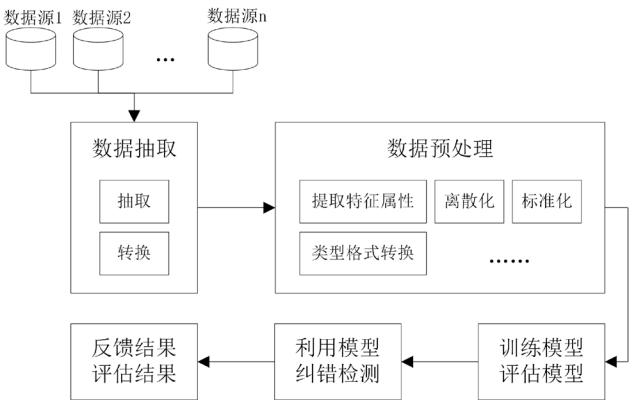

Figure 4. Data error correction process of TMS
图 4. TMS 数据纠错流程

3.1. 特征选择与数据抽取

本文以站点检修次数异常纠错分析为例，站点检修次数是项目业务中常见的数据，对于衡量不同站点管理信息有着直观明显的作用，同时也是录入信息时常出现错误的地方。在实验前可以通过相应的专家经验知识，再结合基尼系数、特征选择等相关性分析技术，确定对站点检修次数影响较大的几个特征数据，包括：站点类型、站点建成年限、调度等级以及站点设备数量等等组合特征。由此可以提取相关数据，建立如表 2 所示的结构，以方便学习算法建立模型实验使用。

将表 2 做为样本存储空间，其中每条记录就是一个样本数据。Site_jxNum 为类别标号，其他数据字段为属性特征值，对样本数据做预处理，使用方便训练的数据格式开始模型训练实验。

Table 2. Sample data table structure of station maintenance times
表 2. 站点检修次数样本数据表结构

字段名	属性描述	数据类型	说明
ID	对象 ID	Varchar	主键
Site_jxNum	站点检修次数	Number	做为纠错的目标数据，也是分类训练的类别标号
Site_name	站点名称	Varchar	包含了特定的站点类别标识名词 35 kV、11 kV、220 kV、500 kV
Site_type	站点类型	Number	站点类别标识
Site_years	站点建立年份	Number	
Site_nuNum	站点设备数目	Number	
Site_level	站点调度等级	Number	

3.2. 数据预处理

从 TMS 数据库中不同数据源中提取相关数据，整合生成存储站点检修次数及其相关的属性数据字段的表。其中除了 name 属性字段属于字符串的文本模式，其它属性字段都是离散的数值类型，是便于大数据处理分析的形式。经过专家经验知识和过往大量数据分析发现，字段 site_name 和 site_type 字段主要起到判别站点业务类型的作用。依据 site_type 字段可以将站点业务类型划分为五大类：中心站、供电所、县区供电企业、乡镇供电所、电厂以及其它。而 site_name 中包含了明显具有标识意义的名词结构，可以将 site_type 字段划分的其它类别再进一步划分出四类：35 kV、110 kV、220 kV、500 kV。由此把两个有关联的属性值 site_type 和 site_name 简化为一个数值衡量的单一独立特征属性值，可以构成站点业务类型：35 kV、110 kV、220 kV、500 kV、中心站、供电所、县区供电企业、乡镇供电所、电厂以及其它，分别用 0 到 10 这十一位整数编码代表。然而站点业务类型是具有明显文本名词特性的属性值，简单的用数值替换会出现精度上的误差，因为所选取的替换数值大小可能会对同一特征在样本中的权重造成不同的影响，例如选择数值 1000 一定会比 1 为编码时对模型的选择影响更大。因此选择通过独热编码的方式减少编码数值对模型训练可能造成的不良影响。独热编码，又称一位有效编码，即根据特征状态的个数，采用 N 位状态寄存器去描述 N 个不同的状态，其中在某一状态下只有一位被置为 1。面对多个离散特征时，可以采用独立编码而后拼接的方式实现得到最后的独热编码。

处理完文本数据后，针对部分字段缺失的样本，通过统计发现占比不高，于是通过填入 0 值的方式填充数据。之后对数据做标准化处理，消除特征数据间的量纲关系，以便参数寻优问题的求解，需要将数据按一定比例缩放，使得每个属性特征值都聚集在 0 附近，方差为 1。

由实践生产的样本数据类别繁多，且分布不均衡，通过统计发现低值类别中样本数据充足，而高值类别的样本数据较少。于是整合其他省份的相似数据补足，并通过重新划分类别等方法扩充样本空间。

最后将经过数据预处理的样本数据的数据格式采用类别标签、特征索引和特征值组合的形式，其中类别标签为从 0 开始的连续整数，特征索引为从 1 开始的连续整数，和特征值间用冒号隔开，其它的数值间用空格分隔。

将数据预处理结束后得到的离散的标准化的特征矩阵，按照所需 txt 格式上传到 Hadoop 平台的 HDFS 分布式文件系统中保存，以待之后模型训练算法的读取。

3.3. 训练纠错模型

选择将分布式支持向量机的模型构造重点放在对凸二次规划的对偶问题的并行求解上。而模型参数的求解思路也很明确，结合 Spark.GradientDescent 求解器 oprimizer 在 SVMWithSGD 中迭代求解最优模型参数，梯度计算：$\text{gradient} = -(2y-1)*x$，梯度更新方法：$\omega = \omega - \alpha(\text{gradient} + \text{regParam} * \omega)$。计算输入参数包括：data 样本输入数据；gradient 梯度对象，用于对每个样本计算梯度及误差；updater 权重更新对象，用于每次更新权重；stepSize 初始步长；numIterations 迭代次数；regParam 正则化参数；miniBatchFraction 迭代因子，每次迭代参与计算的样本比例。具体模型参数求解流程以及模型建立逻辑如下：

(1) SVMWithSGD 建立 SVM 分类模型入口，定义训练方法 train()，设置迭代次数，选择 hinge 损失函数；

(2) Train()方法调用 GradientDescent 求解器利用随机梯度下降迭代求解 SVM 模型参数[10]；

(3) 求解器参数 miniBatchFraction 做为迭代因子，利用每次迭代参与计算的样本比例对样本划分抽取，获得小样本集；

(4) 计算小样本集上的梯度值 gradient；

(5) 根据迭代步长、正则化系数、迭代次数等参数，利用 treeAggregate 的 RDD 操作实现分布式聚合计算更新模型权重；

(6) 判断终止条件是否达成，即精度收敛或者迭代次数达到上限，否则继续上面步骤对抽样数据集进行迭代计算，从而找出最优的特征权重向量解，利用随机梯度下降求解模型参数；

(7) 返回模型参数，建立 SVMModel，输出 LinearSVC。

考虑到线性支持向量机是针对二分类问题的分类器，同时通过随机梯度下降实现的线性支持向量机 SVMWithSGD 也是一个二分类器，所以还需要将二分类器推广到实践中的多分类问题。于是基于常见的一对多思想(one-VS-all)的多分类训练思想，它可以针对 K 分类问题，训练 K 个二分类器，每个分类器将把某一类归为一类，其它不属于该类的样本划为另一类，最终分类结果考虑 K 个分类器的分类函数值，选择其中最大的那一类为分类结果。因此实现思路便可以通过随机梯度下降优化的支持向量机 SVMWithSGD 结合 One-vs-Rest classifier 完成支持向量机在分布式框架 Spark 上的多分类实现。然后实验中模型将采用随机交叉检验的方式，以随机挑选的 80%站点进行训练，剩余 20%站点进行测试。训练时以站点的特征为输入，站点的检修次数为输出。模型的好坏通过模型在测试集上的预测检修次数与实际检修次数的拟合优度 R^2 决定。

实验流程主要包括：

(1) 数据预处理包括数据清洗、数据转换、规范化等操作。

(2) 将经过数据预处理后样本数据上传到 Hadoop 分布式文件系统中。

(3) 通过 Load 方法加载 HDFS 中保存的数据，存储到规范格式的 Spark.sql.dataframe 数据类型中，Label 标签为类别，Features 标签为特征属性。用随机交叉检验的方式，以随机挑选的 80%站点进行训练，剩余 20%站点进行测试。训练时以站点的特征为输入，站点的检修数量为输出[11]。

(4) 设置 LinearSVC 即线性支持向量机分类的迭代参数。多次实验对比不同迭代参数下模型评分，确定最优的迭代参数。

(5) 以 LinearSVC 为基础分类器，通过 One-vs-Rest 建立多分类评估器，训练并创建模型[12]。

(6) 通过模型在测试集上的预测检修次数与实际检修次数的拟合优度 R^2 评估模型好坏。

(7) 利用选取的模型进行数据纠错，检测异常数据，将异常数据导出保存，等待进一步分析处理。

在模型训练中包含了可以自定义模型参数，即迭代次数、迭代步长和正则化系数。为了提高模型性能，对三个参数进行多次基于对比实验的参数调优，确定下了较优的参数组合：最大迭代次数 20，迭代步长 1.0，正则化系数 0.1。使用训练数据为样本数据中交叉随机抽取的一千条样本记录。

针对三个参数，分别采用网格搜索对比不同取值下训练模型的评分高低及变化趋势可以发现，迭代次数在增长到一定值后，模型准确率的增长相比于时间开销的增长显得不够明显，对结果影响较小。迭代步长影响了模型梯度下降收敛的快慢，但步长过大会导致下降精度不够，可能得到局部最优解。正则化系数则是为了防止训练模型过拟合，低数值的正则化系数对准确率影响不大，过大的系数可能会因为欠拟合影响模型性能。对比实验结果如图 5、图 6 和图 7 所示。

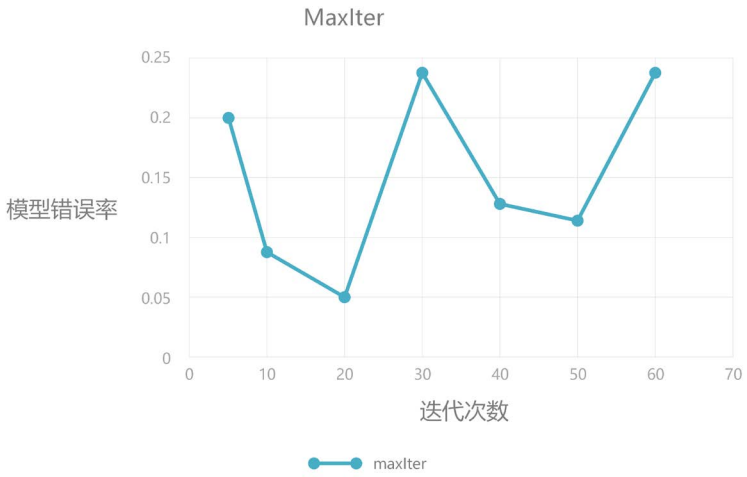

Figure 5. Optimization results of iterations
图 5. 迭代次数调优结果

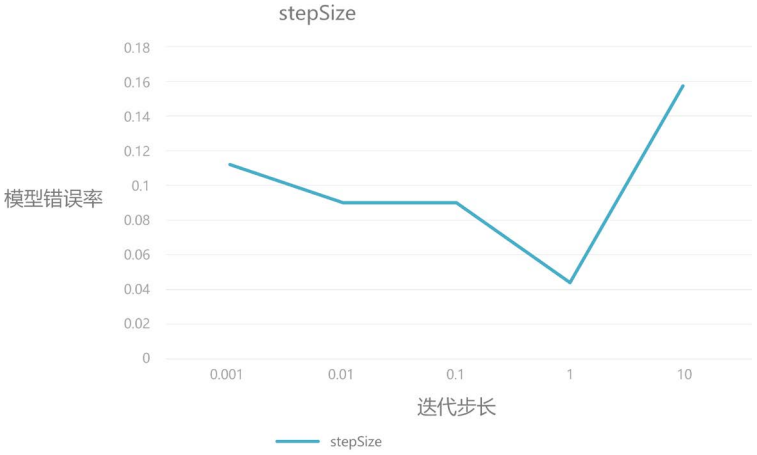

Figure 6. Optimization results of iterative step size
图 6. 迭代步长调优结果

3.4. 纠错结果分析

利用来自 UCI/Adult 的公共数据集 a9a，包含 32561 条数据，123 个特征 [2]，使用 LibSVM 算法包中

[2] http://archive.ics.uci.edu/ml/datasets/Adult.

的单机支持向量机算法和我们在 Spark 下实现的分布式支持向量机算法 SVMWithSGD 做对比实验，可以发现在准确率相近的情况下，时间开销有了明显的提升，如表 3 所示。

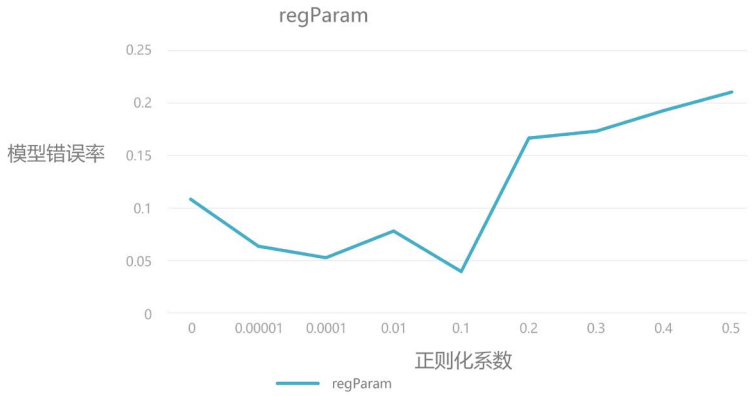

Figure 7. Optimization results of regularization coefficient
图 7. 正则化系数调优结果

Table 3. Comparison of parallel and distributed experiments
表 3. 并行和分布式实验的对比

a9a		
	准确率	执行时间(s)
SVMWithSGD	82.83%	30
LibSVM	84.29%	130

在利用待分析的业务数据作为样本，使用数据源为待分析的省份业务数据，包含一万条记录，16 个特征数。实验结果准确率和召回率相近的情况下，时间开销可以缩减近一半。结果如表 4 所示。

Table 4. Contrast result
表 4. 对比结果

样本数据			
	准确率	召回率	执行时间(s)
SVMWithSGD	88.43%	89.2%	11
LibSVM	89.96%	90%	19

然后将模型应用到业务数据的纠错预测上。选取出一个省份的业务数据，经过数据预处理后共一万条记录，包含特征数 16 条。利用训练好的模型做为整体的规律，对所有站点进行预测。若预测结果和实际数据相差较大，则说明该站点数据和整体规律不符合，有异常存在的可能。将综合模型预测得到的站点检修次数和数据库中记录的站点检修次数的差值和比值划分站点数据的异常等级。用 score 表示异常评分，prediction 表示通过模型得到的预测值，data 表示数据库中记录的样本值，则依据公式(5)可得站点数据的异常评分。

$$\text{score} = \left(\left|\text{prediction} - \text{data}\right|\right)^{\text{pediction}/\text{data}} \tag{5}$$

异常评分越高，异常等级就越高，表示数据异常偏差越大，急需排查。

同时随机抽取 1000 个正常站点，删除其站点检修次数后再次进行模型检测，发现有 892 个站点异常等级被提升，召回率可达 89.2%。

4. 结语

基于 TMS 中产生的大量数据信息有待分析总结和传统技术架构的 TMS 在数据分析处理方面缺乏有效手段的背景，针对数据信息存在的账务和实物不一致、数据录入错误以及存在缺失数据等问题，本文首先在讨论支持向量机(SVM)的基础上，进一步讨论 Spark-MLlib 是如何通过 Spark 分布式计算框架实现支持向量机的并行训练模型，然后以探索站点检修次数的异常数据检测为例，尝试将基于 Hadoop 分布式集群基础框架和 Spark 通用并行计算平台上实现的分布式支持向量机算法应用到 TMS 系统数据纠错中，提出一种针对 TMS 系统数据站点检修次数中的异常数据纠察分析的解决方法，从 TMS 系统数据分析入手，针对经常出错的目标数据，完成了基于 Spark 分布式支持向量机的 TMS 数据纠错实现及实验分析。为人工纠察数据异常难，开销大的问题提供了一种 AI 辅助解决方法。

基金项目

楚雄师范学院科学研究基金项目(数字化校园信息系统开发研究专项项目)，项目名称：自定义播放的数据缓存视频服务系统的研究与实现，项目编号：SZZX1303，项目负责人：王松。

参考文献

[1] 维克托•迈尔•舍恩伯格. 大数据时代：生活、工作与思维的大变革[M]. 周涛译. 杭州: 浙江人民出版社, 2012: 8-10.

[2] 杨斌, 杨济海. 大数据在电力系统通信中的应用[J]. 电子技术应用, 2015, 41(SI): 394-396, 400.

[3] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016: 121-132.

[4] Bello-Orgaz, G., Jung, J.J. and Camacho, D. (2016) Social Big Data: Recent Achievements and New Challenges. *Information Fusion*, **28**, 45-59. https://doi.org/10.1016/j.inffus.2015.08.005

[5] Apache Spark 官方网站[EB/OL]. http://spark.apache.org/, 2019-10-15.

[6] Apache Spark ML 官方指导文献[EB/OL]. http://spark.apache.org/docs/latest/ml-guide.html, 2019-11-05.

[7] 上海交通大学模式分析与机器智能实验室. LibSVM-2.6 程序代码注释[EB/OL]. http://www.doc88.com/p-6159926915557.html, 2018-05-07.

[8] Xie, Z.X. and Li, Y.D. (2019) Large-Scale Support Vector Regression with Budgeted Stochastic Gradient Descent. *International Journal of Machine Learning and Cybernetics*, **10**, 1529-1541. https://doi.org/10.1007/s13042-018-0832-7

[9] 陶杭. 基于 Hadoop 的 SVM 算法优化及在文本分类中的应用[D]: [硕士学位论文]. 北京: 北京邮电大学, 2015.

[10] 吴云蔚, 宁芊. 基于 Hadoop 平台的分布式 SVM 参数寻优[J]. 计算机工程与科学, 2017, 39(6): 1042-1047.

[11] 邹红旭, 潘冠华, 李吟. 基于 Spark 框架的改进协同过滤算法[J]. 计算机技术与发展, 2020(5): 1-8.

[12] 李君娣, 张正军, 庄立纯, 等. 基于分类属性 IG 比的多分类 SVM 结构评价方法[J]. 计算机工程与科学, 2019, 41(4): 719-726.