DOI:10.16644/j.cnki.cn33-1094/tp.2016.04.001

基于Spark的大规模网络流量分类研究*

刘泽燊，潘志松

(解放军理工大学指挥信息系统学院，江苏 南京 210007)

摘 要：机器学习算法处理流量分类问题已经成为网络安全领域一个研究热点。为了提高大规模网络流的分类效率，引入并行SVM算法来识别网络流量，提出了一种基于Spark平台的大规模网络流在线分类方案。该方案利用置信域牛顿法(TRON)并行优化线性SVM算法构建流量分类模型，然后融合最新的实时计算框架，实现对大规模网络流的在线识别。实验结果表明，利用并行SVM算法在损失较小精度的前提下可以加快网络流的模型训练和分类速度，符合大规模网络流在线分类的需要。

关键词：流量分类；网络安全；Spark；并行SVM；大规模数据

中图分类号：TP391 **文献标志码**：A **文章编号**：1006-8228(2016)04-01-05

Study on large scale network traffic classification on Spark platform

Liu Zeshen, Pan Zhisong

(*College of Command Information System, PLA University of Science and Technology, Nanjing, Jiangsu 210007, China*)

Abstract: Internet traffic classification using machine learning has become a hot research topic in the field of network security. In order to improve the classification efficiency of large scale network flow, this paper introduces a parallel SVM algorithm to identify the network traffic, and proposes a real-time classification scheme for large scale network flow based on Spark. This method builds a classification model using parallel SVM algorithm, and then it is integrated with the latest flow processing framework for real-time classification of large-scale networks. Experimental results show that parallel SVM algorithm can greatly improve the training and classification speed of the network flow model, on the premise of little loss of precision.

Key words: traffic classification; network security; Spark; parallel SVM; large scale data

0 引言

随着互联网的快速发展，网络安全和网络拥塞等问题也日益严重。为了更好的识别异常流量及优化配置网络资源，必须准确分类网络流中各种应用类型。因此对大规模网络流量进行快速、准确的分类具有十分重要的意义。

近年来网络应用多元化的发展趋势给网络流分类带来一系列的挑战，动态端口以及随机端口技术的出现，使得最初分析端口号的分类手段已经不能准确的对P2P等新型应用进行分类[1]，同时基于有效负载的方法很难处理加密流量[2]。针对上述网络流分类技术日益凸显的缺点，将流量分类问题转化为机器学习问题成为当前研究趋势。在聚类算法中，Erman等人[3]分析了K-Means、DBSCAN和AutoClass三种聚类算法的网络流分类性能，因为此类无监督算法无需使用训练样本的类标，所以能够识别新型网络应用，但是聚类完成后必须进行人工标记，同时整体分类精度偏低。对于分类算法，Moore等人[4]将朴素贝叶斯和改进的贝叶斯方法应用到网络流分类，能将准确率提高到95%，但是贝叶斯算法要求样本特征遵循高斯分布，

收稿日期：2015-12-28

***基金项目**：国家自然科学基金项目(61473149)

作者简介：刘泽燊(1991-)，男，湖南怀化人，硕士研究生，主要研究方向：机器学习和分布式计算。

通讯作者：潘志松(1973－)，男，教授，博士生导师，主要研究方向：模式识别，机器学习和网络安全。

然而实际的网络流数据很难满足，所以具有不稳定等缺点。徐鹏等人[5-6]针对贝叶斯分类过分依赖样本空间分布的问题，提出了利用训练数据信息熵的决策树分类方法和基于结构风险最小化的SVM分类方法，这两种方法都取得了较高的准确率和稳定性，但是决策树学习高维样本时具有很高的复杂度，而SVM算法训练大规模样本时存在时间长和内存占用大等问题。Yang等人[7]则将lasso特征选择算法应用到网络流异常检测，通过降低训练样本维度加快了模型训练以及流量识别速度。Groleat等人[8]则基于FPGA设计了一个用于在线检测的实时SVM分类器，通过硬件加速，极大的提高了检测的效率。通过上述研究，基于机器学习的网络流分类取得了一定的进展，然而对大规模流量分类缺乏实时响应，同时很少有人关注网络流在线识别。Spark[9]作为一种新的计算框架，通过扩展集群能够对大规模数据进行快速处理，同时具有自动处理失效节点和负载均衡的功能。

本文深入分析了单机SVM算法分类网络流的不足之处，以分布式计算为基础，给出了一种基于Spark平台的大规模网络流在线分类方案 。主要工作有：①应用置信域牛顿法快速优化分布式的SVM分类算法；②对比分析了大规模网络流在线分类方案跟单机SVM算法的性能。实验结果表明，该方案不但能够快速训练分类模型和识别流量数据，而且具有较高的准确率和稳定性。

1 Apache Spark技术

Spark是由UC Berkeley大学开源的新一代大数据处理平台，是一种集流式计算、数据查询、机器学习和图挖掘于一体的通用计算框架。在迭代计算中，基于内存运算的Spark平台可将中间结果持久的保存于内存中，适合迭代运算和交互式数据分析。

弹性分布式数据集（RDD）[10]是一种支持多次重复利用的数据抽象，而且在并行计算阶段可以高效的共享数据，是Spark平台的数据存储核心和基础。RDD是只读、可分区和高效容错的分布式内存数据集，每个RDD是由许多数据分片组成的对象集合，它只能通过物理存储在本地或者分布式文件系统上的数据集，以及其他已有的RDD通过一些特定数据操作来创建，并且RDD仅支持粗粒度数据转换。Spark在RDD上提供转换(Transformations)和动作(Actions)两类并行编程算子，用户通过操作并行算子可以显式的控制RDD缓存、物化以及分区策略等。转换操作主要包括map、union、join等，是一种不会立即执行的懒操作，预期由一个已有RDD创建一个新RDD。动作操作则包括reduce、count、save等，会真正触发预先设定在RDD数据上的计算流程，结果是返回一个普通类型或者将RDD数据输出并保存到外部存储系统。经过转换操作的RDD之间相互形成依赖关系，如图1所示。RDD的依赖关系有两种：窄依赖和宽依赖，其中窄依赖是子RDD的分区只依赖父RDD的某个分区，而宽依赖则指子RDD的每个分区依赖父RDD的多个分区。不同的依赖关系对划分作业阶段以及调度都有重要意义。

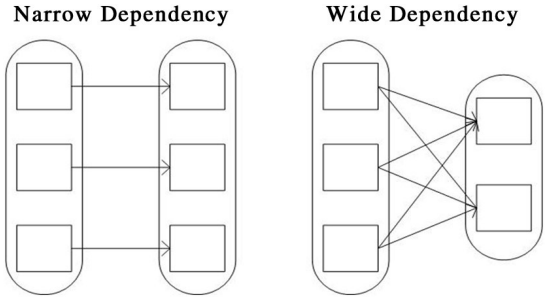

图1 窄依赖和宽依赖

2 基于Spark的大规模网络流在线分类方案

主干网络流量的急剧增长对现有的网络流分类方案提出了更高的性能要求，不仅需要系统拥有强大的处理能力能够短时间内对大规模数据进行模型训练，还要求在线分类时响应速度快，并准确的对网络应用进行实时分类。针对现有流量分类技术不能很好的分类大规模网络数据，提出了一种以Spark分布式平台为基础的大规模网络流在线分类方案。

2.1 大规模网络流在线分类方案

这个大规模网络流分类架构由离线模型训练和在线实时分类两部分组成。如图2所示，在离线训练阶段，首先需要实时采集原始网络流量，然后数据预处理步骤从原始流中提取特征(如端口号、报文生存时间、最大分组长度)并进行补全和归一化等处理，接着将已标注应用类型的流量样本导入分布式文件系统(HDFS)中作为训练集。最后在训练分类模型时，如果数据样本庞大，通常可以进行数据抽样或者特征选择以降低训练复杂度，但可能会导致分类准确率降低或结果不稳定。为了保证在线分类的性能需求，基于Spark内存计算平台的并行SVM分类算法在建立分类模型时训练所有网络数据，所以在加快模型训练和实

时分类的同时可以保证模型的稳定性和准确率。此外,可利用采集的实时网络流量更新分类模型以达到更好的测试效果。在线分类阶段主要利用 Spark Streaming 流数据处理模块和流量属性特征进行网络流实时分类。首先根据分类器的需求提取相应的网络流量特征作为测试样本,并将其组织成流数据;然后 Spark Streaming 模块加载离线训练完成的分类器,并使用分类模型对输入的实时流量进行预测处理;最后输出系统识别的网络应用类型。

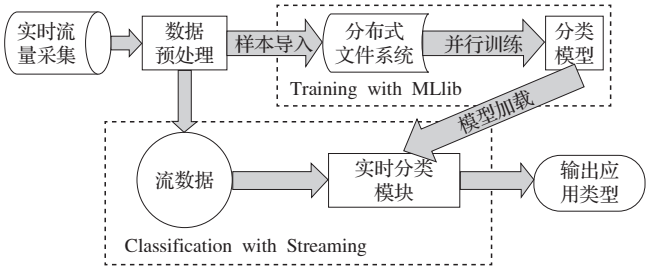

图2 大规模在线分类方案

2.2 基于分布式置信域牛顿法的线性SVM算法

支持向量机(SVM)在线性可分情况下的学习策略是使边缘超平面的距离最大化,寻找具有最大边缘距离(Maximum Marginal)的分类超平面。通常情况下,SVM算法使用一个非线性的核函数将样本数据映射到高维空间进行线性分类。由于非线性核映射对高维数据分类的性能影响不大,因此才能直接优化没有使用非线性映射的大规模线性SVM问题。对于给定训练数据集 $X=\{(x_i,y_i):i=1,2,\cdots,n\}$,其中输入样本 $x_i\in R^d$,类标 $y_i\in\{-1,1\}$,带有正则化函数的线性SVM算法(L2-SVM)优化模型如下:

$$\min_{w} f(w)=\frac{1}{2}w^Tw+C\sum_{i}^{n}\max(0,1-y_iw^Tx_i)^2 \qquad (1)$$

由于L2-SVM算法的目标函数是可微的,所以有很多无约束优化算法可以训练求解。二阶优化的置信域牛顿法因在迭代中具有较快的收敛速度,所以在大规模数据的场景中,十分适合优化L2-SVM模型。其基本思想是,下一步迭代更新的方向被限制在当前搜索点的一个有效范围内。为了最小化函数f(w),在置信域牛顿法的每一轮迭代中可得到如下二次规划问题:

$$\min_{s} q_i(s)=\frac{1}{2}s^T\nabla^2 f(w^i)s+\nabla f(w^i)^Ts \quad s.t.\ \|s\|\le\Delta_i \qquad (2)$$

其中,w^i是当前迭代的权重,Δ_i是置信域半径。然后利用式(3)与设定的阈值进行比较,更新权重。

$$w^{i+1}=\begin{cases}w^i+s^i & if\ \rho_i>\eta\\ w^i & if\ \rho_i\le\eta\end{cases} \qquad (3)$$

对于并行TRON算法在Spark上实现的主要思想是利用Slave节点并行计算每个数据分片的 $f(w^i)$、$\nabla f(w^i)$和 $\nabla^2 f(w^i)$的值,将结果传输到master节点进行汇总和计算更新。算法1详细描述了在Spark集群上利用分布式的TRON算法求解线性支持向量机。

算法1:优化线性SVM的分布式TRON算法-Spark

1. 给定初始的:$w^0,\Delta_0,\eta,\varepsilon$
2. For i=0,1,…
3. 主节点(master)广播(broadcast)w^i值到从节点(slave)
4. Slaves 计算 $f(w^i)$和 $\nabla f(w^i)$,然后发送给master
5. 如果 $\|\nabla f(w^i)\|<\varepsilon$,算法结束
6. 通过计算公式(2)得到 s^i
7. 计算 $\rho_i=\dfrac{f(w^i+s^i)-f(w^i)}{q_i(s^i)}$
8. 根据下式将 w^i 更新到 w^{i+1}

$$w^{i+1}=\begin{cases}w^i+s^i & if\ \rho_i>\eta\\ w^i & if\ \rho_i\le\eta\end{cases}$$

9. 得到 Δ_{i+1}

3 实验

3.1 实验环境和数据集

本文使用的数据挖掘工具是LibSVM3.17[11]和Spark LIBLINEAR[12]。对于单机SVM算法选择了径向基核函数(RBF), 同时利用步长搜索方法得到了参数惩罚因子C=586和核参数g=0.0078125。Spark集群由5台Dell服务器组成,分为1台Master节点和4台Slave节点。硬件配置:机器为8G内存,四核CPU,i5处理器。软件配置:集群搭建使用spark-1.3.0-bin-hadoop1稳定版,Java选用JDK1.7.0_71,Scala则为Scala-2.10.3,操作系统选择ubuntu12.04Server版。

实验采用Moore数据集[13],一共包含10个流量子集,共有377526个流量样本,被分为12种应用类型,每个样本由248维特征和1个类标组成。为了让数据符合分类算法的要求,在预处理时将缺失的特征设置为0,使用{1,-1}代替原始样本的特征值{Y,N},并且将所有的特征值归一化到区间[1,-1]之间。

3.2 实验结果与性能分析

3.2.1 算法时间比

为了对比分析并行SVM算法和单机SVM算法在真实网络流数据集上的训练时间和测试时间,首先从

所有Moore数据上随机选取了2万、4万、8万、12万、16万和20万等样本作为训练集和测试集，然后利用并行SVM算法和单机SVM分别训练分类模型并进行流量识别。相应的时间如表1所示。

表1 训练和分类时间对比

	并行SVM(s)		单机SVM(s)	
	训练时间	测试时间	训练时间	测试时间
20000	28	0.008	88	73
40000	37	0.008	257	200
80000	46	0.008	780	561
120000	46	0.009	1326	1059
160000	49	0.008	3350	1600
200000	51	0.009	4915	2278

对比表1结果可看出，随着数据量的增加并行SVM算法比单机SVM在训练时间和测试时间上都有明显的优势。这是因为模型训练时并行SVM算法将训练数据分散到Spark集群的Worker节点进行分布式计算，从而缩短了算法训练时间；而单机SVM利用SMO优化算法训练支持向量，所需时间为$O(m^2)$，其中m为样本数量，因此不适合进行大规模数据集的训练。随后，在进行流量分类时，并行SVM算法只需将样本特征输入分类模型即可得到样本类别，处理比较简单；而单机SVM算法则需要将测试样本跟每个支持向量进行内积运算，过程相对复杂。

3.2.2 数据扩展性测试

为了评估数据规模变大时算法运行时间的变化情况，本实验通过将数据集进行简单的组合复制得到了从10万到60万不等的6个训练集，并行SVM算法训练时间的实际变化情况如图3所示，从中可看出，随着数据规模的增大，并行SVM算法的运行时间基本呈线性增长走势，即具备较理想的数据可扩展性。

3.2.3 系统扩展性测试

集群的规模很大程度上决定了算法的运行时间，为了发现并行SVM算法的性能随着集群规模增大的变化情况，我们在相同数据集下进行了系统扩展性相关实验。测试结果如图4所示，随着节点数量的增加，算法的训练时间首先会迅速下降，然后基本稳定在一个固定数值。这是因为节点越多意味着集群能够同时处理的数据分片也越多，从而降低整个程序的运行时间。而最终所达到的稳定状态是由于集群规模的进一步扩大，节点间的通信开销迅速增加，所增加的系统额外开销和降低的任务排队时间维持一个稳定水平，因此并行程序的运行时间基本保持不变。由算法训练相同数据集的计算时间随着集群规模增大而呈下降趋势，可见系统具有很好的可扩展性，所以能够通过扩大集群规模，解决大规模的网络流实时分类问题。

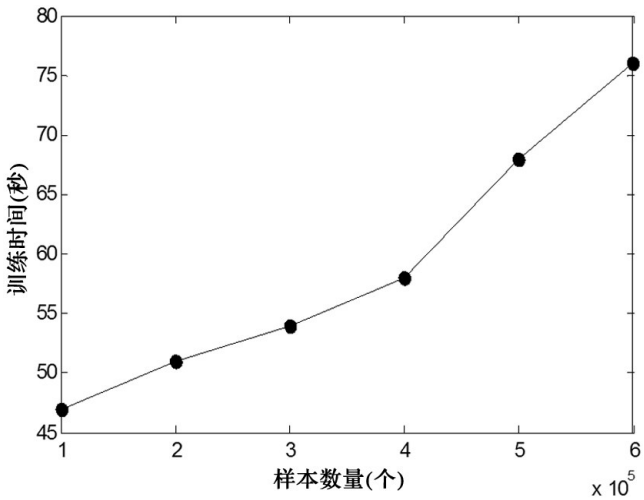

图3 训练时间与数据量关系

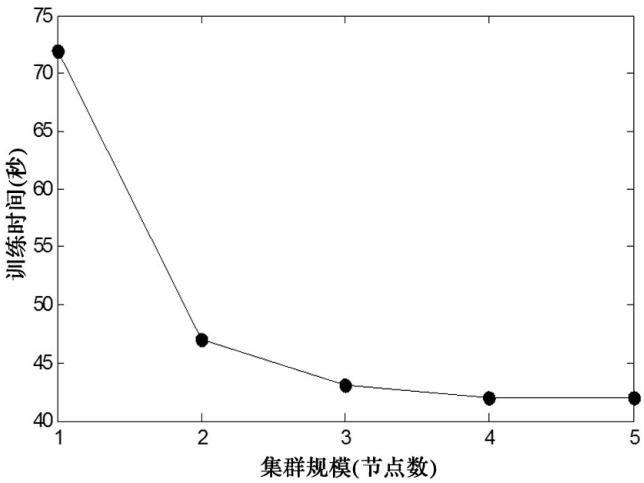

图4 训练时间与集群大小关系

3.2.4 算法准确性指标

为了对比分析并行SVM算法和单机SVM的整体准确率，首先随机将Moore数据子集的50%划分为训练集，另外50%构成测试集，然后分别运行并行SVM和单机SVM算法得到网络流分类模型，并使用模型识别测试集以获得相应准确率。从图5中可看出，并行SVM算法的整体准确率略低于单机SVM，这是因为并行SVM算法没有利用核函数将样本映射到高维进行分类，并且网络数据的维度也不是很高，所以在降低分类模型训练负载的同时，也损失了一点准确率。

但并行SVM算法的准确率也大都在99%以上，同样具有较高的分类准确率，因此能够有效的识别大规模网络应用。

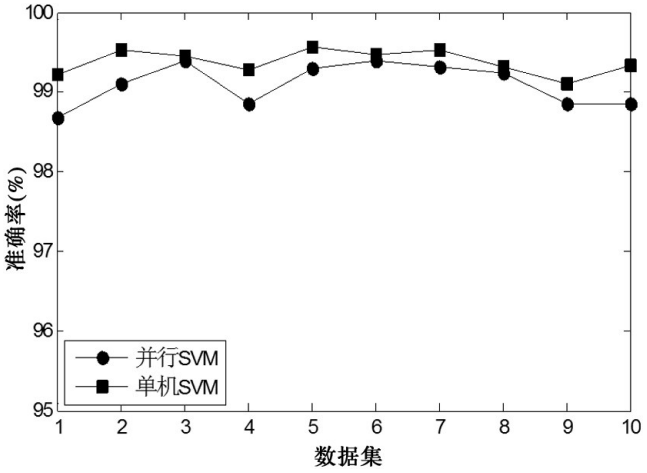

图5 整体分类准确率

4 结束语

利用机器学习方法分类流量是近年来网络流分类领域的一个研究热点。本文针对单机SVM算法分类大规模流量的时间复杂度高、内存消耗大等问题，提出了一种基于Spark的大规模网络流在线分类方案。通过真实网络数据的对比实验可得到并行SVM算法的训练时间和测试时间远远小于单机SVM算法，同时，在Spark集群上具有很好的数据及系统可扩展性，但是并行SVM的准确性指标略低于单机SVM算法。由于目前并行SVM算法没有使用核函数，所以准确性指标不如利用核映射分类的单机SVM算法。因此，如何把核函数应用到基于Spark的并行SVM算法将是我们未来的研究重点。

参考文献(References)：

[1] Madhukar A, Williamson C. A Longitudinal Study of P2P Traffic Classification[C]//Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on. IEEE,2006:179–188

[2] Kumar, Sailesh, Dharmapurikar, Sarang, Yu, Fang, et al. Algorithms to accelerate multiple regular expressions matching for deep packet inspection[M].ACM,2006.

[3] Erman J, Arlitt M, Mahanti A. Traffic Classification Using Clustering Algorithms[C]//In ACM SIGCOMM MineNet Workshop,2006:281–286

[4] Moore A W, Zuev D. Internet traffic classification using bayesian analysis techniques[J]. Acm Sigmetrics Performance Evaluation Review, 2005.33(1):50–60

[5] 徐鹏,林森.基于C4.5决策树的流量分类方法[J]. 软件学报, 2009.20(10):2692–2704

[6] 徐鹏,刘琼,林森.基于支持向量机的Internet流量分类研究[J]. 计算机研究与发展,2009.3:407–414

[7] Yang L, Hu G, Li D, et al. Anomaly detection based on efficient Euclidean projection[J]. Security & Communication Networks,2015.

[8] Groleat T, Arzel M, Vaton S. Hardware Acceleration of SVM–Based Traffic Classification on FPGA[C]// Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International. IEEE, 2012:443–449

[9] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica,"Spark: cluster computing with working sets," in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing USENIX Association, 2010:10–10

[10] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley,M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets:A fault–tolerant abstraction for in–memory cluster computing," in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation,2012.

[11] Chang C C, Lin C J. LIBSVM: a Library for Support Vector Machines[J]. Acm Transactions on Intelligent Systems & Technology,2006.2(3):389–396

[12] Lin C Y, Tsai C H, Lee C P, et al. Large–scale logistic regression and linear support vector machines using spark[C]//Big Data (Big Data),2014 IEEE International Conference on. IEEE,2014:519–528

[13] Moore A, Zuev D, Crogan M. Discriminators for use in flow–based classification[J]. Department of Computer Science Research Reports;RR–05–13–August 2005.