

基于半监督学习的应用流分类方法

柳 斌, 李之棠, 涂 浩

(华中科技大学 网络与计算中心, 湖北 武汉 430074)

摘 要: 将半监督学习应用到应用流分类问题中, 提出了一种基于半监督聚类的应用流分类算法(PSOSC). 首先采用粒子群优化的 K 均值聚类方法对大量的无标记数据和少量的标记数据进行聚类, 利用少量标记数据确定簇与应用类型的映射关系, 实现应用流分类. 实验表明 PSOSC 算法有较高的流准确率, 同时, 降低了对标记数据的需求.

关键词: 流量分类; 半监督学习; 特征选择

中图分类号: TP393.08

文献标识码: A

文章编号: 1000-7180(2010)08-0024-03

A Semi-Supervised Clustering Method for Network Application Flow Classification

LIU Bin, LI Zhi-tang, TU Hao

(Network Centre, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: In this paper, a feature selection method based on entropy is proposed firstly. Five TCP connection features are selected for classifier. Then, a semi-supervised classification method that allows classifiers to be designed from training data consisting of only a few labelled and many unlabelled is proposed. The experiment shows that the detection results precedes on Kmeans method.

Key words: traffic classification; Semi Supervised Learning; feature selection; entropy

1 引言

应用层是 TCP/IP 网络体系架构中的最高层. 按照应用层对流进行识别和分类在用户行为分析、网络带宽管理、入侵检测等诸多方面有着十分重要的应用. 应用流分类本质上是一个模式分类问题, 因而, 可以将机器学习技术应用到流量分类领域^[1-2]. 在机器学习领域, 分为有监督学习和无监督学习. 基于监督学习的方法如贝叶斯方法、决策树方法等, 检测率高, 但要求样本数据事先正确标记类别, 因此无法发现未知的类别, 且需要足够的训练数据, 以生成具有良好的泛化性能的检测模型. 非监督学习方法如聚类的方法, 根据数据的相似性进行分组, 克服了监督学习方法中需要标记数据的不足, 但其检测精度明显低于有监督的分类方法. 它只能对未标记数

据建模, 不利用标记数据的类别信息. 本文提出了一种基于半监督聚类的应用流分类算法(PSOSC), 利用少量的标记数据指导聚类过程. 首先采用粒子群优化的 K 均值算法将数据划分为不同的簇, 利用少量的标记数据标记簇, 并将标记信息扩展到簇内所有数据, 实现数据的应用类型映射. 算法能发现未知的新的应用类型, 提高了检测精度, 同时, 降低了对标记数据的需求.

2 基于粒子群优化的 K 均值聚类算法

2.1 基本思想

PSOSC 算法主要包括两个部分: 基于粒子群优化的 K 均值聚类算法和簇标记算法. K 均值算法是聚类分析的一种经典算法, 广泛应用于数据挖掘和知识发现领域中. 传统的 K 均值算法存在两个缺

点:(1)结果与初始值选取有关.对于不同的初始值可能会导致不同的聚类结果,甚至存在着无解的情况;(2)该算法是基于梯度下降的算法,因此常常陷入局部极优.

粒子群优化算法(PSO)不但具有全局寻优能力,通过调整参数,PSO还可同时具有较强的局部寻优能力,由于没有个体杂交、变异等运算操作,PSO的参数调整简单,易行.本文将粒子群算法应用到K均值算法中,给出了一种基于粒子群优化的K均值聚类算法.

设数据集为 $\mathbf{X} = \{\mathbf{X}_i, i = 1, 2, \dots, n\}$, 其中 $\mathbf{X}_i$ 为 n 维模式向量, 聚类问题就是要找到一个划分 $\omega = \{\omega_1, \omega_2, \dots, \omega_M\}$, 使得总的类内离散度和 J 达到最小.

$$J = \sum_{j=1}^M \sum_{\mathbf{X}_i \in \omega_j} d(\mathbf{X}_i, \overline{\mathbf{X}}^{(\omega_j)}) \quad (1)$$

式中, $\overline{\mathbf{X}}^{(\omega_j)}$ 为第 j 个聚类中心, $d(\mathbf{X}_i, \overline{\mathbf{X}}^{(\omega_j)})$ 为数据到对应聚类中心距离, 聚类准则函数 J 为各类数据到对应聚类中心距离的总和.

当聚类中心确定时, 数据的划分由最近邻法则决定. 对数据 $\mathbf{X}_j$, 若第 j 类的聚类中心 $\overline{\mathbf{X}}^{(\omega_j)}$ 满足式(1), 则 $\mathbf{X}_j$ 属于类 j .

$$d(\mathbf{X}_i, \overline{\mathbf{X}}^{(\omega_j)}) = \min_{t=1,2,\dots,M} d(\mathbf{X}_i, \overline{\mathbf{X}}^{(\omega_t)}) \quad (2)$$

在新的算法中, 每个粒子代表聚类中心的集合, 即: 在一个具有 M 个聚类中心, 向量维数为 n 的聚类问题中, 用粒子的位置值表示 M 个聚类中心.

每个粒子 i 由位置、速度和适应度值三部分组成. 粒子结构 i 表示如下:

$[\overline{\mathbf{X}}^{(\omega_1)}, \overline{\mathbf{X}}^{(\omega_2)}, \dots, \overline{\mathbf{X}}^{(\omega_M)}]$	$V_1 V_2 \dots V_{m \times d}$	fitness
---	--------------------------------	---------

粒子的位置编码结构表示为 $[\overline{\mathbf{X}}^{(\omega_1)}, \overline{\mathbf{X}}^{(\omega_2)}, \dots, \overline{\mathbf{X}}^{(\omega_M)}]$, 其中, $\overline{\mathbf{X}}^{(\omega_j)}$ 表示第 j 类的聚类中心, $\overline{\mathbf{X}}^{(\omega_j)}$ 是一个 n 维矢量.

每个粒子还有一个速度, 其编码结构为 $[V_1, V_2, \dots, V_M]$, V_j 表示第 j 个聚类中心的速度值, V_j 也是一个 n 维矢量.

粒子适应度值 fitness 为一个实数, 表示粒子的适应度, 采用以下方法计算适应度.

$$\text{fitness} = k/J = k / \sum_{j=1}^M \sum_{\mathbf{X}_i \in \omega_j} d(\mathbf{X}_i, \overline{\mathbf{X}}^{(\omega_j)}) \quad (3)$$

式中, J 是总的类内离散度和, k 为常数. 粒子所代表的聚类划分的总类间离散度越小, 则粒子的适应度值越大.

每个粒子在进化过程中记忆一个局部最优解 P_{id} , 表示该粒子经历的最优位置和适应度值, $p_{id} = [\overline{\mathbf{X}}^{(\omega_1)}, \overline{\mathbf{X}}^{(\omega_2)}, \dots, \overline{\mathbf{X}}^{(\omega_M)}, \text{fitness}]$.

全局最优解 P_{gd} , 表示粒子群经历的最优位置和适应度, $p_{gd} = [\overline{\mathbf{X}}^{(\omega_1)}, \overline{\mathbf{X}}^{(\omega_2)}, \dots, \overline{\mathbf{X}}^{(\omega_M)}, \text{fitness}]$.

粒子的速度和位置更新公式如式(4)和式(5)所示.

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 \text{rand}() (p_{id} - x_{id}(t)) + c_2 \text{rand}() (p_{gd} - x_{id}(t)) \quad (4)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (5)$$

式中, $v_{id}(t+1)$ 表示第 i 个粒子在 $t+1$ 次迭代中第 d 维上的速度, ω 为惯性权重, c_1, c_2 为加速度常数, $\text{rand}()$ 为 $0 \sim 1$ 之间的随机数. 粒子的移动方向由三部分决定: 自己原有的速度 $v_{id}(t)$ 、与自己最佳经历的距离 $p_{id} - x_{id}(t)$ 、与群体最佳经历的距离 $p_{gd} - x_{id}(t)$.

基于粒子群优化的K均值聚类算法描述如下:

(1) 初始化粒子群, 给定参数 K , 粒子数量 m . 随机生成各类的聚类中心, 作为粒子 i 的位置编码, 计算粒子的适应度, 设置粒子 i 的各中心的初始速度为 0. 重复生成 m 个粒子.

(2) 根据初始粒子群计算粒子的个体最优位置 $P_{id}(i)$ ($i = 1, \dots, m$) 和全局最优位置 P_{gd} .

(3) 根据式(4)和(5)更新所有粒子的速度和位置.

(4) 每个数据按照最近邻法则, 确定其所在的簇.

(5) 重新计算新的聚类中心, 更新粒子的适应度值.

(6) 比较每个粒子 i 的适应度值和它经历过的最好位置的适应度值 $P_{id}(i)$, 如果更好, 更新 $P_{id}(i)$.

(7) 比较每个粒子 i 的适应度值和群体所经历的最好位置 P_{gd} 的适应度值, 如果更好, 更新 P_{gd} .

(8) 如果达到结束条件(最大迭代次数), 则结束算法. 输出 P_{gd} ; 否则转步骤(3)继续迭代.

2.2 簇标记算法

聚类生成后需要对簇进行标记. 利用已知的少量标记样本建立簇与应用的映射关系, 确定每个簇的应用类型. $P(Y = y_j | C_k)$ 表示在 C_k 条件下, 流 Y 的应用类型的概率密度分布. 其中 j 为应用类型, $k = 1 \dots K$, K 为聚类数目. 利用标记数据对每个簇的 $P(Y = y_i | C_k)$ 进行似然估计, 如下所示.

$$P(Y = y_i | C_k) = \frac{n_{jk}}{n_k}$$

式中, n_{jk} 为分配到簇 k 中标记为 j 的流的数目, n_k 为簇中总的流数目。

簇标记算法用类 C 语言描述如下:

```
while(簇不为空)
{ // 只要不为空, 循环向下查找下一个簇

    计算簇  $k$  的  $P(Y = y_i | C_k) = \frac{n_{jk}}{n_k}$ 
    if (arg max $y_1 \dots y_q$   $P(y_i | C_k) > T$ )
        // 当应用类型概率的似然估计大于某个门限
        label $i$  = arg max $y_1 \dots y_q$   $P(y_i | C_k)$ 
    Else
        label $i$  = unknown 标记为未知簇
```

3 实验与分析

3.1 特征选择

采用了文献[3]中 Andrew Moore 数据集作为原始数据集. Andrew Moore 数据集利用 TCPTRACE^[5]等工具提取了不同应用的 248 种 TCP 连接特征. 数据集一共有 32 000 条记录. Andrew Moore 数据集将应用分成了 6 类: (1) BULK 类, 如 FTP, xunlei 应用等; (2) Interactive 类, 如 telnet 应用等; (3) WWW 类, 如 http, https 应用; (4) Service 类, 如 Dns 应用; (5) P2P 类, 如 bittorrent, skybe, edonkey 应用; (6) Mail 类, 如 POP3, SMTP 应用等. 采用特征采用了表 1 所示的 11 种特征.

表 1 流特征

序号	特征名称	描述
1	Server port	Port Number at server
2	Clinet port	Port Number at client
3	Max_data_wire	Maximum of bytes in (Ethernet) packet
4	Max_data_ip	Maximum of total bytes in IP packet
5	Mean_data_control	Mean of control bytes in packet
6	Min_segmn_size_b a	The minimum segment size observed during the lifetime of the connection. (server - client)
7	Initial_window_bytes_a b	The total number of bytes sent in the initial window(client_server)
8	Mean_data_control_a b	Mean of control bytes in packet(client_server)
9	Max_data_ip_b a	Maximum of total bytes in IP packet(server-client)
10	Mean_data_control_b a	Mean of control bytes in packet(server-client)
11	Max_data_control_b a	Maximum of control bytes packet(server-client)

3.2 PSOSC 算法实验

采用流准确率作为 PSOSC 算法实验的评价标准.

流准确率 = 正确标记的流数目 / 总的标记流数目. 准确率计算不包括未标记的流数目.

(1) 流准确率测试

首先分析在标记流数量一定的条件下, 流准确率随着未标记流样本数目变化的情况, 对 Andrew Moore 数据集 entry1 进行了测试.

PSOSC 算法的初始类数 $K = 200$, 粒子数 $m = 30$, $c_1 = 2$, $c_2 = 2$, $\omega = 0.75$, 最大迭代次数为 300, 采用 3.1 节中的 11 种特征.

从实验可以看出在标记流样本一定的情况下, 增加未标记流样本数, 算法流准确率可以提高, 这个

结果在实际中很有价值. 因为与获得标记流样本或者错误标记所付出的代价比较, 未标记流样本是很容易获得的, 因此, 可以通过增加未标记的流数来提高学习算法的流准确率^[4].

实验中还可以看出即使在很少的标识流样本的条件下, 仍然可以获得较高的流准确率, 随着标识流样本数的增加, 流准确率增长缓慢, 但这时流未标记率显著下降.

(2) K 值的选择

聚类类别 K 对算法的准确率和算法开销时间有着重要的影响, 但是没有合适的办法来确定, 试探性地选取不同的 K 和标记流数目进行了实验. 图 1 给出了 K 从 40 到 800, 标记流样本从 500 到 10 000 时的流准确率.

量.

4 结束语

在云计算环境中,资源的管理与调度是关系到计算任务执行效率和服务质量的关键问题,也是从理论研究向实际应用所必需跨过的一步.目前,基于扩展 MVVM 模式的 SaaS 面向服务计算模型的研究主要集中在原型阶段,通过理论研究以及初步的测试,验证了该模型的可行性.应用该模型可以有效地解决在 SaaS 平台下对资源服务质量进行评估处理时所出现的对压力超负荷问题,提高了 SaaS 平台的服务质量.

参考文献:

- [1] Hand E. Head in the clouds[J]. Nature, 2007(449):963.
- [2] Papazoglou M P. Service-oriented computing: concepts characteristics and directions[C]// Proceedings of the 4th

International Conference on Web Information Systems Engineering. Italy:Rome,2003.

- [3] Papazoglou M P, Georgakopoulos D. Service-oriented computing: introduction[J]. Communications of the ACM, 2003,46(10):24-28.
- [4] Wikipedia. Model view view model[EB/OL]. (2010-05-10)[2010-06-05]. http://en.wikipedia.org/wiki/Model_View_ViewModel.
- [5] 廖军.面向服务的计算(SOC)中服务组合的研究-服务计算中一个关键问题的解决方案[D].成都:电子科技大学,2006.

作者简介:

陈明男,(1949-),教授,博士生导师.研究方向为分布式计算、并行计算、云计算.

李猛坤男,(1979-),博士研究生,讲师.研究方向为分布式计算、云计算、软件工程.

张强男,(1982-),硕士研究生.研究方向为软件工程.

(上接第 26 页)

可以看出:当标记流样本超过 2 500 后,流准确率超过 92%,但继续增加标记样本,流准确率增加缓慢.当 K 增加时,流的准确率也在增加,如图中在 10 000 标记流数据,K=800 的情况下,流的准确率达到 96.8%.但是,在实际中取如此大的 K 值是不合适的,增加了运行时间.增加标记流数据可以提高流的准确率,增加 K 也可以提高流的准确率.因此,在实际的使用中,K 值的选取考虑到时间开销和流准确率的平衡,另一方面尽可能增加标记流样本^[5].

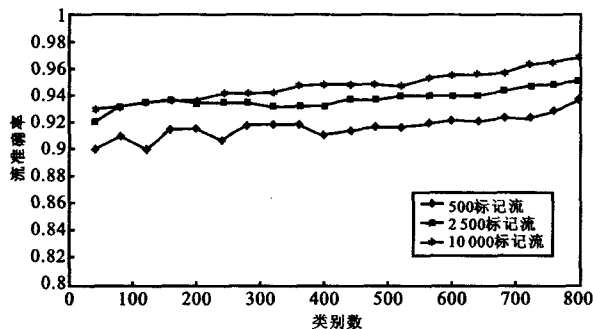


图1 算法准确率

参考文献:

- [1] Auld A W M T, Gull S F. Bayesian neural networks for internet traffic classification[J]. IEEE Transactions on Neural Networks, 2007(18):223-239.
- [2] Zander S, Nguyen T T T, Armitage G. Automated traffic classification and application identification using machine learning[C]// Proceedings of IEEE LCN. Australia, 2005.
- [3] Moore A W, Zuev D. Discriminators for use in flow-based classification[R]. Cambridge: Intel Research, 2005.
- [4] 俞研,黄皓.一种半聚类的异常入侵检测算法[J].计算机应用,2006,26(7):64-66.
- [5] Manikantan Ramadas. TCP TRACE manual[EB/OL]. [2009-12-19]. <http://www.tcptrace.org>

作者简介:

柳斌男,(1971-),博士,讲师.研究方向为网络管理、网络安全.