一种能够适应概念漂移变化的数据流分类方法

富春岩,葛茂松
(佳木斯大学 公共计算机教研部,黑龙江 佳木斯 154007)

摘 要:目前多数的数据流分类方法都是基于数据稳定分布这一假设,忽略了真实数据在一段时间内会发生潜在概念性的变化,这可能会降低分类模型的预测精度.针对数据流的特性,提出一种能够识别并适应概念漂移发生的在线分类算法,实验表明它能根据目前概念漂移的状况,自动地调整训练窗口和模型重建期间新样本的个数.

关键词:数据流;分类;概念漂移;在线学习;决策树
中图分类号:TP311.13 **文献标识码**:A **文章编号**:1673-4785(2007)04-0086-06

A data stream classification methods adaptive to concept drift

FU Chun-yan, GE Mao-song
(Commonality Teaching Department of Computer, Jiamusi University, Jiamusi 154007, China)

Abstract: At present, most classification methods for data streams are developed with the assumption of steady data distribution. However, the data collected from the real world will change over a period of time in the underlying concepts (known as concept drifting). This lowers the predictive precision of a classification model. This paper proposes a classification algorithm that can identify and adapt to occurrences of concept drifting according to the characteristics of the data stream. Experiments show that the proposed algorithm dynamically adjusts the size of the training window and the number of new examples during model reconstruction according to the current rate of concept drifting.

Keywords: data streams; classification; concept drifting; online learning; decision tree

数据流分类是数据流挖掘领域中一个非常重要的问题,其目标是利用训练数据集建立一个分类预测模型,然后利用该模型对新的数据进行分类预测,分类决策的应用领域很多,如客户分类、气候预测、信用风险评估等.

适用于静态数据集的 ID3[1]、C4.5[2] 和 CART[3]等传统分类方法需要存储和处理全部训练样本,而对于动态的潜在无限的数据流,不断增加的训练数据量和待分类数据量为上述分类方法提出了更高的需求,连续的数据流可能超出内存容量,也可能导致计算时间过长.即使所有有效的样本可以被系统处理,由于数据生成过程中不可估计的变化(例如政治事件对股票价格的影响),从历史数据中发现的分类模式对于几小时甚至几分钟之后接到的数据通常是无用的.因此与静态数据分类处理不同,数据流分类挖掘必须先解决概念漂移(drifting concept)问题.

数据流分类为传统的分类技术提出了许多新的挑战,由于分类理论和方法在不同领域有着相当广泛的应用,因此研究快速、精确、稳定的数据流分类系统具有极高的理论价值和应用价值.

本文以适合数据流分类的决策树分类算法为核心,提出一种在线数据流分类方法,在每个时间点,系统使用当前的分类模式为下一个到达的样本预测一个正确的分类,系统能自动地检测概念漂移并重建分类模式以保证预测的精确度.

1 相关工作

Helmbold 与 long 首先研究了概念随时间变化的学习概念漂移问题,漂移的速率被定义为对于 2

收稿日期:2007-03-20.

个连续的样本目标函数不一致的概率[4]. 他们提出一种在固定数据集上最小化这种不一致的算法,其复杂度为样本数量的多项式规模. 然而,对于海量非固定的数据流,漂移发生的实际速率事先无法知道,其算法的运行时间可能不确定地增加.

数据流社区中有关数据流的分类问题研究比较活跃,近几年出现了许多研究成果. Wang 等提出一个通用框架用于挖掘概念漂移数据流[5], Ganti 等开发了一个在插入和删除数据记录时维护模型的算法[6], Widmer 和 Kubat[7] 提出了一簇纯增量算法来处理概念漂移, Domingos 等开发了 VFDT[8], Papadimitriou 等提出 AWSOM (arbitrary window stream modeling method) 用于在传感器网络中发现感兴趣的模式[9], Aggarwal 等采用 On-Demand 分类中 CluStream 的微簇思想,获得了很高的分类精度[10], Last 提出可以适应概念漂移的在线分类系统[11], Ding 等开发了基于 Peano Count Tree 数据结构的决策树[12], Gaber 等开发了 Lightweight 轻权值分类 LWClass 模型[13].

上述已存在的数据流分类模型和算法存在 2 方面问题,首先,未能有效地解决数据流在线训练、测试、分类的速度问题,对于大规模高维数据流的在线分类,目前还没有公认的解决方案;其次,都没有较好地解决变化数据流上的概念漂移问题,未能有效地解决当概念漂移发生时,分类模式快速转变的问题,分类精度偏低.

2　StreamClassifier 分类算法

决策树学习是一种逼近离散值目标函数的有指导的学习方法,在这种方法中学习到的函数被表示为一棵决策树. 学习得到的决策树也能被表示为多个 if-then 规则,以提高可读性. 这种学习算法是最流行的归纳推理算法之一,已经被成功地应用到从学习医疗诊断到学习评估信贷申请的信用风险等广阔领域.

ID3、C4.5 算法仅适用于小规模数据集,在大规模数据的数据挖掘中具有可伸缩性的决策树算法包括 SLIQ[14]、SPRINT[15]、SLIQ 和 SPRINT 算法使用预排序技术以及新的数据结构,大大提高了对于大规模数据集的可扩展性. 其中 SPRINT 算法还非常容易并行化. 但是,这些常见的决策树算法无法处理连续的数据流,并且不具备增量学习能力,使其应用受到了较大限制. 鉴于 SPRINT 算法在可伸缩性、并行性方面的优点,在 SPRINT 算法的基础上构建了数据流分类算法 StreamClassifier.

算法反复从滑动的窗口获取最近的样本来重建分类模型,使用最新的模型为下次模型重建期间的样本进行分类,并暂存最新的分类结果以便为下次模型重建提供训练样本. 当分类误差相对与可以接受的训练误差出现大幅的上升时,意味着检测到概念漂移. 当概念稳定时,系统不断增加训练窗口和分类窗口的大小直到一个预定的上限. 当出现概念漂移时,训练窗口和分类窗口被重置,并提供正确的分类标签用于下次训练.

本文提出数据流分类算法 StreamClassifier 的基本思想如图 1 所示,在连续不断的样本数据流中,用间隔 $[t_0, t_2]$ 中的 T_0 个训练样本产生的模式,对间隔 $[t_2, t_3]$ 中 V_0 个验证样本进行分类. 训练窗口中的样本数与待测试窗口中的样本数不必相等. 在时刻 t_3,学习模块使用训练间隔 $[t_1, t_3]$ 中的 T_1 个样本,对网络进行重建,然后对测试窗口 $[t_3, t_4]$ 中的 V_1 个样本进行分类. 假设在间隔 $[t_3, t_4]$ 中的第 1 个样本在分类模式已经重建完成后才到达.

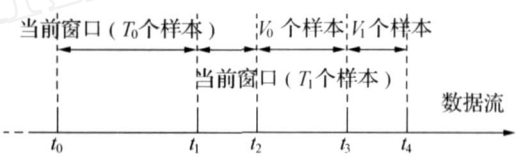

图 1　StreamClassifier 算法训练分类工作原理

Fig. 1　Principle of training and classifying of StreamClassifier algorithm

算法运行需要计算下列参数:训练窗口的尺寸,待分类窗口的尺寸,训练误差和分类误差之间的最大差异.

2.1　计算分类窗口中的样本个数

待分类窗口中的样本个数即 2 次分类模型重建的间隙,为了能够自适应概念漂移的发生,采用概念漂移发生的频率直接决定待分类窗口中的样本个数的思想. 为了检测概念是否发生了偏移,采用纯粹增量的方法对于海量数据流是不合适的,因为在连续的样本实例抵达的间隙,概念通常并不会剧烈频繁地变化,对此 Hulten 等提出的 CVFDT 算法仅在固定个数的样本(20 000)之后才检测一次漂移是否发生. 本文参照 OLIN[11] 里采用的一种启发规则,动态调整 2 次模型重建的间隙之间的样本的个数:如果概念呈现稳定态势,为当前的模型保留更多的样本;如果检测到概念漂移的发生,就大幅度减小待分类窗口的尺寸.

2.2 概念漂移的检测

如果概念是稳定的,训练窗口和紧随其后的分类间隔中的样本应该具有同样的分布.因此分类模型不会在训练和分类误差率之间存在统计意义上重要的差异.从已有的静态数据集上的研究得知,与此相反,误差比率的剧烈增加暗示着发生了概念漂移[7].应用正态分布近似二项分布,根据下列公式计算2个误差率之差[6]:

$$V = \frac{E_{tr}(1 - E_{tr})}{W} + \frac{E_{val}(1 - E_{val})}{C}. \quad (1)$$

式中:W 表示训练窗口大小,C 表示分类窗口大小,E_{tr} 表示训练误差,E_{val} 表示分类误差,V 表示2个误差率之差.

如果概念是稳定的,2种误差率之间在99%置信度级别的最大差异为

$$D_{max} = Z_{0.99}\sqrt{V} = 2.326\sqrt{V}. \quad (2)$$

如果2种误差率之间最大差异超过 D_{max},表示检测到了概念漂移,训练窗口回复到初始值.同时,下一个待分类间隔以百分之 C_{red} 缩减,否则表示概念是稳定的,训练窗口和待分类间隔均增加到最大尺寸.

单纯由式(2)来检测概念漂移存在一定的问题,需要加以修正.

1) D_{max} 反映的是2种误差之间存在的差异,如果训练误差很大,分类误差也很大,那么训练误差和分类误差之间的差异可能小于 D_{max},这时系统没有检测到概念漂移,但是由于分类误差率可能远远超过了能接受的值,需要重建模型.因此需要设置一个用户允许的分类误差率 E_user.当分类误差率超过 E_user 时,认为产生了概念漂移.

2) 令训练误差 $E_{tr} = 0$,若不发生概念漂移,则由 $E_{val} < D_{max}$,可得 $E_{val} < 5.4/(5.4 + C)$,当 C 较大时,E_{val} 接近0.对于分类误差率,一般不可能达到这样的精确度,这样系统会检测到概念漂移.这时认为,只要 $E_{val} <$ E_user,则认为没有产生概念漂移.

2.3 StreamClassifier 分类算法描述

SPRINT 算法是 JohnShafer 和 RakeshAgrawal 于1996年提出的针对大型数据库的一种高速可伸缩的数据挖掘分类算法.其建树过程如下:1)为每个属性建立属性列表;2)对数值属性列表进行排序;3)不断分割节点生成决策树.SPRINT 采用基尼指数(Gini index)来度量最佳分裂点,进行属性选择.Gini 值越小,表明信息增益量(information gain)越大,节点分裂质量越好.为便于描述,涉及的参数的含义如表1所示.

表1 StreamClassifier 算法中的参数含义

Table 1 Parameters meanings in StreamClassifier algorithm

参数	含义
S	训练样本集
A	候选输入属性集(包括离散和连续)
E_user	用户设定的最大分类误差率
W	训练窗口中样本的个数
n_{min}	将被系统分类的第1个样本号(已经到达 i_{min} - 1个样本)
n_{max}	将被分类的最后一个样本号
C_{init}	最开始模型的待分类样本个数
C_{inc}	模型稳定时,模型重建期间样本增加的百分比
C_{max}	模型重建期间运行的最大样本个数
C_{red}	检测到概念漂移时,模型重建期间样本减少的百分比
C_{min}	模型重建期间运行的最小样本个数
W_{max}	训练窗口中允许样本的最大个数

确定了计算训练窗口大小和分类窗口大小的方法以及检测概念漂移是否发生的方法之后,可以构造完整的 StreamClassifier 算法,算法首先执行一些初始化的工作,然后处理连续数据流中输入样本,算法启动时第1批训练样本的分类标签需要显示地指明.算法的伪代码为

Procedure StreamClassifier

Inputs: S, A, C_{init}, C_{inc}, C_{max}, C_{red}, C_{min}, W_{max}

Output: SDT (Stream Decision Tree)

计算训练窗口的初始训练窗口尺寸 W_{init}

训练窗口尺寸 $W = W_{init}$

初始化第1个训练样本的索引 i 为 $n_{min} - W$

初始化最后一个训练样本的索引 j 为 W

初始化分类样本的个数 C 为 C_{init}

while $j < n_{max}$ do begin

对最近 W 个训练样本使用 BuildTree(AtrList, W, N)算法得到分类模式 SDT

计算得到模式的训练误差率 E_{tr}

计算最后需要分类样本的索引 $k = j + C$

计算在 C 个待分类样本上得到模式的分类误差率 E_{val}

更新最近训练样本的索引 $j = k$

确定训练误差和分类误差之间的最大差异 D_{max}

if $(E_{val} - E_{tr}) < D_{max}$ && $E_{val} <$ E_user then // 概念不变

$C = \mathrm{Min}(C^{*}(1 + (C_{inc}/100)), C_{max})$

$W = \mathrm{Min}(W + C, W_{max})$

训练集使用最近分类样本获得的类标签

```
else
    if  E_tr = 0  && E_val < E_user   // 概念不变
    C = Min(C* (1 + (C_inc/100)), C_max)
    W = Min(W + C, W_max)
    训练集使用最近分类样本获得的类标签
    else   // 检测到概念漂移
      重新计算训练窗口的尺寸 W
      训练集采用正确的标签
    C = Max(C* (1 - (C_red/100)), C_min)
    end if
      end if
        返回当前分类模式 SDT
    end do
```

其中决策树的构建采用改进的 SPRINT 分类算法. SPRINT 算法中,当分割属性和分割点确定后,算法开始对属性列表进行分裂,使其以分裂值为分割点分配到左右子节点. 这样要每次为新节点创建一个属性列表集. 而每个节点都维护一个属性列表集需付出较大的系统开销. 为此本文提出的方法是令所有的节点共用一个属性列表集. 将属性列表的记录格式 < 属性值,类标号,样本号 > 中添加一个字段 leaf,用于表明当前记录属于哪个节点. 初始时,所有记录的 leaf 都为 0,即都属于根节点. 当节点分裂时,不用再分裂属性列表,而是将相应记录的 leaf 字段修改,使其隶属于新节点.

```
    Procedure  BuildTree(AtrList, W, N)
if  N 纯  then
    标记为叶节点,标记类属性
    else
    for  N 的每一个分割点 F 生成属性直方图,计算该节点 F 上的基尼指数
    end for
```

选出最佳分割点 F^*,以计算得出的分裂值为分界生成当前节点的左右子节点 $N1$, $N2$. 在该分割点属性列表分割的同时,用该表的 tid 生成记录所属节点的哈希表,并用哈希表分裂其他的属性列表.

```
      BuildTree(AtrList, W, N1)
//  对左右节点递归调用 BuildTree
    BuildTree(AtrList, W, N2)
      end if
```

基于 StreamClassifier 算法构建的分类系统包括采用客户—服务器模式,客户端作为数据采集器. 服务器含 3 个模块:学习模块:采用 StreamClassifier 分类算法构建决策树;分类模块:使用学习模块生成的决策树对样本进行分类;控制模块:控制学习模块和分类模块,根据训练精度和分类精度检测概念漂移,动态调整训练和分类窗口大小.

3　性能评价

实验数据集来自 http://www.ics.uci.edu/～mlearn/ML Repository.html. 数据集是一个葡萄酒化学成分的分析,记录总共 178 条,每条记录包括 13 个连续属性和 1 个类标签,整个数据集有 3 个类别. 系统配置为 Celeron 2.4 GHz/256 MB 内存.

为了模拟数据流,采用 Java 的面向连接的 Socket 通信模式,客户端作为数据发生器,服务器端对接收的样本缓存并执行训练和分类算法. Socket 服务端作为主程序的一个线程,在后台负责监听客户端连接,建立连接后,利用 while 循环不断接收来自客户端的数据,并存储在缓冲区中.

实验 1　不同分类策略的比较:

在线分类系统的不变参数设定为: E_user = 0.4, n_{max} = 178, C_{init} = 20 C_{inc} = 0.5, C_{max} = 50, C_{red} = 0.75, C_{min} = 5, W_{max} = 100. 在表 2 中,将在线分类系统的性能与其他的分类方法做了对比(客户端数据发送间隔 1 ms). 测试 5 和测试 6 是在线分类系统运行结果. 测试 1 和测试 2 显示了模型不重建的方法. 即分别假设到达 50 和 78 个训练样本,启动算法生成决策树,对后面陆续到达的样本进行分类,期间决策树不再重建. 测试 1,2 显示了静态分类的思想,可以看到训练窗口大小对分类精度的影响. 如果模式发生变化,用较早的分类模式对较晚到达的数据进行分类,容易产生更大的误差. 相对于测试 1 来说,测试 2 的训练样本更多更新,因此具有更高的分类精度. 相对于模型不重建的方法,在线分类系统的优势是明显的. 测试 3 和 4 显示了静态窗口分类的结果. 根据实验结果得知,在线分类系统具有更高的精确度,并且运行时间更少,但静态窗口使用的内存空间要小一些.

表 2　StreamClassifier 在不同分类模式的性能对比

Table 2　Performance comparison of StreamClassifier in different classification mode

测试编号	初始化窗口	分类样本个数	窗口个数	平均错误率	运行时间/s
1	50	128	1	0.469	1.900
2	78	100	1	0.260	1.500
3	40	10	13	0.292	2.750
4	50	20	6	0.325	2.770
5	30	动态	9	0.278	2.231
6	50	动态	14	0.290	2.015

实验2 检测算法对概念漂移的适应情况.

图2显示了每次训练分类的训练误差和分类误差,图3显示了对应的训练窗口和分类窗口.算法刚开始时采用正确的类标签进行训练,因此训练误差为0.这时概念稳定,分类训练窗口和分类窗口不断扩大,分类误差也不断上升,第3次分类结束后检测到概念漂移,模型采取相应调整措施,训练窗口恢复到初始窗口,大幅度缩小分类窗口的大小.采用正确的类标签进行训练,训练误差率回到0.然后概念恢复稳定,在第5次分类时再次检测到概念漂移,第6次调整恢复正常.因此6、7、8次分类和1、2、3次分类的窗口和错误率发展情况类似.第8次的分类窗口本应大于第7次,但由于数据流接近末尾,提供的分类样本数量比较少,分类完后算法就结束了,所以出现图示的情况.

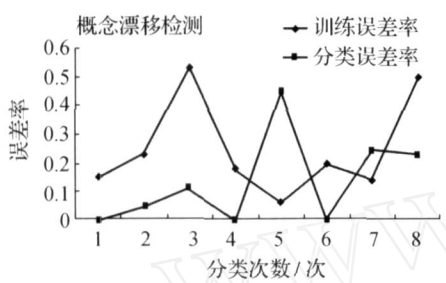

图2 分类的训练误差率与分类误差率

Fig.2 Error rate training and classifying error rate

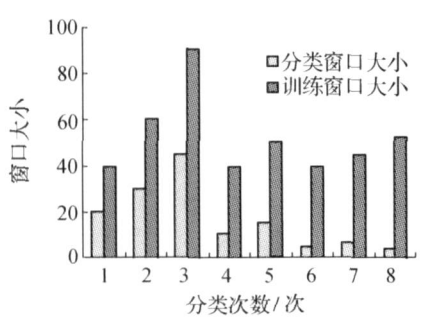

图3 训练窗口和分类窗口变化情况

ig.3 The change of training and classifying window

Java定时器的精度只能达到1 ms,可以看到系统有个运行的极小时间2.5 ms,当发送间隔在10 ms以内时发送速度对运行时间不构成影响.当发送间隔1 ms时,发送178个数据需要0.178 s,但系统运行2.5 s左右,其主要时间开销来自缓冲区的读写.

表3 系统运行时间表

Table 3 Sending interval Vs. running time s

客户端发送间隔	运行时间
1.000	158.0
0.500	79.0
0.100	17.3
0.050	9.9
0.010	2.5
0.001	2.5

从表3中的数据测试发现,对50个样本一次建树时间63 ms,对128个样本分类时间为1.83 s,建树时间约为分类时间的3%左右.分类的时间开销主要来自于读缓冲区,因为缓冲区使用了并发控制,客户端不断将样本写进缓冲区,导致分类时从缓冲区读取样本可能等待,所以耗费了大量时间.从发送间隔0.1 s开始,系统运行时间和样本的总传输时间相当.

4 结束语

在许多数据流应用中,短时间内有大量数据连续到达,这些数据具有随时间动态变化的趋势,往往又是高维的,怎样使用有限存储空间对这些数据流进行快速处理以获取有用信息,为数据挖掘及其应用研究带来了新的机遇和挑战.本文针对数据流的特性,提出了一个能够适应概念漂移发生的在线分类算法,实验证明基于此算法构建的在线分类系统可以对连续变化的数据流进行分类,耗用较少的资源,并具有较高的分类精度.目前的工作尚不完善,还需要运用更多的实时数据集来测量系统的性能,可以探究其他的方法检测概念漂移的发生.

参考文献:

[1]QUINLAN J R. Induction on decision trees[J]. Machine Learning,1986,13(1):81-106.

[2]QUINLAN J R. C4.5:programs for machine learning [M]. San Mateo:Morgan Kaufmann,1993.

[3]BREIMAN L,FRIEDMAN J,OLSHEN R,et al. Classification and regression trees monterey[M]. Belmont:Wadsworth International Group, 1984.

[4]HELMBOLD D P, LONG P M. Tracking drifting concepts by minimizing disagreements[J]. Machine Learning, 1994,21(14):27-45.

[5]WANG H, FAN W, YU P, HAN J. Mining concept-drifting data streams using ensemble classifiers[A]. The 9th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD's03)[C]. New York:

ACM Press, 2003.
[6] GANTI V, GEHRKE J, RAMAKRISHNAN R. Mining data streams under block evolution[A]. SIGKDD's02 [C]. New York: ACM Press, 2002.
[7] WIDMER G, KUBAT M. Learning in the presence of concept drift and hidden contexts[J]. Machine Learning, 1996, 23(1): 69 - 101.
[8] DOMINGOS P, HULTEN G. Mining high-speed data streams[A]. In Proceedings of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining[C]. New York: ACM Press, 2000.
[9] PAPADIMITRIOU S, FALOUTSOS C, BROCKWELL A. Adaptive, hands-off stream mining[A]. Proceedings of the 29th International Conference on Very Large Data Bases (VLDB's03)[C]. Berlin: Springer Press, 2003.
[10] AGGARWAL C, HAN J, WANG J, YU P S. On demand classification of data streams[A]. Proc 2004 Int Conf on Knowledge Discovery and Data Mining[C]. New York: ACM Press, 2004.
[11] LAST M. Online classification of nonstationary data streams[J]. Intelligent Data Analysis, 2002, 6(2): 129 - 147.
[12] DING Q, DING Q, PERRIZO W. Decision tree classification of spatial data streams using peano count trees [A]. Proceedings of the ACM Symposium on Applied Computing[C]. New York: ACM Press, 2002.
[13] GABER M M, KRISHNASWAMY S, ZASLAVSKY A. On-board mining of data streams in sensor networks [M]. Springer Verlag, 2003
[14] MEHTA M, AGRAWAL R, RISSANEN J. SLIQ: A fast scalable classifier for data mining[A]. Proc 1996 Int Conf Extending Database Technology (EDBT's96)[C]. Springer Press, 1996.
[15] SHAFER J, AGRAWAL R, MEHTA M. SPRINT: A fast scalable parallel classifier for data mining[A]. Proc 1996 Int Conf Very Large Data Bases (VLDB's96)[C]. Springer Press, 1996.
[16] MITCHELL T M. Machine learning[M]. New York: McGraw Hill, 1997.

作者简介:

富春岩,女,1974 年生,讲师,主要研究方向为现代数据管理技术、数据流、海量数据处理.

E-mail: jmsfu@126.com.

葛茂松,男,1971 年生,高级实验师,硕士研究生,主要研究方向为数据挖掘、数据流、海量数据处理.

《机器人技术与应用》杂志征订启事

《机器人技术与应用》是由国家 863 计划机器人技术主题专家组和北方科技信息研究所联合主办,创刊于 1988 年,是中国学术期刊(光盘版)与《中国期刊网》全文收录期刊,是机器人行业唯一综合性技术刊物,在国内自动化领域享有很高的声誉. 本刊为国际机器人联合会(IFR)会员单位.

《机器人技术与应用》主要报道工业自动化、智能化工程机械及零部件、数控机床、机器人技术领域所取得的新技术、新成果、科技动态与信息. 传播企业信息和市场行情,交流业内创新成果,推动行业技术进步. 本刊涵盖面广,集知识性与趣味性于一体,具有很强的技术性和可读性. 读者对象主要是从事自动化、汽车、电子、机械、航天航空等行业的广大管理人员、技术人员、销售人员以及研究院所师生和机器人爱好者.

《机器人技术与应用》为双月刊,逢单月月底出版,大 16 开本,正文 48 页,每期定价 10 元,全年 60 元.

联系方式:

地址:北京市海淀区车道沟 10 号科研一号楼 1403 室

通讯地址:北京市 2413 信箱 41 分箱

邮编:100089

电话(传真):(010) 68961813

网址:www.rta.org.cn

E-mail: robot@onet.com.cn