

一种自适应局部概念漂移的数据流分类算法

尹志武 黄上腾

(上海交通大学计算机科学与工程系 上海 200240)

摘要 本文基于 DB2 算法提出一个能实时检测局部概念漂移,并随之自适应调整的数据流分类算法 IncreDB2。该算法动态增量维护一个层次分类树。当局部概念漂移出现时,IncreDB2 不是重新构造一个全新的分类树,而是仅更新漂移所影响到的局部结点,具有较高的时间效率。实验结果表明了该算法的正确性和有效性。

关键词 数据流挖掘,多分类,局部概念漂移

Adaptive Method for Handling Local Concept Drift of Data Streams Classification

YIN Zhi-Wu HUANG Shang-Teng

(Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240)

Abstract Based on the DB2 method, an adaptive method called IncreDB2 is proposed to detect and adapt to local concept drift continuously in data stream classification. This method dynamically maintains a hierarchical classification tree. When local concept drift is detected, IncreDB2 only updates the nodes that affected by this drift rather than rebuilds a new classification tree from scratch, which means that it has better time efficiency. Experimental results demonstrated the validity and efficiency of this method.

Keywords Data stream mining, Multi-classification, Local concept drift

1 引言

作为一种新的数据形式,数据流^[1-3]在入侵检测、传感器网络、股票分析等许多领域有着广泛的应用。数据流不是静态有限数据的集合,而是大量不断流入数据的集合,这对模式分类^[4-5]问题提出新的挑战;数据流分类模式的一部分会随着时间不断变化,而另一部分对数据的分类仍然有效,这称之为局部概念漂移。在这种情况下,分类模式的改变仅仅影响局部而不是整个的实例空间。因此,检测分类模式的局部变化从而依此来调整分类器对数据流分类来说更有效率。

最近有一些算法被提出来解决数据流分类中出现的局部概念漂移问题。Gama 等^[4]提出增量维护一个决策树分类模型,在决策树各个内部结点上跟踪检测概念漂移的产生。在概念漂移出现时,仅仅更新决策树模型中被漂移所影响到的结点。这种算法的不足是对高维数据分类的时间效率低。Tsymbal 等^[5]提出在实例层次上集成若干基本分类器,依据对所测试实例分类准确率为每个分类器分配成比例的权重。这种算法的不足是多个分类器集成造成分类结构的不够清晰。

DB2 算法^[6]是一种对高维数据有较好时间效率和分类准确率的层次式多类分类算法。基于 DB2 算法,本文提出一种新的能处理局部概念漂移的数据流分类算法 IncreDB2。该算法初始构建一个 DB2 层次分类树,然后持续判断子结点上分类器对不断变化的数据流的分类准确率。如果部分模式发生改变或者出现新的模式时,IncreDB2 不是重新构建一个新的层次分类树,而是更新调整概念漂移所影响到的局部分类结点,使得该算法具有较好的时间效率和适应性。

2 DB2 算法分析

Volkan 等提出了 DB2 层次式多分类支持向量机的算法。训练期间,DB2 将共 N 类的数据逐层划分为两个子集,同类数据属于同一子集,直至每个子集包含一类数据,共构建 N-1 个二分类器。划分过程如下步骤进行。

(1)使用 k-means 聚类算法,将训练数据划分为两个子集 A 和 B。

(2)使用子集 A 和 B 的数据训练 SVM,得到一个二分类支持向量机。

(3)重复上述的两个步骤,直到所有的子集中只含有一个类。

DB2 使用训练得到的层次分类树对未知类别数据进行分类。所有类开始作为测试数据真实类的候选。从根结点开始,使用该结点的 SVM 对测试数据进行分类,将那些不属于该数据子集的数据类别从候选类中消去,接着处理数据所属的下一分支结点,重复此过程直到候选类中只剩一个类,将此类作为该实例的预测类别。DB2 算法与一对多算法^[7]和一对一算法^[8]有相似的分类准确率和训练速度,但是有更快的分类速度。

3 自适应局部概念漂移的数据流分类算法

基于 DB2 算法,本文提出一个自适应局部概念漂移的数据流多类分类算法 IncreDB2。该算法初始构建一个 DB2 层次分类树,然后持续地跟踪各结点分类器对最新数据窗口内的流数据分类的准确率,并以此对层次分类树进行更新、添加和删除等操作,使得分类器能够在数据流的概念漂移出现的时候进行自适应的调整。

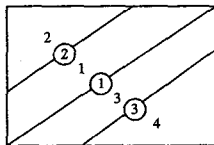


图1 初始数据分布图

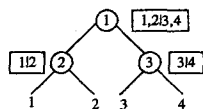


图2 初始层次分类树

在最新数据窗口内既包含已处理类别(当前层次树中所包含类)数据,也包含尚未处理类别(当前层次树中尚未包含类)数据。对某个已处理类别的数据进行测试分类时,如果结点分类器的准确率低于事先设定的阈值,IncreDB2 使用最新窗口内该类数据对该结点为根结点的分类子树进行重新构建。如果最新窗口出现尚未处理类别的数据时,则从根结点开始更新与此类数据相关的各个结点分类器信息,处理后的层次分类树结点会增加一个。另外,IncreDB2 记录各个类最近出现窗口的时间。为了节省空间和提高时间效率,如果某个类的数据很久没有出现,则从层次树中删除与此类数据相关的信息。处理后的层次分类树结点会减少一个。

IncreDB2 的具体实现步骤如下:

(1)初始化构建一个 DB2 层次分类树。如图 1,2 所示,初始的 4 类数据生成具有 3 个子结点和 4 个叶结点的初始层次分类树。

(2)收集一个窗口的数据并将这些数据分类存储,记录类别信息,更新各类的生存期。在最新窗口内出现的数据类别生存期设置为 1,没有出现的数据类别生存期加 1。

(3)对于最新窗口内已处理类别的各类数据,分别进行以下的更新处理:从根结点开始,测试层次树中与此类别相关的各个结点分类准确率。如果某结点的准确率小于预先设定的结点更新阈值,则使用最新的该类数据重新构建以此结点为根结点的子树。否则,不进行层次树更新。如图 3 和 4 所示,类 3 与类 4 之间的分类模式发生改变,这时候只需要使用最新的类 3 和类 4 的数据重新生成子结点 3,而结点 1 和 2 的分类器都保持不变而继续使用。

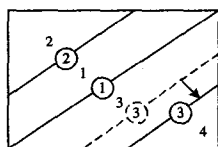


图3 已有模式变化后数据分布图

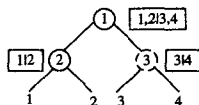


图4 已有模式变化后层次分类树

(4)对于最新窗口内尚未处理类别的各类数据,分别进行以下处理:从根结点开始,使用当前层次树上各结点分类器对该类数据进行分类,统计各结点的分类结果。如果该类数据划分到该结点某一支结点概率超过某设定的结点添加阈值,添加相应的该类别信息到决策函数,沿着该分支重复上述过程,直到该类数据和叶结点划分同组,以此两类数据构建一个新的支持向量机。否则,使用该类数据重新构建以此结点为根结点的分类子树。例如,如图 5 和 6 所示,在最新数据窗口中有新类 5 数据出现时,结点 2 的分类器保持不变继续

使用,结点 1 和 3 的分类器只需要添加类 5 相关信息而不需要重构,结点 4 的分类器使用最新的类 4 和 5 的数据生成并添加到当前的层次分类树上。

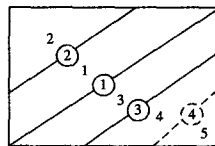


图5 新模式出现后数据分布图

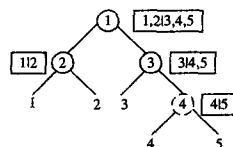


图6 新模式出现后层次分类树

(5)检查层次树中各个使用类别的生存期。对超过最大生存期的每一个类别作以下处理:从根结点开始,删除关于该类的信息。如果层次树中只存在一个类,则层次树为空。如果该类的叶结点的父结点是根结点,则将根结点的另一子结点提升为根结点。如果该类的叶结点的兄弟结点是一个子结点,则将该兄弟叶结点提到父结点的位置,类型为子结点。如果该类的叶结点的兄弟结点也是一个叶结点,则将该兄弟叶结点提到父结点的位置,类型为叶结点。如图 7 和 8 所示,类 4 超过最大生存期,兄弟结点类 3 是叶结点,这时类 3 结点提到类 4 的父结点位置,类型仍然是叶结点。

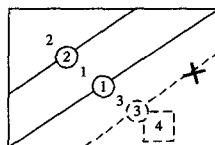


图7 已有模式过时后数据分布图

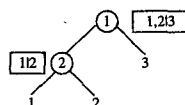


图8 已有模式过时后层次分类树

(6)对于持续到来的窗口数据重复上述步骤(2)~(5),直到所有的数据处理完成。

IncreDB2 不断跟踪测试层次分类树对最新数据的分类准确率,并进行自适应调整增量更新分类树,因此可以解决流数据的概念漂移问题。而且,IncreDB2 仅仅调整更新那些概念漂移所影响到的子结点,而不是重新构建一个新的层次树,具有较高的时间效率。

4 实验结果

本实验对 IncreDB2 算法的分类准确率和时间效率进行评估,然后将结果与传统的 IDB2 算法(当检测到概念漂移时,重新构造一个全新的层次分类树)进行比较。测试环境采用 PC 个人电脑:CPU 是 Intel Pentium 4,内存 512M,操作系统 Windows 2000,算法采用 MATLAB 实现。实验采用 KDD-CUP99 网络入侵检测数据集^[9]。该数据集由一系列

(下转第 143 页)

置的预测提供相关服务。而移动对象预测问题的不确定性是普遍存在并且急需解决的问题,不确定性处理的好还是差直接影响到查询、管理结果的精确度,进而影响到用户需求。其中,空间不确定性中的移动对象预测位置的模糊不确定性必然会累及传统 KNN 得到的 k 近邻的精确度,同时会引起 k 近邻集合基于“ k 近邻”不分明关系的粗糙不确定性。针对这种已往研究没有涉及到的不确定性的处理,本文通过传统方法扩大 k 近邻查询范围,应用模糊-粗糙集理论分析预测得出的 $k+m$ 近邻集合,借助移动对象预测位置的模糊隶属度和模糊-粗糙隶属函数,最终确定所求 k 近邻集合中的各个点。尽管计算量略微增大,但数值分析可以表明本方法很大程度上提高了移动对象 k 近邻预测的精确度。

另外,本方法的模糊-粗糙隶属函数中用到了 $k+m$ 近邻集合 W 中各点到模糊空间 R 中 $2n$ 个移动对象 P 可能位置取样点的距离。这种取样的方法会一定程度上影响精确度。所以,未来的工作将致力于基于非取样方法的移动对象 k 近邻预测的精确度的进一步提高。

参考文献

- Konig-Ries B, Makki K. Research Direction for Developing an Infrastructure for Mobile & Wireless Systems. Consensus Report of the NSF Workshop, Scottsdale, AZ, October 2001
- Chon H D, Agrawal D, Abbadi A E. Data Management for Moving Objects. In: Proceeding of International Telematics and LBS Workshop, 2003
- Trajcevski G. Managing Uncertainty in Moving Objects Databases [J]. ACM Transactions on Database Systems, 2004, 29(3): 463~507
- Agarwal P K, Arge L, Erickson J. Indexing Moving Points. In: Proceedings of PODS, 2000. 175~186
- Patel J M, Chen Y, Chakka V P. STRIPES: An Efficient Index for Predicted Trajectories. In: Proceedings of SIGMOD, 2004.

(上接第 139 页)

TCP 连接记录组成,每个记录有 34 个连续属性,共有 5 个类别。在本实验中,主要参数设置如下:窗口大小 100,结点更新阈值 90%,结点添加阈值 90%,类最大生存期 10。实验中采用共 5 组数据,每组数据为 4M,表 1 显示了 InnoDB2 算法和传统算法的分类准确率和时间效率的平均值。可以看出 InnoDB2 算法和传统算法有相似的分类准确率,但是 InnoDB2 有较好的时间运行效率。

表 1 算法分类准确率和训练时间比较

	分类准确率(%)		训练时间(s)	
	InnoDB2	IDB2	InnoDB2	IDB2
1	98.93	98.00	0.2528	0.3029
2	98.36	98.47	0.5950	0.6394
3	93.61	93.94	0.7491	0.8434
4	96.28	95.34	0.6010	0.6805
5	95.67	95.98	0.4511	0.5416

结束语 本文提出了一个能实时检测局部概念漂移并随之自适应调整的数据流分类算法 InnoDB2。当局部概念漂移出现时,该算法不是重新构造一个全新的 DB2 层次分类树,而是仅更新漂移所影响的局部结点,使得该算法具有较好的时间运行效率。实验结果证实该算法可以较好解决数据流分类中局部概念漂移的问题。

637~646

- Saltenis S, Jensen C S, Leutenegger S T, et al. Indexing the Positions of Continuously Moving Objects. In: Proceedings of SIGMOD, 2000. 331~342
- Procopiu C M, Agarwal P K, Har-Peled S. STAR-TREE: An Efficient Self-Adjusting Index for Moving Objects. In: Proceedings of the Intl Workshop on Algorithm Engineering and Experiments, 2002. 178~193
- Tao Y, Papadias D, Sun J. The TPR*-TREE: An Optimized Spatio-Temporal Access Method for Predictive Queries. In: Proceedings VLDB, 2003. 790~801
- Jensen C S, Lin D, Ooi B C. Query and Update Efficient B+-TREE Based Indexing of Moving Objects. In: Proceedings of VLDB, 2004. 768~779
- Xu B, Wolfson O. Time-Series Prediction with Applications to Traffic and Moving Objects Database [c]. In: The Third International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'03), San Diego, California, 2003. 56~60
- Cheung K L, Fu A W. Enhanced Nearest Neighbour Search on the R-TREE [J]. ACM SIGMOD Record, 1998, 27(3): 16~21
- Roussopoulos N, Kelley S, Vincent F. Nearest Neighbor Queries [C]. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1995. 71~79
- Song Z, Roussopoulos N. K-Nearest Neighbor Search for Moving Query Point [C]. In: Proceedings of the 7th International Symposium on Spatial and Temporal Databases, 2001. 79~96
- Benetis R, Jensen C S, Karciuskas G, et al. Nearest Neighbor and Reverse Nearest Neighbor Queries for Moving Objects [C]. In: Proceedings of the International Data Engineering and Applications Symposium, 2002. 44~53
- Tao Y, Papadias D. Time-parameterized Queries in Spatio-temporal Databases [C]. In: Proceedings of the ACM SIGMOD Conference, 2002. 334~345
- Zadeh L A. Fuzzy Sets. Information and Control, 1965. 338~353
- Pawlak Z. Rough Sets. International Journal of Computer and Information Sciences, 1982. 341~356
- Dubois D, Prade H. Rough fuzzy sets and fuzzy rough sets [J]. International Journal of General Systems, 1990. 191~209

参考文献

- Babcock B, Babu S, Datar M, et al. Models and Issues in Data Stream Systems. In: Proc. of the 21st ACM Symposium on Principles of Database Systems (PODS2002), 2002. 1~16
- 金澈清,钱卫宁,周傲英. 流数据分析与管理综述. 软件学报, 2004, 15(8): 1172~1181
- 孙玉芬,卢炎生. 流数据挖掘综述. 计算机科学, 2007, 34(1): 1~5
- Gama J, Castillo G. Learning with Local Drift Detection. In: ADMA, 2006. 42~55
- Tsymbal A, Pechenizkiy M, Cunningham P, et al. Handling Local Concept Drift with Dynamic Integration of Classifiers: Domain of Antibiotic Resistance in Nosocomial Infections. In: CBMS, 2006. 679~684
- Volkan V, Jennifer G D. A Hierarchical Method for Multi-class Support Vector Machines. In: ICML, 2004. 105~113
- Vapnik V. Statistical Learning Theory. Wiley, New York, 1998
- Kreel U. Pairwise Classification and Support Vector Machines. Advances in Kernel Methods-Support Vector Learning, Cambridge, MIT Press, 1999. 255~268
- Newman D J, Hettich S, Blake C L, et al. UCI Repository of Machine Learning Databases, 1998