GPU高性能计算技术在数据挖掘领域的研究进展

张义飞 201821080630

UESTC — April 11, 2019

Abstract

将计算密度高的部分迁移到 GPU 上是加速经典数据挖掘算法的有效途径。首先介绍 GPU 特性和主要的GPU 编程模型,随后针对数据挖掘主要任务类型分别介绍基于 GPU 加速的工作,包括分类、聚类、关联分析、时序分析和深度学习。

关键字：数据挖掘、高性能计算、GPU加速、并行计算

Transferring the procedure involving dense computation to GPU is known as an effective way to accelerate the whole procedure of many classical data mining algorithms. In this paper, features of GPU as well as existing programming models of GPU are introduced firstly. The representative works of fundamental data mining tasks are covered respectively, including classification, clustering, association analysis, time series analysis and deep learning.

Keywords: Data Mining, High-performance computing, GPU, parallel computing

1 引言

数据挖掘是一个从大量历史数据中发现有趣模式的过程[1]，目前已经应用在商务智能、Web搜索、金融、数字图书馆等领域。数据挖掘对社会具有很大影响，并且未来这种影响将继续。随着信息技术和互联网的快速发展，各领域收集的数据规模已经达到 TB 甚至 PB 量级，这对计算速度提出了严峻挑战。GPU（Graphics Processing Unit）是相对于 CPU 的一个概念，伴随着多媒体计算需求的迅猛增长而被意识到，并最终成为一个独立于传统 CPU 的处理器。

如今，GPU 已经不再局限于 3D 图形处理了，GPU 通用计算技术发展已经引起业界不少的关注，事实也证明在浮点运算、并行计算等部分计算方面，GPU 可以提供数十倍乃至上百倍于 CPU 的性能。相比 CPU 而言，GPU 有以下特点[2-3]：（l）GPU 通常具有更大的内存带宽；（2）GPU 具有更多的执行单元；（3）GPU 较同等级的 CPU 具有价格优势。基于 GPU 高性能和性价比等优势，现在 GPU 的应用已经扩展到了图形领域之外的范围内，如数值天气预报[4]，地质勘探[5]、代数计算[6-7]、分子动力学模拟[8]、数据库操作[9]等。

本文首先介绍 GPU 特性和主流编程模型，随后着重介绍已有的通过 GPU 加速经典数据挖掘算法的工作，包括聚类、分类、关联分析、时序分析以及当下热门的深度学习，并讨论适合于 GPU 加速数据挖掘算法的特征。

2 基于GPU的计算和编程

2.1 GPU

GPU 之所以在某些应用中较 CPU 能够获得更高的性能，主要是因为 GPU 和 CPU 在硬件结构设计上存在很大差异。如图 1 所示[10]，GPU 将大量的晶体管用作ALU 计算单元，从而适应密集且可并行的图像渲染计算处理需要。相对 GPU 而言，CPU 却是将更多的晶体管用作复杂的控制单元和缓存等非计算功能，并以此来提高少量执行单元的执行效率。

此外，存储带宽是另一个重要问题。存储器到处理器的带宽已经成为许多应用程序的瓶颈。目前 GPU 的芯片带宽是 CPU芯片带宽的 6倍左右[11]。

2.2 CPU/GPU协同并行计算

在诸多适用于高性能计算的体系结构中，采用通用多核 CPU 与定制加速协处理器相结合的异构体系结构成为构造千万亿次计算机系统的一种可行途径。而在众多异构混合平台中，基于 CPU/GPU 异构协同的计算平台具有很大的发展潜力。在协同并行计算时，CPU和GPU 应各取所长，即 CPU 承担程序控制，而密集计算交由 GPU 完成。另外，除管理和调度 GPU 计算任务外，CPU也应当承担一部分科学计算任务[12]。

新型异构混合体系结构对大规模并行算法研究提出了新的挑战，迫切需要深入研究与该体系结构相适应的并行算法。事实上，目前基于 GPU 加速的数据挖掘算法实现都有 CPU 参与协同计算，只是讨论的重点多集中在为适应 GPU 而进行的并行化设计上。实践中，需要找出密集计算部分并将其迁移到 GPU 中执行，剩余部分仍然由 CPU来完成。

2.3 CUDA

为了加速GPU通用计算的发展，NVIDIA公司在2007”年推出统一计算设备架构（ComputeUnifiedDeviceArchitecture，CUDA）[10，13]。CUDA编程模型将CPU作为主机，GPU作为协处理器，两者协同工作，各司其职。CPU负责进行逻辑性强的事务处理和串行计算，GPU则专注于执行高度线程化的并行处理任务。CUDA采用单指令多线

程（SIMT）执行模式，而内核函数（kernel）执行GPU上的并行计算任务，是整个程序中一个可以被并行执行的步骤。CUDA计算流程通常包含CPU到GPU数据传递、内核函数执行、GPU到CPU数据传递三个步骤。

CUDA不需要借助于图形学API，并采用了比较容易掌握的类C/C++语言进行开发，为开发人员有效利用GPU的强大性能提供了条件。CUDA被广泛应用于石油勘探、天文计算、流体力学模拟、分子动力学仿真、生物计算和图像处理等领域，在很多应用中获得了几倍、几十倍，乃至上百倍的加速比[13]。

2.4 并行编程语言和模型

过去几十年里，人们相继提出了很多并行编程语言和模型，其中使用最广泛的是为可扩展的集群计算设计的消息传递接口（Message Passing Interface，MPI）和为共享存储器的多处理器系统设计的OpenMP[14]。OpenMP最初是为 CPU执行而设计的。

OpenACC[15]是计算机厂商为异构计算系统提出的一种新编程模型，其主要优势是为抽象掉许多并行编程细节提供了编译自动化和运行时系统支持。这使得应用程序在不同厂商的计算机和同一厂商不同时代的产品中保持兼容性。然而，学习 OpenACC 需要理解所有相关的并行编程细节。

在 MPI编程模型中，集群中的计算节点之间相互不共享存储器；节点之间的数据共享与交互都通过显式传递消息的方式实现。MPI 成功应用于高性能科学计算（HPC）领域。现在很多 HPC 集群采用的是异构的 CPU/GPU 节点。在集群层次上，开发人员使用 MPI 进行编程，但在节点层次上，CUDA 是非常高效的编程接口。由于计算节点之间缺乏共享存储器机制，要把应用程序移植到 MPI中需要做大量针对性分析和分解工作。

包括苹果公司在内的几大公司在 2009 年共同开发了一套标准编程接口，称之为 OpenCL[16]。与 CUDA 类似，OpenCL 编程模型定义了语言扩展和运行时 API，使程序员可以在大规模并行处理中进行并行管理和数据传递。与 CUDA 相比，OpenCL 更多地依赖 API，而不是语言的扩展，这允许厂商快速调整现有编译器和工具来处理 OpenCL 程序。OpenCL 和 CUDA 在关键概念和特性上有诸多相似之处，因此 CUDA程序员可以很快掌握OpenCL。

2.5 MATLAB

因提供丰富的库函数库以及诸多其他研究者贡献和共享的函数库，MATLAB 是研究人员实现算法的常用平台。通过封装的数据容器（GPUArrays）和函数，MATLAB 允许没有底层 CUDA 编程能力的研究人员可以较容易获得 GPU计算能力，因此 MATLAB较 OpenCL更容易上手。截止准备本文时，2014版本的MATLAB提供了 226 个内置的 GPU 版本的库函数。对于有 CUDA编程经验的人员，MATLAB 允许直接集成 CUDA 内核进 MATLAB应用。

2.6 JACKET引擎

JACKET[17]是一个由 AccelerEyes 公司开发专门用于以 MATLAB 为基础的基于 GPU 的计算引擎，其最新版本已经包含了高层的接口，完全屏蔽了底层硬件的复杂性，并支持所有支持 CUDA 的 GPU 计算，降低了进行CUDA 开发的门槛。JACKET 是 MATLAB 代码在 GPU上运行的插件。JACKET 允许标准的 MATLAB 代码能够在任何支持 CUDA 的 GPU 上运行，这使得广大的MATLAB 及 C/C++用户可以直接使用 GPU 强大的计算能力进行相关应用领域的快速原型开发。JACKET 包含了一套运行于 MATLAB 环境中优化并行计算的基础函数库。并且支持 MATLAB 数据类型，可将任何存储于 MATLAB CPU 内存中的变量数据转换为 GPU 上的数据类型，对以往的 MATLAB 程序来说，只需更改数据类型，就能迁移到 GPU 上运行。

3 相关工作综述

3.1 基于CPU的数据挖掘算法实现

数据挖掘算法的研究一直很活跃，许多成熟和经典的算法已经实现在诸多研究或商用软件包/平台，例如开源的Weka[18]和KNIME，以及商用的IBM公司的PASWModeler（即之前SPSS公司的Clementine®）。这些软件默认都是单机版本，可运行在普通 PC 或高性能服务器上，基于 CPU 的计算能力。为了适应目前大规模的计算，出现了基于 Google 公司提出的 MapReduce[19]计算框架实现的开源数据挖掘平台 Mahout[20]。相关的研究起源于斯坦福大学Andrew Ng研究组2006年的经典论著[21]。由于现有的算法需要先找到可“迁移”到 MapReduce 的方式，因此目前 Mahout 平台上仅有几个能支持分布式部署的数据挖掘算法，包括用于分类的朴素贝叶斯、随机森林，用于聚类的 k -Means，基于项目的协同过滤等。目前 Mahout仍然是基于 CPU的计算能力。

3.2 聚类算法

聚类是数据挖掘中用来发现数据分布和隐含模式的一种无监督学习，每个训练元组的类标号是未知的，并且要学习的个数或集合也可能事先不知道。对于给定的数据集，聚类算法按照一定的度量，将数据对象分组为多个簇，使得在同一个簇中的对象之间具有较高的相似度，而不同簇中的对象差别很大[22-23]。$k-Means$算法是经典的基于距离划分的聚类分析算法，也是应用得最广泛的算法之一，采用距离作为相似性的评价指标，即认为两个对象距离越近，其相似度就越大。

计算各个数据对象到各簇中心的欧氏距离和将数据对象分配到最近的簇的时候，数据对象之间都是相互独立的，不需要进行交换，且没有先后顺序，后计算的对象不需要等待前一次计算的结果，仅在完成全部分配过程之后，才需要进行一次数据汇总。所以文献[25]的作者们使用 GPU 并行优化了一维数据的 k -Means算法的步骤 2，并使用带缓存机制的常数存储器保存中心点数据，能获得更好的读

取效率。文献中还展示了实验结果，在 8600GT上取得了 14倍左右的加速效果。

DBSCAN 属于基于密度的聚类算法中最常被引用的，G-DBSCAN 是它的一个 GPU 加速版本[26]。文献[26]的实验显示较 DBSCAN可以实现高达 112倍的加速。

BIRCH 是经典的基于层次的聚类算法，文献[27]中基于 CUDA 实现的 GPU 加速版本在实验中获得了高达154倍的加速。

3.3 分类算法

分类是数据挖掘中应用领域极其广泛的重要技术之一，至今已经提出很多算法。分类算法[28] 是一种监督学习，通过对已知类别训练集的分析，从中发现分类规则，以此预测新数据的类别。分类算法是将一个未知样本分到几个已存在类的过程，主要包含两个步骤：首先，根据类标号已知的训练数据集，训练并构建一个模型，用于描述预定的数据类集或概念集；其次，使用所获得的模型对新的数据进行分类。近年来，许多研究已经转向实现基于 GPU 加速分类算法，包括 k -NN（k 近邻）分类算法[29]，支持向量机分类算法[30]，贝叶斯分类算法[31-32]等。

kNN 算法[33]是数据挖掘中应用最广泛的一种分类算法，简单易实现。它是一种典型的基于实例的学习法，将待判定的检验元组与所有的训练元组进行比较，挑选与其最相似的 k 个训练数据，基于相应的标签和一定的选举规则来决定其标签。

在 Shenshen Liang等人的文章[34]指出，由于 kNN 算法是一种惰性学习法，对于每个待分类的样本，它都需要计算其与训练样本库中所有样本的距离，然后通过排序，才能得到与待分类样本最相邻的 k 个邻居。那么当遇到大规模数据并且是高维样本时，kNN 算法的时间复杂度和空间复杂度将会很高，造成执行效率低下，无法胜任大数据分析任务。所以加速距离的计算是提高kNN 算法的核心问题。因为每个待分类的样本都可以独立地进行 kNN 分类，前后之间没有计算顺序上的相关性，因此可以采用 GPU 并行运算方法解决 kNN 算法串行复杂度高的问题。将计算测试集和训练集中点与点之间的距离和排序一步采用 GPU 并行化完成，其余如判断类标号一步难以在 GPU 上高效实现，由 CPU 完成。文献[34]通过 GPU 并行化实现 kNN 算法，让 kNN算法时间复杂度大幅度减少，从而说明 GPU 对 kNN 算法的加速效果是非常明显的。

3.4 关联分析算法

关联规则挖掘是数据挖掘中较成熟和重要的研究方法，旨在挖掘事务数据库频繁出现的项集。因此，挖掘关联规则的问题可以归结为挖掘频繁项集[35]。关联分析算法首先找出所有的频繁项集，然后根据最小支持度和最小置信度从频繁项集中产生强关联规则。

Apriori 算法[36]是最有影响力的挖掘布尔关联规则频繁项目集的经典算法。Apriori 算法使用逐层搜索的迭代方法产生频繁项目集，即利用 k 频繁项集来产生(k + 1) 项集，是一种基于生成候选项集的关联规则挖掘方法。在刘莹等人的文章[37]中指出，产生候选项和计算支持度，占据 Apriori 的大部分计算量。产生候选项的任务是连接两个频繁项集，而这个任务在不同线程之间是独立的，所以这个过程适合在 GPU 上被并行化。通过扫描交易数据库，计算支持度程序记录一个候选项集出现的次数。由于每个候选项集的计数与其他项集的计数相对独立，同样适合于多线程并行。所以文献[37]的作者们在实现 Apriori时使用 GPU 并行化了产生候选项和计算支持度这两个过程，取得了显著的加速效果。

文献[38]是目前发现的对于在 GPU 上实现频繁项集挖掘最全面细致的研究。他们使用的是早期的CUDA平台，采用了 bitmap 和 trie两种数据结构来实现 GPU 的挖掘算法，并且根据不同数据集和支持度进行了算法性能的对比，均相对于 CPU 版本的算法获得的一定的加速比。

3.5 时序分析

由于越来越多的数据都与时间有着密切的关系，时序数据作为数据挖掘研究的重要分支之一，越来越受到人们的重视。其研究的目的主要包括以下两个方面：一是学习待观察过程过去的行为特征；二是预测未来该过程的可能状态或表现。

时序数据挖掘主要包含以下几个主要任务：数据预处理，时序数据表示，分割，相似度度量，分类，聚类等。这些任务中很多都涉及到相当大的计算量。由于问题规模的不断扩大，并且对于实时性能的要求，时序数据挖掘的任务就必须要求充分地提高计算速度或者通过优化减少计算量。

时序数据的表示有时候会采取特征来表示，这就涉及到了特征提取问题，当特征数量庞大的时候就需要进行维数约简，主要的方法有奇异值分解法，离散小波变换。这些计算都涉及到很大的时间复杂度，为了减少计算的时间消耗，Sheetal Lahabar 等人使用 GPU 加速SVD的计算，获得了60多倍的加速效果[39]。

动态时间弯曲（Dynamic Time Warping，DTW）起初被应用于文本数据匹配和视觉模式识别的研究领域，是一种相似性度量算法。研究表明这种基于非线性弯曲技术的算法可以获得很高的识别、匹配精度。Berndt和 Clifford 提出了将 DTW 的概念引入小型时间序列分析领域，在初步的实验中取得了较好的结果[40]。随着问题规模的扩大，对于 DTW 的计算成为了时序数据挖掘的首先要处理的问题。在 DTW 中，搜索需要找出与训练数据最近距离的样本，这就需要搜索与每个训练样本的距离，这就可以很好地利用 GPU 进行并行化处理。Doruk Sart 等人在对 DTW 加速的处理中，获得了两个数量级的加速效果[41]。

而对于分类和聚类任务的加速，上面已经提到，这里不再累赘。

3.6 深度学习

深度学习虽然隶属机器学习，但鉴于机器学习和数据挖掘领域的紧密联系，深度学习必定将在数据挖掘领域获得越来越多的应用。从 2006 年 Hinton 和他的学生Salakhutdinov 在《科学》上发表的文章[42]开始，深度学习在学术界持续升温。

深度学习的实质是通过构建具有很多隐层的机器学习模型和海量的训练数据，来学习更有用的特征，从而最终提升分类预测的准确性[43]。

如何在工程上利用大规模的并行计算平台来实现海量数据训练，是各个机构从事深度学习技术研发首先要解决的问题。传统的大数据平台如Hadoop，由于数据处理延迟太高而不适合需要频繁迭代的深度学习。神经网络一般基于大量相似的神经元，故本质上可以高度并行化训练；通过映射到 GPU，可以实现比单纯依赖CPU显著的提升。

谷歌搭建的 DistBelief[44]是一个采用普通服务器的深度学习并行计算平台，采用异步算法，由很多计算单元独立更新同一个参数服务器的模型参数，实现了随机梯度下降算法的并行化，加快了模型训练速度。百度的多 GPU 并行计算平台克服了传统 SGD 训练不能并行的技术难题，神经网络的训练已经可以在海量语料上并行展开[43]。

NVIDIA 在 2014 年 9 月推出了深度学习 GPU 加速库 cuDNN，可以方便地嵌入高层级机器学习框架中使用，例如 Caffe[45]。cuDNN 支持 NVIDIA 的全系列 GPU，包括低端的 Tegra K1 和高端的 Tesla K40，并承诺可向上支持未来的 GPU。

4 总结

并行化能带来多少倍的加速取决于算法中可并行化的部分。例如，如果可并行部分的时间占整个应用程序执行时间的 20%，那么即使将并行部分加速 100 倍，总执行时间也只能减少 19.8%，整个应用程序的加速只有 1.247 倍；即使无限加速也只能减少约 20%的执行时间，总加速不会超过 1.25倍。

对于一个数据挖掘（学习和预测）算法进行 GPU 加速实现，首先要思考是否存在可并行执行的部分，之后再结合 GPU 的架构特点进行针对性实现优化。然而，由于数据挖掘算法普遍是数据密集型计算，而 GPU 片内存储容量有限，如何降低与内存交换数据集是一个要解决的关键问题。

通过以上相关工作的分析，可以发现数据挖掘算法在 GPU 上的加速具有数据独立，可并行化共同特征。本文提出数据挖掘算法在 GPU 上加速实现的一种解决思路：在大数据下，分析算法的性能瓶颈，从而确定算法中耗时大，时间复杂度高的部分，将此部分在 GPU 上执行，不耗时部分在 CPU 上串行执行，以达到加速效果。为了更充分利用 GPU 的并行计算的体系结构，可深入分析耗时大的部分，将具有数据独立，可并行化的部分在 GPU上并行执行，达到更进一步的加速效果。

参考文献

1.Han Jiawei，Kamber M，Pei Jian.Data Mining：conceptsand techniques[M].3rd ed.[S.l.]：Morgan Kaufmann，2012：5-8.

2.Owens J D，Houston M，Luebke D，et al.GPU computing[J].Proceedings of the IEEE，2008，96（5）：879-899.

3.NVIDIA CUDA Zone[EB/OL].（2012-03-09）.http：//developer.nvidia.com/page/home.html.

4.Michalakes J，Vachharajani M.GPU acceleration of numericalweather prediction[J].Parallel Processing Letters，2008，18（4）：531-548.

5.刘钦，佟小龙.GPU/CPU 协同并行计算（CPPC）在地震勘探资料处理中的应用[R].2008.

6.Bell N，Garland M.Implementing sparse matrix-vector multiplicationon throughput-oriented processors[C]//Proceedingsof the Conference on High Performance Computing Networking，Storageand Analysis.ACM，2009.

7.Bolz J，Farmer I，Grinspun E，et al.Sparse matrix solverson the GPU：conjugate gradients and multigrid[C]//ACMTransactions on Graphics（TOG）.ACM，2003，22（3）：917-924.

8.Stone J E，Phillips J C，Freddolino P L，et al.Acceleratingmolecular modeling applications with graphics processors[J].Journal of Computational Chemistry，2007，28（16）：2618-2640.

9.Govindaraju N K，Lloyd B，Wang W，et al.Fast computationof database operations using graphics processors[C]//Proceedingsof the 2004 ACM SIGMOD International Conferenceon Management of Data.ACM，2004：215-226

10.NVIDIA Corporation.NVIDIA CUDA programming guideversion 2.3.1[R].2009：1-3.

11.David B K，Wen-mei W H.Programming massively parallelprocessors：A handson approach[M].2nd ed.[S.l.]：ElsevierInc，2006.

12.卢风顺，宋君强，银福康，等.CPU/GPU协同并行计算研究综述[J].计算机科学，2011（3）：5-9.

13.张舒，褚艳利.GPU 高性能运算之 CUDA[M].北京：中国水利水电出版社，2009：1-13.

14.Dagum L，Menon R.OpenMP：an industry standard APforshared-memory programming[J].Computational Science & Engineering. IEEE，1998，5（1）：46-55.

15.Wienke S，Springer P，Terboven C，et al.OpenACC—first experiences with real-world applications[M]//EuroPar2012 Parallel Processing.Berlin Heidelberg：Springer，2012：859-870.

16.Breitbart J，Fohry C.OpenCL-an effective programmingmodel for data parallel computations at the cell broadbandengine[C]//2010 IEEE International Symposium onParallel & Distributed Processing，Workshops and PhdForum（IPDPSW）. IEEE，2010：1-8.

17.王恒，高建瓴.基于 GPU 的 MATLAB 计算与仿真研究[J].贵州大学学报：自然科学版，2012，6：95-98.

18.Mark H，Eibe F，Geoffrey H，et al.The WEKA data miningsoftware：An update[J].SIGKDD Explorations，2009，11（1）.

19.Jeffry D，Sanjay G.MapReduce：Simplied data processingon large clusters[C]//Proceedings of 6th Symposium onOperating System Design and Implementation（OSDI），2004.

20.Sean O，Robin A，Ted D，et al.Mahout in action[M].[S.l.]：Manning Publications，2011.

21.Cheng-Tao C，Sang K K，Yi-An L，et al.Map-Reduce formachine learning on multicore[C]//Proceedings of 20thNeural Information Processing Systems（NIPS），2006：281-288.

22.孙吉贵，刘杰，赵连宇.聚类算法研究[J].软件学报，2008，19（1）：48-61.

23.周涛，陆惠玲.数据挖掘中聚类算法研究进展[J].计算机工程与应用，2012，48（12）：100-111.

24.周爱武，于亚飞. K -Means 聚类算法研究[J].计算机技术与发展，2011，21（2）：62-65.

25.Farivar R，Rebolledo D，Chan E，et al.A parallel implementationof K-Means Clustering on GPUs[C]//Proceedingsof the 2008 International Conference on Parallel andDistributed Processing Techniques and Applications.[S.l.]：Springer-Verlag，2008：340-345.

26.Guilherme A，Gabriel R，Daniel M，et al.G-DBSCAN：A GPU accelerated algorithm for density-based clustering[J]. Procedia Computer Science，2013，18：369-378.

27.Jianqiang D，Fei W，Bo Y.Accelerating BIRCH for clusteringlarge scale streaming data using CUDA dynamicparallelism[C]//Proceedings of 14th International Conferenceon Intelligent Data Engineering and AutomatedLearning（IDEAL），2013.

28.郭炜星.数据挖掘分类算法研究[D].杭州：浙江大学，2008.

29.Garcia V，Debreuve E，Barlaud M.Fast k nearest neighborsearch using GPU[C]//IEEE Computer Society Conferenceon Computer Vision and Pattern Recognition Workshops. IEEE，2008：1-6.

30.Catanzaro B，Sundaram N，Keutzer K.Fast support vectormachine training and classification on graphics processors[C]//Proceedingsof the 25th International Conferenceon Machine Learning.ACM，2008：104-111.

31.Andrade G，Viegas F，Ramos G S，et al.GPU-NB：A fastCUDA-based implementation of Naïve Bayes[C]//201325th International Symposium on Computer Architectureand High Performance Computing（S BAC-PAD）. IEEE，2013：168-175.

32.刘琳，何剑锋，王红玲.GPU 加速数据挖掘算法的研究[J].郑州工业大学学报：理学版，2010，42（2）：31-34.

33.潘丽芳，杨炳儒.基于簇的 K 最近邻 (KNN) 分类算法研究[J].计算机工程与设计，2009，30（18）：4260-4262.

34.Liang S，Liu Y，Wang C，et al.A CUDA-based parallelimplementation of K-nearest neighbor algorithm[C]//InternationalConference on Cyber-Enabled Distributed Computingand Knowledge Discovery. ieee，2009：291-296.

35.Han J，Cheng H，Xin D，et al.Frequent pattern mining：current status and future directions[J].Data Mining andKnowledge Discovery，2007，15（1）：55-86.

36.崔贯勋，李梁，王柯柯，等.关联规则挖掘中 Apriori算法的研究与改进[J].计算机应用，2010（11）：2952-2955.

37.刘莹，菅立恒，梁莘燊，等.基于 CUDA架构的 GPU的并行数据挖掘技术研究[J].科研信息化技术与应用，2010（4）：38-52.

38.Fang W，Lu M，Xiao X，et al.Frequent itemset mining ongraphics processors[C]//Proceedings of the Fifth InternationalWorkshop on Data Management on New Hardware.ACM，2009：34-42.

39.Lahabar S，Narayanan P J.Singular value decompositionon GPU using CUDA[C]//2009 IEEE International Symposiumon Parallel & Distributed Processing. ieee，2009：1-10.

40.Berndt D J，Clifford J.Using dynamic time warping tofind patterns in time series[C]//KDD Workshop，1994，10（16）：359-370.

41.Sart D，Mueen A，Najjar W，et al.Accelerating dynamictime warping subsequence search with GPUs and FPGAs[C]//2010 IEEE 10th International Conference on Data Mining（ICDM）. ieee，2010：1001-1006.

42.Hinton G E，Salakhutdinov R R.Reducing the dimensionalityof data with neural networks[J]. Science，2006，313（5786）：504-507.

43.余凯，贾磊，陈雨强，等.深度学习的昨天、今天和明天[J].计算机研究与发展，2013，50（9）：1799-1804.

44.Dean J，Corrado G，Monga R，et al.Large scale distributeddeep networks[C]//Advances in Neural Information ProcessingSystems，2012：1223-1231.

45.Yang Q J.Caffe：An open source convolutional architecturefor fast feature embedding[Z].2013.