

附件 1

研究课题报告

大型复杂数据和流数据的分析与建模

(课题开始时间 2006.10 至今)

张响亮

法国自动化和计算机国家研究院 (INRIA) &

法国巴黎十一大学

Xiangliangzhang@gmail.com; xlzhang@lri.fr

简介：本课题报告介绍我在法国自动化和计算机国家研究院(INRIA)和巴黎十一大学读博士至今 (2006.10 - 2009.11) 的主要工作。我的博士论文方向是“基于统计机器学习方法的大型复杂数据和流数据的分析与建模”。我将通过 1) 研究动机和目的, 2) 研究方法, 3) 研究结果, 4) 总结等 4 个部分来介绍该课题。

1. 研究动机和目的

本课题的起源是为了解决“网格计算”中的运行管理和监控问题。这里提到的“网格”指的是欧洲最大的网格计算系统, EGEE (**Enabling Grids for E-science**)。它在 50 多个国家拥有 250 个站点, 共包含 68000 个 CPU, 24 小时不间断提供服务, 每天大约有上百万的任务 (jobs) 运行在其中。

对于这样一个庞大而复杂的分布式计算系统, 它的管理和监控需要很多熟练的人工操作去发现某些站点是否有问题, 导致某些任务执行失败或丢失等。然而面对结构复杂、数据量极大的工作日志 (logs) 来说, 仅仅凭借人工简单的查看是不可能做不到的。

我们课题要解决的问题就是, 通过“机器学习” (Machine Learning) 和“数据挖掘” (Data Mining) 的方法, 对大型复杂的日志数据进行建模和分析, 提取出能反映数据的有意义的模式, 以一种简单、可视的方式表达给管理员, 帮助他们能非常直观地发现并解决问题。

在实现这个解决方案的过程中, 我们遇到了一些挑战:

- i) 数据量超大, 数据结果异常复杂。正如前文所述, 整个网格系统中运行的任务非常多, 而每一个任务在日志中都有几十个特征来描述。同时, 这些描述性的特征又有着不同的属性和类别。

- ii) 数据的动态性。网格中的任务来自各个应用领域，如物理，化学，生物，天文等。这些任务在用户和时间方面都表现出非常大的动态变化。比如，有的用户提交的任务需要大的 CPU，有的用户需要大的内存空间；工作日中网格繁忙的时候，任务的处理会需要多的等待时间，而周末比较空闲时，则会相对轻松。这种高速的并随时间出现发生变化的数据，也称为“流数据”（streaming data）。如何处理流数据是一个难题。

在 3 年的研究中，我在导师的指导下可以说很好的解决了这 2 个数据挖掘领域的热点问题。

2. 研究方法

目前的数据挖掘领域，已经有一些针对大型数据和流数据的文献[1-6]。我们的工作与这些文献一样，都关注用聚类（clustering）算法将相似的点聚到一起成为类。这样整个数据就被若干个类（cluster）来表述。我们建立一个模型，这个模型记录这些类的中心，半径以及成员数等等。由此以来，数据就被模型所描述。

对于大型复杂数据和流数据，我们不可能按照常规做法，一次性将所有数据都拿来聚类，因为数据量太大而且数据在动态变化之中，一次性聚类需要很大的计算量，而且没有实时性。常用的解决方案就是，对大型数据进行分割，先局部分别聚类，然后再集中到一起整体聚类；对于流数据，由于其动态变化的特点，一般采用在线聚类（online clustering）的方法，即时更新模型。然而，目前存在的相关研究方法还有一些缺陷。具体表现在：

- i) 聚类时必须提前定义整个数据中类的个数。这是目前经典的聚类算法 k-means, k-centers 的必须要求。这个先验知识是很难获取的。另外更重要的一点是，对于流数据，可聚成的类的个数也是在不断变化的，因此我们很难给定一个固定的值作为参数。
- ii) 流数据的结构和分布是一直在不断变化的。算法的模型应该具备跟踪变化的能力，并且检测新的类模式是否出现，并将它添加到当前模型。同时，如果模型中某些类模式已经老化，我们也应该将其弱化或者剔除，以节省空间且精确描述模型。
- iii) 模型中关于类的描述，要能实时呈现出来，通过可视化方式让管理员清楚的知道目前关于数据的概括描述。

为了解决聚类算法的问题，我们采用了一个非常新的方法，叫做 Affinity Propagation，简称 AP [7]，2007 年发表在 Science 上。该算法通过每一对点与点之间的消息传递迭代来最终收敛到稳定的类的划分。它的优点就是不需要提前给定类的个数，并且聚类结果优于传统的 k-centers 方法。但是它的致命缺点就是计算量非常大，复杂度达到 $O(N^2 \log N)$ ，这里 N 是数据中点的个数。可想而知，大型数据中点的个数非常大，那么用 AP 来聚类将是一个不可能完成的任务。

针对 AP 的这个缺点，我们首先提出了加权 AP（weighted AP（WAP））的方法。在不改变所传递的信息量的前提下，对临近点的进行整合，使得整合过的点的聚类结果与原始点的聚类结果相同。

在此基础上，我们还提出了针对大型数据的分级 AP (Hierarchical AP (Hi-AP)) 的方法。具体做法是，先将大型数据 ξ 分块成 N^2 个子集 ξ_i ，然后对每一个子集 ξ_i 分别使用 AP 聚类，得到类的中心点集合 $E_i = \{e^j, n^j\}$ ，其中 e^j 是某个类的中心点， n^j 是某类中元素的个数。然后，我们对 N^2 个 E_i 做集合，在这些中心点使用 WAP 进行聚类，此时，每个中心点代表的点的个数 n^j 被考虑，影响该点传递信息的量。因此在这些中心点上用 WAP 聚类，就等价于用 AP 对所有点的聚类。

我们提出的 Hi-AP 方法在 2 组数据中分别被验证，实验结果表明，Hi-AP 大大的降低了计算时间（从几分钟到几秒钟），但是聚类的效果并没有太大改变。有关结果发表在欧洲机器学习与数据挖掘领域顶级会议 ECML/PKDD '2008 中 [9]（同时也是国际机器学习与数据挖掘领域最好的会议之一）。同时，为了从理论上更好地验证 Hi-AP 算法的效果，我们对它从数学理论上进行了证明，发表在国际数据挖掘领域中最顶级的会议 SIGKDD '2009 中[10]。

解决了大型数据的聚类之后，我们来考虑流数据的聚类问题。我们提出了一个在线的流数据聚类算法，称为 StrAP。该算法的实现简单但非常实用有效，借助算法描述表 1，可以清楚的描述它的构造。

首先，如果有一个流数据，各个点数据按时间顺序到达。我们对先期到达的一小部分点用 AP 进行聚类，得到 StrAP 初始模型。

然后，当有一个点 x_i 到达，将它与 StrAP 模型中的中心点比较，找出与它距离最近的点 x_j 。

接着，判断点 x_i 与 x_j 之间的距离 $d(x_i, x_j)$ 与阈值 ε 的关系。如果距离比阈值小，说明点 x_i 与模型是可匹配的，因此，我们简单更新 StrAP 模型中对应的以 x_j 为中心的那个类。如果距离比阈值大，说明点 x_i 与模型距离很远，是一个新的类或噪音，当前模型不能接受它，因此将它放入一个暂存盒 (Reservoir)，等待后续处理。

暂存盒中的点有可能是一些新的类模式，有必要在适当的时候将他们加入到模型中去。关键是什么样的时候才适当。我们有 2 个判断的标准去重新建立 StrAP 模型。一是给定暂存盒的大小，另一个是用一种检测变化的统计方法，称为 Page-Hinkley [8]。当满足这 2 个标准之一时，我们用 WAP 对现有模型中的中心点和暂存盒中的点，一起进行聚类，从而得到新的模型。接着对新来的点，我们重复以上过程。通过这个过程，我们既考虑了数据的变化（数据变化检测），又降低了算法的复杂度（分级聚类和加权聚类）。可以满足大型复杂流数据的实时处理。

StrAP 算法描述表 1

输入: 数据流 x_1, \dots, x_t, \dots , 匹配阈值 ε	
初始化: AP 对数据流先期到达的部分 (x_1, \dots, x_T) T 个点聚类, 得到 StrAP 模型; 暂存盒={}	
在线聚类:	
--1	某新的点 x_t 到达
--2	在模型中找到与 x_t 距离最近的中心点 x_i 如果 $d(x_t, x_i) < \varepsilon$
--3	更新 StrAP 模型 否则
--4	将 x_t 放入 暂存盒 中 如果 检测到分布情况变化
--5	重新建立 StrAP 模型
--6	清空 暂存盒
去第 1 步 重复执行直到结束	

我们对 StrAP 分别在 一组人工数据和基准数据 (benchmark), 进行了多方位的验证。同时还与文献[3]中的方法进行了比较。实验结果表明, 我们比文献[3]中的 DenStream 方法, 在聚类准确度方面表现好很多。具体结果发表在 ECML /PKDD 文章中[9]。

在验证了 StrAP 在基准数据上的有效性之后, 我们将它用在解决开篇提到的“网格监控问题”。图 1 是该系统的结构图。

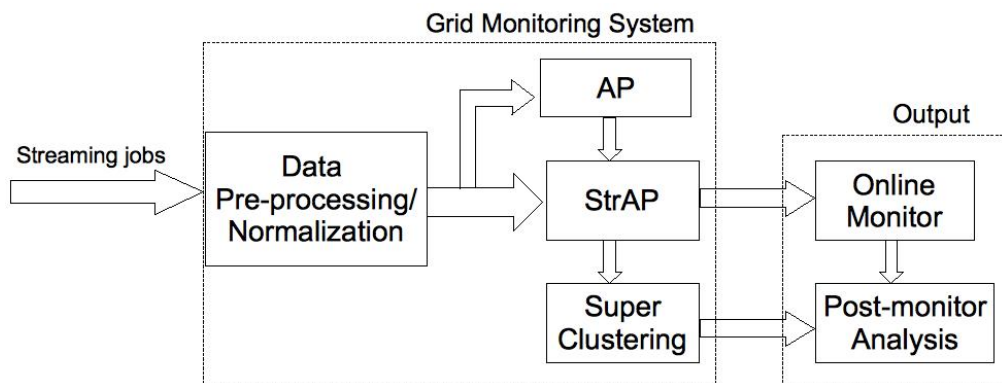


图 1 实时在线网格监控系统结构图

我们将“任务流” (job streams) 作为输入, 经过初始化之后, 每个任务被在线聚类。StrAP 模型以一种总结性描述的方式记录了所有的“任务模式类” (job pattern)。这种简洁的描述可

以很容易被可视化(online monitor), 作为输出供管理员查看。同时还有一个 super clustering, 就是像 Hi-AP 第二层那样对 StrAP 中产生的类再次聚类, 用来做历史数据分析(Post-monitor)。关于这个“在线实时网络监控系统”我们发表在网格计算类顶级会议 CCGrid 中[11]。

3 研究结果

在本报告中, 我们将呈现一部分的实验结果, 更多的详细研究结果发表在[9-11]中。

图 2 是关于大规模数据聚类算法 Hi-AP 的验证, 我们使用了 237,087 个 EGEE 任务。图 2 中以聚类的 Distortion 作为衡量标准, 我们比较了 Hi-AP 算法和作为基准的 k-centers 算法。聚类结果的 Distortion 定义为每个点和它所属类的中心点之间距离的平方和, $D = \sum_{i=1}^N d^2(x_i, c_i)$, 其中 c_i 是 x_i 点所属类的中心。很明显, Distortion 越小, 就说明相似的点都聚在一个类中, 因此聚类的效果就越好。当然, Distortion 的比较是要基于相同的类个数 (K), K 越大, Distortion 会越小。所以, 从图 2 中我们可以看出在相同的 K 条件下, Hi-AP 比基准 Hi-k-centers 有更小的 Distortion。这里的 Hi-AP simple 与 Hi-AP 的唯一不同, 就是再次聚类时, Hi-AP 使用 WAP, 而 Hi-AP simple 使用 AP。从而, Hi-AP 的聚类效果也比 Hi-AP simple 要好。

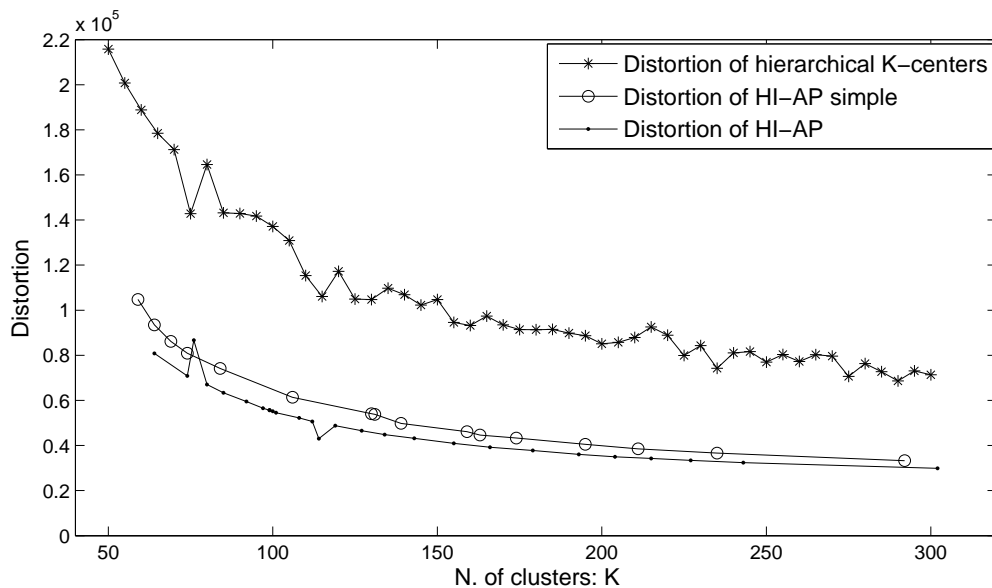


图 2 Hi-AP 在 EGEE 任务上的验证结果

图 3 是我们提出的流数据聚类的 StrAP 方法与[3]中的 DenStream 方法比较的结果。两个方法都是基于完全相同的数据, 一种在入侵检测领域里广泛使用的 Benchmark 数据 (KDDcup 1999)。494,021 条网络连接被在线聚类, 关于聚类结果的衡量, 我们使用了聚类纯度这个标准

$$purity = \frac{1}{K} \sum_{i=1}^K \frac{|C_i^d|}{|C_i|},$$

其中 K 是类个数, $|C_i|$ 是 i 类中包含的点数, $|C_i^d|$ 是 i 类中占主导性的

的点数。这里主导性可以解释为, 在一个类 C_i 包含的点中, 它们都有自己的 label (表明自己的

属性)，有一种 label 是主导性的，因为拥有这个 label 的点最多。所以，如果一个类中的点都有相同的 label，那么它的纯度就是 100%。从全局出发，我们对 K 个类的纯度做平均，得到一种分类结果的纯度。

从图 3 可以看出，我们提出的 StrAP 明显比 DenStream 方法有较高的聚类纯度，具有更好的聚类效果。

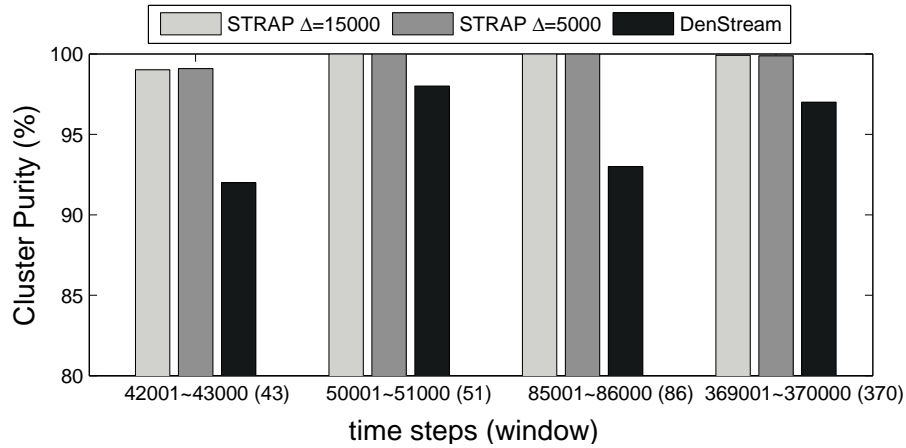
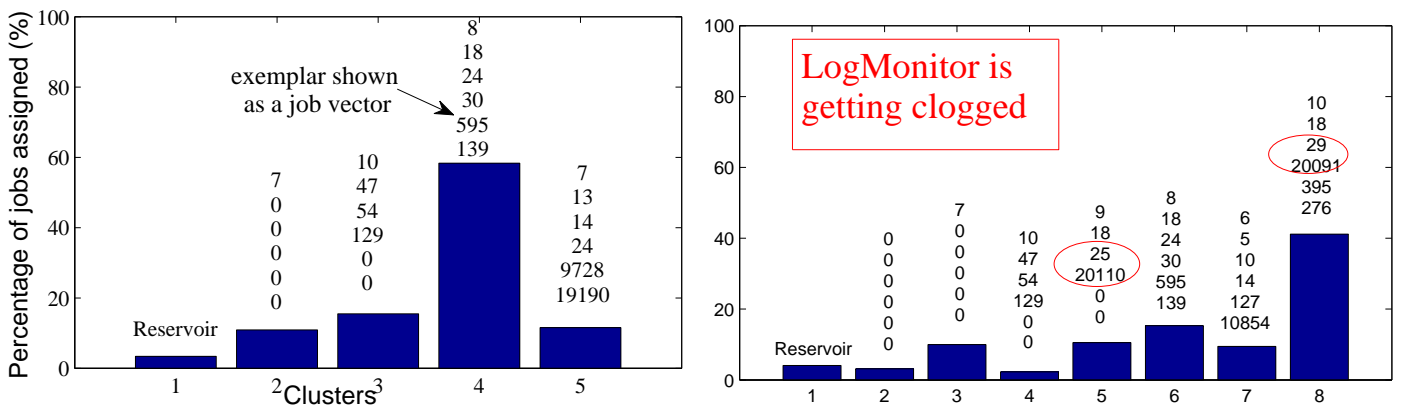


图 3 StrAP 在 benchmark 数据上的验证和比较结果

图 4 我们呈现的是基于 StrAP 而建立的“在线实时网络监控系统”的一部分在线输出。初步实验中，系统检测了 5 百多万条任务数据记录。在线输出以柱状图的方式表现模型中各个类。柱子的高度表示属于该类的点所占百分比。柱子上面的数字描述了该类中心的数字特征，其实这个类中其它的点都与中心点有相似的描述。图 4(a)是初始化后不久某时刻的输出，图 4 的(b)是另一个时刻的输出。在图 4(b)中，我们发现 2 个类，第 5 个和第 8 个，它们的中心点描述有不正常的现象。经过专家判断，这种情况的发生是因为，网络系统中叫 LogMonitor 的设备有堵塞。

图 4 说明了，“在线实时网络监控系统”的在线输出的确可以有效地帮助管理员查看和分析网络的运行状况，以便采取相应的措施来维护网络的正常工作。除了在线输出外，该系统还可以提供历史数据分析，帮助找出某几天内频繁发生的某种错误是源于某站点的执行故障。具体结果可参见文章[10],[11]中的分析结果。



(a)

(b)

图 4 在线实时网络监控系统的在线输出, (a)和(b)分别是不同时刻的输出

4 总结

关于“大型数据”和“流数据”的统计分析课题, 源起于“复杂系统网格”的管理和分析。在对本课题的研究中, 首先我们通过 **Hi-AP** 解决了“大型数据”的聚类问题, 通过 **StrAP** 解决了“流数据”的聚类问题。在此基础上, 我们建立了“在线实时网络监控系统”。

关于这 3 个部分的贡献, 我们都分别在多种数据上验证过, 并与基准方法进行了比较。实验结果表明, 本课题中提出的方法 **Hi-AP** 和 **StrAP**, 可以更有效, 更方便的解决大型复杂数据和流数据的聚类问题。同时, 所开发的监控系统, 也确实能够将网格的运行状况以可视化的方式呈现给管理员, 帮助他们分析和维护网格情况。

事实上, **Hi-AP** 和 **StrAP** 还可以被应用在更多方面, 只要是需要对大型数据和流数据进行聚类的。一个例子就是对网络连接的检测, 网络中的异常情况有千万种, 但正常情况的模式都比较接近。通过 **StrAP**, 可以自主地建立并更新正常模型, 则与正常模型相悖的就要引起注意了。这方面的研究工作已经开始, 例如其他研究人员应用我们提出的 **StrAP** 方法在自治的入侵检测中, 相关的结果发表在[12-14]中。

我们下一步的研究分为两个方面, 一方面是理论部分, 我们将完善 **StrAP** 在线聚类的参数问题, 之前的参数都是根据经验值选择, 在[10]中我们尝试了用优化的方式来设置参数, 接下来更多的简便方法将进一步使参数自适应的调整。另一个方面是应用部分, 除了网络连接的应用外, 我们还正在尝试用 **StrAP** 对法国移动供应商 **ORANGE** 的手机用户进行分析。将来还会有更多的应用领域等待我们尝试。

参考文献

- [1] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for clustering evolving data streams. In Proceedings of the International Conference on Very Large Data Bases(**VLDB**), pages 81–92, 2003.
- [2] Paul S. Bradley, Usama M. Fayyad, and Cory Reina. Scaling clustering algorithms to large databases. In Knowledge Discovery and Data Mining, pages 9–15, 1998.
- [3] F. Cao, M. Ester, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. In SIAM Conference on Data Mining (**SDM**), pages 326–337, 2006.
- [4] Yixin Chen and Li Tu. Density-based clustering for real-time stream data. In KDD '07: Proceedings of the 13th ACM **SIGKDD** international conference on Knowledge discovery and data mining, pages 133–142, 2007.
- [5] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. In **SIGMOD** '98: Proceedings of ACM **SIGMOD** international conference on Management of data, pages 73–84, 1998.
- [6] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan.

- Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 15:515–528, 2003.
- [7] B. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315: 972–976, 2007.
- [8] D. Hinkley. Inference about the change-point from cumulative sum tests. *Biometrika*, 58: 509–523, 1971.
- [9] Xiangliang Zhang, Cyril Furtlehner, and Michele Sebag. Data streaming with affinity propagation. In *European Conference on Machine Learning and Practice of Knowledge Discovery in Databases, ECML/PKDD*, pages 628–643, 2008.
- [10] Xiangliang Zhang, Cyril Furtlehner, Julien Perez, Cecile Germain-Renaud, and Michele Sebag. Toward autonomic grids: Analyzing the job flow with affinity streaming. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.
- [11] Xiangliang Zhang, Michele Sebag, and Cecile Germain-Renaud. Multi-scale real-time grid monitoring with job stream mining. In *9th IEEE International Symposium on Cluster Computing and the Grid (CCGrid)*, 2009.
- [12] Wei Wang, Thomas Guyet, Svein J. Knapskog, "Autonomic Intrusion Detection System", *Proceedings of 12th International Symposium On Recent Advances in Intrusion Detection (RAID)*, pp. 359-361, 2009.
- [13] Wei Wang, Florent Masegla, Thomas Guyet, Rene Quiniou and Marie-Odile Cordier, "A General Framework for Adaptive and Online Detection of Web attacks". *Proceedings of 18th international World Wide Web conference (WWW)*, pp. 1141-1142, 2009 .
- [14] Wei Wang, Thomas Guyet, Rene Quiniou, Marie-Odile Cordier, Florent Masegla, "Online and adaptive anomaly Detection: detecting intrusions in unlabelled audit data streams". *Proceedings of conference Extraction et Gestion des Connaissances (EGC)*, pp. 457-458, 2009.