Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Information Sciences 176 (2006) 2016–2041

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Streaming data reduction using low-memory factored representations

David Littau [*], Daniel Boley

Department of Computer Science and Engineering, University of Minnesota Twin Cities,
200 Union Street, SE, Minneapolis, MN 55455, United States

Abstract

Many special purpose algorithms exist for extracting information from streaming data. Constraints are imposed on the total memory and on the average processing time per data item. These constraints are usually satisfied by deciding in advance the kind of information one wishes to extract, and then extracting only the data relevant for that goal. Here, we propose a general data representation that can be computed using modest memory requirements with limited processing power per data item, and yet permits the application of an arbitrary data mining algorithm chosen and/or adjusted after the data collection process has begun. The new representation allows for the at-once analysis of a significantly larger number of data items than would be possible using the original representation of the data. The method depends on a rapid computation of a factored form of the original data set. The method is illustrated with two real datasets, one with dense and one with sparse attribute values.

Keywords: Streaming data; Data reduction; Clustering; PDDP; Matrix approximation

[*] Corresponding author.
E-mail addresses: littau@cs.umn.edu (D. Littau), boley@cs.umn.edu (D. Boley).

0020-0255/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2005.11.002

1. Introduction

As data sets have become larger and often unbounded, the concept of a data stream has become a useful model in data mining applications. The data stream model is applicable to data sets in which new data arrive constantly, such as network connection data, credit card transaction data, etc. Such data are usually examined once and either archived or deleted. The data stream model is also applicable to data sets which are so large that they cannot fit into memory at once. If the data cannot fit into memory it becomes expensive to apply data mining algorithms based on the static data model, since such algorithms usually scan the data several times.

In the data stream model [13], data points can only be accessed in the order in which they appear in the stream. Random access of data is not allowed. The amount of memory available is severely limited, much less than what would be required to store all the data at once. Working under these constraints, the goal of our streaming data mining pre-processing method is to extract as much useful information as possible from the data in a reasonable amount of time, while not fixing the task to be performed on the data in advance.

The usual approach in streaming data mining applications [1,2,8,10,12, 16,15,21] is to first decide what data mining task is to be performed, and then tailor the processing to gather the information necessary for the specific task. This is an efficient approach to mining stream data. It allows the data to be collapsed into some very tiny, efficient representation whose memory footprint is limited to some constant size. The drawback to this technique is that if another task needs to be completed, it is often necessary to process the data again to gather the information for the different task. This can become very expensive very quickly.

One way to maintain the flexibility with respect to the task is to apply some kind of approximation technique. Representations of the data are constructed which are designed to reflect the most important qualities of the data while taking up less room than the original data, such as in the summaries used in [6,24]. Most representations of the data assign many data points to one representative, which means that individual data points are no longer distinguishable.

We present a pre-processing method for streaming data which does not require the data mining task to be selected beforehand. The data are represented in the vector space model. We create a low-memory factored representation (LMFR) of the data, such that each data point has a unique representation in the factored form. We accumulate data from the stream in the form of chunks we call *sections*. Once a given section of data has been processed it does not need to be accessed again. The low-memory representation allows for the at-once mining of a much larger piece of data than would be possible when using the original representation of the data. Any tasks which require the data points to be distinct, such as searching for outliers or