Available online at www.sciencedirect.com
ScienceDirect
Computational Statistics & Data Analysis 51 (2006) 215–234
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

Extending fuzzy and probabilistic clustering to very large data sets

Richard J. Hathaway[a,*], James C. Bezdek[b]

[a] *Department of Mathematical Sciences, Georgia Southern University, Statesboro, GA 30460, USA*
[b] *Department of Computer Sciences, University of West Florida, Pensacola, FL 32514, USA*

Available online 2 March 2006

Abstract

Approximating clusters in very large (VL = unloadable) data sets has been considered from many angles. The proposed approach has three basic steps: (i) progressive sampling of the VL data, terminated when a sample passes a statistical goodness of fit test; (ii) clustering the sample with a literal (or exact) algorithm; and (iii) non-iterative extension of the literal clusters to the remainder of the data set. Extension accelerates clustering on all (loadable) data sets. More importantly, extension provides feasibility—a way to find (approximate) clusters—for data sets that are too large to be loaded into the primary memory of a single computer. A good generalized sampling and extension scheme should be effective for acceleration and feasibility using any extensible clustering algorithm. A general method for progressive sampling in VL sets of feature vectors is developed, and examples are given that show how to extend the literal fuzzy (c-means) and probabilistic (expectation-maximization) clustering algorithms onto VL data. The fuzzy extension is called the generalized extensible fast fuzzy c-means (geFFCM) algorithm and is illustrated using several experiments with mixtures of five-dimensional normal distributions.

Keywords: Clustering; Data mining; Extensibility; Fuzzy c-means; Goodness-of-fit; Progressive sampling; Very large data sets

1. Introduction

Huber (1996) classifies data set size as in Table 1. Huber states that "Some simple standard database management tasks with computational complexity $\mathbf{O}(n)$ or $\mathbf{O}(n \log n)$ remain feasible beyond terabyte monster sets, while others (e.g., clustering) blow up already near large data sets." We have added one column to Huber's table, called very large (VL) data.

Today data of more than 10 gigabytes is probably beyond the primary memory capacity of most workstations. Different computers can handle different maximally sized data sets, and their capacity will continue to increase, but so will data size. There will always be data sets that are simply too large for any computer, so methods that are extensible to VL data sets are of continued importance (Cutting et al., 1992; Baeza-Yates and Ribeiro-Neto, 1999).

The data set to be clustered is either X_{L} or X_{VL}. These two clustering situations are shown in Fig. 1, where X_{∞} denotes the population from which the data is drawn; X_{VL} represents a very large data set that cannot be loaded into primary memory; X_{L} represents a large data set that can be loaded into primary memory; and X_{SS} represents a subset (or subsample) of either X_{VL} or X_{L}. In a nutshell, the proposed fuzzy (geFFCM: generalized extended fast

* Corresponding author. Tel.: +1 912 681 5619; fax: +1 912 681 0654.
E-mail addresses: rhathaway@georgiasouthern.edu, r.hathaway@ieee.org (R.J. Hathaway).

0167-9473/$ - see front matter © 2006 Published by Elsevier B.V.
doi:10.1016/j.csda.2006.02.008

Table 1
Size of data sets, after Huber (1996)

Bytes	10^2	10^4	10^6	10^8
"size"	tiny	small	medium	large
	10^{10}	10^{12}	$10^{n>12}$	∞
	huge	monster	VL	Infinite

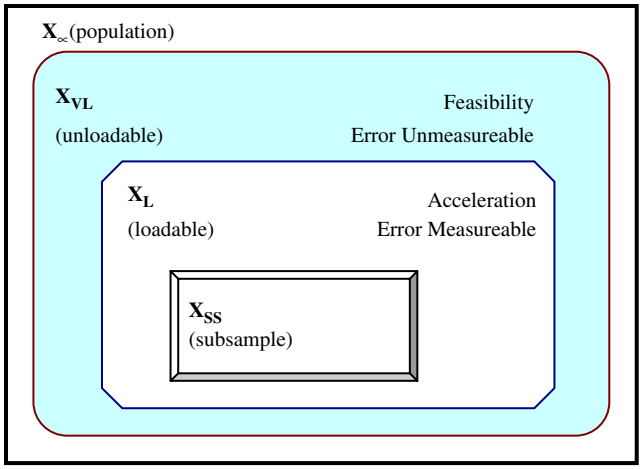

Fig. 1. Population X_∞ and samples X_{VL}, X_L, X_{SS}.

fuzzy c- means) and probabilistic (geFEM: generalized extended fast expectation maximization) algorithms choose X_{SS}, cluster it, and then extend the result to X_L or X_{VL}.

There are some fundamental differences between the two cases. Our test for judging whether X_{SS} is representative of the source sample depends on being able to process the full sample. We can load X_L into primary memory and do the required processing (to test the subsample). We cannot load X_{VL}, but it is often feasible to page through X_{VL} once to gather simple statistics (e.g., bin counts for a histogram) needed to assess the quality of candidate subsamples.

Another fundamental difference between the two cases is the calculation of approximation error. We call the application of any algorithm to an entire data set a literal implementation. (The machine learning community terminology is ⟨unsupervised⟩ learning with all the examples.) In this paper FCM refers to the popular fuzzy c-means clustering algorithm, which we will also sometimes denote as LFCM (literal FCM) to distinguish the exact (literal) implementation of FCM from approximations to it. If the available data set is X_L, we can assess the quality of an extended clustering by comparing it to the literal clustering obtained using the same parameters on the whole data set. But if the set is X_{VL}, then a quality comparison is not possible because the literal clustering (using all the examples) cannot be obtained. Our confidence in the accuracy of geFFCM (generalized extended fast expectation maximization) in the unverifiable case (X_{VL}) is based on its verified good behavior for various X_L experiments.

Scalability is often cited as a qualification for clustering algorithms for VL data. The usual definition of scalability: an algorithm is scalable if its runtime complexity increases linearly with the number of records in the input data (Ganti, 1999a). Scalability is often confused with a related issue—viz., acceleration of existing algorithms. Indeed, the c-means algorithms (Bezdek et al., 1999) are all scalable in the just defined sense, but are famously slow when processing lots of samples, so scalability alone is not enough. And while we are always on the lookout for ways to make clustering algorithms faster, no amount of acceleration solves the VL data problem, wherein the data cannot be processed in aggregate at all.

Progressive sampling is used in various ways for both c-means and other clustering approaches. This interesting field has seen a lot of growth in recent years. Provost et al. (1999) provide a very readable analysis and summary of progressive sampling schemes. They assert that the central component of any progressive sampling scheme is the