Report Number: wucse-2009-30

2009

Online Bayesian Analysis

Ruibin Xi, Yongjin Kim, Nan Lin, Yixin Chen, and Gruia-Catalin Roman

In the last few years, there has been active research on aggregating advanced statistical measures in multidimensional data cubes from partitioned subsets of data. In this paper, we propose an online compression and aggregation scheme to support Bayesian estimations in data cubes based on the asymptotic properties of Bayesian statistics. In the proposed approach, we compress each data segment by retaining only the model parameters and a small amount of auxiliary measures. We then develop an aggregation formula that allows us to reconstruct the Bayesian estimation from partitioned segments with a small approximation error. We show that the Bayesian... **Read complete abstract on page 2.**

Online Bayesian Analysis

Ruibin Xi, Yongjin Kim, Nan Lin, Yixin Chen, and Gruia-Catalin Roman

Complete Abstract:

In the last few years, there has been active research on aggregating advanced statistical measures in multidimensional data cubes from partitioned subsets of data. In this paper, we propose an online compression and aggregation scheme to support Bayesian estimations in data cubes based on the asymptotic properties of Bayesian statistics. In the proposed approach, we compress each data segment by retaining only the model parameters and a small amount of auxiliary measures. We then develop an aggregation formula that allows us to reconstruct the Bayesian estimation from partitioned segments with a small approximation error. We show that the Bayesian estimates and the aggregated Bayesian estimates are asymptotically equivalent.

2009-30

Online Bayesian Analysis

Authors: Ruibin Xi, Yongjin Kim, Nan Lin, Yixin Chen

Corresponding Author: chen@cse.wustl.edu

Abstract: In the last few years, there has been active research on aggregating advanced statistical measures in multidimensional data cubes from partitioned subsets of data. In this paper, we propose an online compression and aggregation scheme to support Bayesian estimations in data cubes based on the asymptotic properties of Bayesian statistics.

In the proposed approach, we compress each data segment by retaining only the model parameters and a small amount of auxiliary measures. We then develop an aggregation formula that allows us to reconstruct the Bayesian estimation from partitioned segments with a small approximation error. We show that the Bayesian estimates and the aggregated Bayesian estimates are asymptotically equivalent.

Type of Report: Other

Online Bayesian Analysis

Ruibin Xi
Department of Mathematics
Washington University
rxi@math.wustl.edu

Youngjin Kim
Department of Computer Science
Washington University
moah.kim@gmail.com

Nan Lin
Department of Mathematics
Washington University
nlin@math.wustl.edu

Yixin Chen
Department of Computer Science
Washington University
chen@cse.wustl.edu

Abstract

1 Introduction

In the last few years, there has been active research on aggregating advanced statistical measures in multi-dimensional data cubes [10] from partitioned subsets of data. In this paper, we propose a compression and aggregation scheme to support Bayesian estimations in data cubes based on the asymptotic properties of Bayesian statistics. The main application of the technique developed in this paper is data warehousing and the associated on-line analytical processing (OLAP) computing. OLAP allows for interactive analysis of multidimensional data to facilitate effective data mining at multiple levels of abstraction.

Earlier work in data cubes [10] supports aggregation of simple measures such as sum() and average(). However, the fast development of OLAP technology has led to high demand for more sophisticated data analyzing capabilities, such as prediction, trend monitoring, and exception detection of multidimensional data. Oftentimes, existing simple measures such as sum() and average() become insufficient, and more sophisticated statistical models are desired to be supported in OLAP. Recently, some researchers developed aggregation schemes for more advanced statistical analysis including parametric models such as linear regression [6, 11] general multiple linear regression [5, 14] logistic regression analysis [23] and predictive filters [5], as well as nonparametric statistical models such as naive Bayesian classifiers [4] and linear discriminant analysis [15]. Along this line, we develop an aggregation scheme support Bayesian estimations in data cubes.

Bayesian methods are statistical approaches to parameter estimation and statistical inference which use prior distributions over parameters. Bayesian methods have been successfully applied in many different fields such business, computer science, economics, epidemiology, genetics, imaging and political science. The premise of Bayesian statistics is to incorporate prior knowledge, along with a given set of

current observations, in order to make statistical inferences. The prior information could come from previous comparable experiments, from experiences of some experts or from existing theories.

Bayes' rule provides the framework for combining prior information with sample data. Suppose that $f(D|\theta)$ is the probability model of the data D with parameter (vector) $\theta \in \Theta$ and $\pi(\theta)$ is the prior probability density function (pdf) on the parameter space Θ. The posterior distribution of θ given the data D, using Bayes' rule, is given by

$$f(\theta|D) = \frac{f(D|\theta)\pi(\theta)}{\int_{\theta\in\Theta} f(D|\theta)\pi(\theta)d\theta}.$$

The posterior mean $\theta^* = \int_{\theta\in\Theta} \theta f(\theta|D)d\theta$ is then a Bayesian estimate of the parameter θ.

While it is easy to write down the formula of the posterior mean θ^*, a closed form existed only in a few simple cases, such as normal sample with a normal prior. In practice, Markov chain Monte Carlo (MCMC) methods such as Gibbs samplers and Metropolis algorithms are often employed to evaluate the posterior mean. However, these algorithms are usually slow especially for large data sets, which makes the OLAP processing based on these algorithms infeasible. Furthermore, these MCMC algorithms requires using the complete data set. In many data mining applications such as mining stream data applications, we often encounter the difficulty of not having the complete set of data in advance. One-scan algorithms are required for such applications.

In this paper, we propose a compression scheme and its associated theory to support high-quality aggregation of Bayesian estimation in a multi-dimensional data space. In the proposed approach, we compress each data segment by retaining only the model parameters and a small amount of auxiliary measures. We then develop an aggregation formula that allows us to reconstruct the Bayesian estimation from partitioned segments with a small approximation error. We show that the Bayesian estimates and the aggregated Bayesian estimates are asymptotically equivalent.

2 Concepts and Problem Definition

We develop our theory and algorithms in the context of data cubes and OLAP. In this section, we introduce the basic concepts related to data cubes and define our research problem.

2.1 Data cubes

Data cubes and OLAP tools are based on a multidimensional data model. The model views data in the form of a data cube. A *data cube* is defined by dimensions and facts. Dimensions are the perspectives or entities with respect to which an organization wants to keep records. Usually each dimension has multiple levels of abstraction formed by conceptual hierarchies. For example, country, state, city, and street are four levels of abstraction in a dimension for location.

To perform multidimensional, multi-level analysis, we need to introduce some basic terms related to data cubes. Let $\mathcal{D}$ be a relational table, called the base table, of a given cube. The set of all *attributes* $\mathcal{A}$ in $\mathcal{D}$ are partitioned into two subsets, the *dimensional attributes* DIM and the *measure attributes* M (so $DIM \cup M = \mathcal{A}$ and $DIM \cap M = \emptyset$). The measure attributes depend on the dimensional attributes in $\mathcal{D}$ and are defined in the context of data cube using some typical aggregate functions, such as count(), sum(), avg(), or some Bayesian analysis related measures to be studied here.

A tuple with schema $\mathcal{A}$ in a multi-dimensional data cube space is called a **cell**. Given three distinct cells c_1, c_2 and c_3, c_1 is an **ancestor** of c_2, and c_2 a **descendant** of c_1 if on every dimensional attribute, either c_1 and c_2 share the same value, or c_1's value is a generalized value of c_2's in the dimension's concept hierarchy.

A tuple $c \in \mathcal{D}$ is called a **base cell**. A base cell does not have any descendant. A cell c is an **aggregated cell** if it is an ancestor of some base cells. For each aggregated cell, the values of its measure attributes are derived from the set of its descendant cells.

2.2 Aggregation and classification of data cube measures

A data cube measure is a numerical or categorical quantity that can be evaluated at each cell in the data cube space. A measure value is computed for a given cell by aggregating the data corresponding to the respective dimension-value pairs defining the given cell. Measures can be classified into several categories based on the difficulty of aggregation. 1) An aggregate function is **distributive** if it can be computed in a distributed manner as follows. Suppose the data is partitioned into n sets. The computation of the function on each partition derives one aggregate value. If the result derived by applying the function to the n aggregate values is the same as that derived by applying the function on all the data without partitioning, the function can be computed in a distributive manner. For example, count() can be computed for a data cube by first partitioning the cube into a set of subcubes, computing count() for each subcube, and then summing up the counts obtained for each subcube. Hence, count() is a distributive aggregate function. For the same reason, sum(), min(), and max() are distributive aggregate functions. 2) An aggregate function is **algebraic** if it can be computed by an algebraic function with several arguments, each of which is obtained by applying a distributive aggregate function. For example, avg() (average) can be computed by sum()/count() where both sum() and count() are distributive aggregate functions. min_N(), max_N() and stand_dev() are algebraic aggregate functions. 3) An aggregate function is **holistic** if there is no constant bound on the storage size needed to describe a sub-aggregate. That is, there does not exist an algebraic function with M arguments (where M is a constant) that characterize the computation. Common examples of holistic functions include median(), mode(), and rank().

Except some simple special cases like normal sample with normal prior, Bayesian estimates seems to be holistic measures because they require the information of all the data points in an aggregated cell in order to calculate the Bayesian estimates. In general, Bayesian estimates are compressible measures [5, 23]. An aggregation function is **compressible** if it can be computed by a procedure with a number of arguments from lower level cells, and the number of arguments is *independent* of the number of tuples in the data cell. In other words, for compressible aggregate functions, we can compress each cell, regardless of its size (i.e., the number of tuples), into a constant number of arguments, and aggregate the function based on the compressed representation. The data compression technique should satisfy the following requirements: (1) the compressed data should support efficient lossless or asymptotically lossless aggregation of regression measures in a multidimensional data cube environment; and (2) the space complexity of compressed data should be low and be independent of the number of tuples in each cell, as the number of tuples in each cell may be huge.

In this paper, we will show that Bayesian estimates are compressible measures. Especially, the compression scheme developed in this paper for Bayesian estimates can support asymptotically lossless

aggregation. Therefore, the compression and aggregation scheme in this paper is more similar to the compression and aggregation scheme for logistic regression [23] than for linear regression [5].

3 Bayesian Statistics

Suppose that $x_1, \cdots, x_n$ are n observations from some probability model $f(x|\theta)$, where $\theta \in \Theta$ is the parameter (vector) of the probability model $f(x|\theta)$. The prior information in Bayesian statistics is given by a prior distribution $\pi(\theta)$ on the parameter space Θ. Then, under the independence assumption of the observations $x_1, \cdots, x_n$ given the parameter θ, the posterior distribution, $f(\theta|x_1, \cdots, x_n)$, of the parameter θ can be calculated using Bayes' rule,

$$\begin{aligned} f(\theta|x_1, \cdots, x_n) &= \frac{f(x_1, \cdots, x_n|\theta)\pi(\theta)}{\int_{\theta\in\Theta} f(x_1, \cdots, x_n|\theta)\pi(\theta)d\theta} \\ &= \frac{\prod_{i=1}^n f(x_i|\theta)\pi(\theta)}{\int_{\theta\in\Theta} \prod_{i=1}^n f(x_i|\theta)\pi(\theta)d\theta}, \end{aligned} \tag{1}$$

where $f(x_1, \cdots, x_n|\theta)$ is the joint distribution of $x_1, \cdots, x_n$ given the parameter θ.

Then, we could use the posterior mean θ_n^* as an estimate of the parameter θ, i.e.

$$\begin{aligned} \theta_n^* &= \int_{\theta\in\Theta} \theta f(\theta|x_1, \cdots, x_n)d\theta \\ &= \left(\int_{\theta\in\Theta} \prod_{i=1}^n f(x_i|\theta)\pi(\theta)d\theta\right)^{-1} \int_{\theta\in\Theta} \theta \prod_{i=1}^n f(x_i|\theta)\pi(\theta)d\theta. \end{aligned} \tag{2}$$

While it is easy to write down the formula (2) for the posterior mean, a closed form existed only in a few simple cases. MCMC methods such as Gibbs samplers and Metropolis-Hastings algorithms are often employed to evaluate the formula (2). These algorithms are based on constructing a Markov chain that has the posterior distribution (1) as its equilibrium distribution. After running the Markov chain a large number of steps, called burn-in steps, a sample from the Markov chain could be viewed as a sample from the posterior distribution (1). We then can approximate the posterior mean θ_n^* with any accuracy we wish by taking a large enough sample from the posterior distribution (1).

We consider the following example [17, 8, 22] to illustrate the algorithm of the Gibbs sampler.

Example 1: 197 animals are distributed multinomially into four categories and the observed data are $y = (y_1, y_2, y_3, y_4) = (125, 18, 20, 34)$. A genetic model specifies cell probabilities

$$\left(\frac{1}{2} + \frac{\theta}{4},\ \frac{1-\theta}{4},\ \frac{1-\theta}{4},\ \frac{\theta}{4}\right).$$

Assume that the prior distribution is Beta$(1, 1)$, which is also the uniform distribution on the interval $(0, 1)$ and therefore is a non-informative prior. The posterior distribution of θ is

$$f(\theta|y) \propto\ (2+\theta)^{y_1}(1-\theta)^{y_2+y_3}\theta^{y_4}$$

It is difficult, though not impossible, to calculate the posterior mean. However, a Gibbs sampler can be easily developed by augmenting the data y. Specifically, let $x = (x_1, x_2, x_3, x_4, x_5)$ such that

$y_1 = x_1 + x_2$, $y_2 = x_3$, $y_3 = x_4$ and $y_4 = x_5$. Assume the cell probabilities for x is

$$\left(\frac{1}{2},\ \frac{\theta}{4},\ \frac{1-\theta}{4},\ \frac{1-\theta}{4},\ \frac{\theta}{4}\right).$$

Then the distribution of y is the marginal distribution of x. The full conditional distribution of θ is $f(\theta|x_2, y) \propto \theta^{x_2+y_4}(1-\theta)^{y_2+y_3}$, which is Beta$(x_2+y_4+1, y_2+y_3+1)$. The full conditional distribution of x_2 is $f(x_2|y, \theta) \propto (2/(2+\theta))^{y_1-x_2}(\theta/(2+\theta))^{x_2}$, i.e. the binomial distribution Binom$(y_1, \theta/(2+\theta))$. The Gibbs sampler starts with any value $\theta^{(0)} \in (0, 1)$ and iterates the following two steps.

1. Generate $x_2^{(k)}$ from the full conditional distribution $f(x_2|y, \theta^{(k-1)})$, i.e. from Binom$(125, \theta^{(k-1)}/(2+\theta^{(k-1)}))$.

2. Generate $\theta^{(k)}$ from the full conditional distribution $f(\theta|x_2, y)$, i.e. from Beta$(x_2 + 35, 39)$.

Then, we could take average over $\theta^{(b+s)}, \cdots, \theta^{(b+sn)}$ to get an estimate of θ, where b is a large positive integer and s is a positive integer. The first b iterations are often called burn-in iterations and b is usually chosen large enough such that the Markov chain converges after b iterations. When n is large enough, this average will be a very good approximation to the posterior mean. The integer s is to reduce the correlation between two successive samples and is usually chosen to be small. Set $\theta^{(0)} = 0.5$, $s = 1$, $b = 1000$ and $n = 5,000$. The sample average we got is 0.622646 and is very close to the true posterior mean 0.622806. □

4 Aggregation of Bayesian Estimation

Since the computation of the Bayesian estimates θ_n^* involves the integral (2) and MCMC methods are often used, the compression and aggregation of Bayesian estimation are more difficult compared to the OLS estimate of regression coefficients in linear regression model. In general, it is very difficult to achieve lossless compression for the Bayesian estimation and we have to resort to the asymptotic theory of Bayesian estimations to get the asymptotic lossless compression.

We first review the notion asymptotically lossless compression representation (ALCR) introduced in [23].

Definition 4.1 In data cube analysis, a **cell function** g is a function that takes the data records of any cell with an arbitrary size as inputs and maps into a fixed-length vector as an output. That is:

$$g(c) = \mathbf{v},\ \text{for any data cell } c \tag{3}$$

where the output vector $\mathbf{v}$ has a fixed size.

Suppose that we have a probability model $f(x|\theta)$, where x are attributes and θ is the parameter of the probability model. Suppose c_a is a cell aggregated from the component cells $c_1, \cdots, c_k$. We define a cell function g_2 to obtain $m_i = g_2(c_i)$, $i = 1, \ldots, k$ and use an aggregation function g_1 to obtain an estimate of the parameter θ for c_a by

$$\tilde{\theta} = g_1(m_1, \cdots, m_k). \tag{4}$$

We say $\hat{\theta}$, an estimate of θ, is an asymptotically losslessly compressible measure if we can find an aggregation function g_1 and a cell function g_2 such that
a) the difference between $\tilde{\theta} = g_1(m_1, \cdots, m_k)$ and $\hat{\theta}(c_a)$ tends to zero in probability as the number of tuples in c_a goes to infinity, where $m_i = g_2(c_i)$, $i = 1, \ldots, k$;
b) $\hat{\theta}(c_a) = g_1(g_2(c_a))$; and
c) the dimension of m_i is independent of the number of tuples in c_i.

The measures m_i are called ALCR of the cell $c_i, i = 1, \cdots k$. In the following, we develop an ALCR for Bayesian estimation (2) based on its asymptotic property. We show that the asymptotic distributions of the estimates obtained from aggregation of the ALCR for each component cell and the Bayesian estimates in the aggregated cell are the same and further show that the the difference between them approaches zero in probability as the number of tuples in c_a goes to infinity. Further, the space complexity of ALCR is independent of the number of tuples. Therefore, the Bayesian estimations are asymptotically losslessly compressible measures.

4.1 Compression and Aggregation Scheme

Consider aggregating K cells at a lower level into one aggregated cell at a higher level. Suppose that there are n_k observations in the k^{th} component cell c_k. Denote $\{x_{k,1}, \cdots, x_{k,n_k}\}$ are the observations in the component cell c_k. Note that the observations $x_{k,j}$ $(j = 1, \cdots, n_k)$ could be multidimensional. Based on the observations in the k^{th} component cell c_k, we have the Bayesian estimate

$$\theta^*_{k,n_k} = \left(\int_{\theta \in \Theta} \prod_{j=1}^{n_k} f(x_{k,j}|\theta)\pi(\theta)d\theta \right)^{-1} \int_{\theta \in \Theta} \theta \prod_{j=1}^{n_k} f(x_{k,j}|\theta)\pi(\theta)d\theta. \tag{5}$$

We propose the following asymptotically lossless compression technique for the Bayesian estimation

- **Compression into ALCR**. For each *base* cell c_k, $k = 1, \cdots, K$, at the lowest level of the data cube, calculate the Bayesian estimate θ^*_{k,n_k} using (5). Save

  $$\text{ALCR}=(\theta^*_{k,n_k}, n_k)$$

  in each component cell c_k.

- **Aggregation of ALCR**. Calculate the aggregated ALCR $(\tilde{\theta}_a, n_a)$ using the following formula

  $$n_a = \sum_{k=1}^{K} n_k, \qquad \tilde{\theta}_a = n_a^{-1} \sum_{k=1}^{K} n_k \theta^*_{k,n_k}$$

  Such a process can be used to aggregate base cells at the lowest level as well as cells at intermediate levels. But for any non-base cell, $\tilde{\theta}$ is used in place of θ^*_{k,n_k} in its ALCR.

4.2 Compressibility of Bayesian Estimations

We now show that $(\tilde{\theta}_a, n_a)$ is an ALCR. We denote the Bayesian estimate for the aggregated cell to be θ^*_a and the corresponding estimates derived from ALCR compression and aggregation to be $\tilde{\theta}_a$. We

will show that the asymptotic distributions of θ_a^* and $\tilde{\theta}_a$ are the same and their difference tends to zero in probability.

Suppose that $\Theta \subset \mathbb{R}^p$ is an open subset of $\mathbb{R}^p$. For simplicity, we will only give the theorem in the case of $p = 1$ and multidimensional extension follows naturally. We make the following regularity assumptions on $f_\theta(\cdot) = f(\cdot|\ \theta)$ before giving the main theorem.

(C1) $\{x :\ f_\theta(x) > 0\}$ is the same for all $\theta \in \Theta$.

(C2) $L(\theta, x) = \log\ f_\theta(x)$ is thrice differentiable with respect to θ in a neighborhood, $U_{\delta_0}(\theta_0) = (\theta_0 - \delta_0, \theta_0 + \delta_0)$, of $\theta_0 \in \Theta$. If L', L'' and $L^{(3)}$ stand for the first, second and third derivatives, then $E_{\theta_0}|L'(\theta_0, X)|$ and $E_{\theta_0}|L''(\theta_0, X)|$ are both finite and

$$\sup_{\theta \in U_{\delta_0}(\theta_0)} |L^{(3)}(\theta, x)| \le M(x) \ \text{ and } \ E_{\theta_0} M(X) < \infty.$$

(C3) Interchange of the order of expectation with respect to θ_0 and differentiation at θ_0 are justified, so that $E_{\theta_0} L'(\theta_0, X) = 0$ and $E_{\theta_0} L''(\theta_0, X) = -E_{\theta_0}[L'(\theta_0, X)]^2$.

(C4) $I_{\theta_0} = E_{\theta_0}[L'(\theta_0, X)]^2 > 0$.

(C5) If $X_1, \cdots, X_n$ are random variables from f_{θ_0} and $L_n(\theta) = \sum_{i=1}^n L(\theta, X_i)$, then for any $\delta > 0$, there exists an $\varepsilon > 0$ such that $P_{\theta_0}\{\sup_{|\theta - \theta_0| > \delta}[L_n(\theta) - L_n(\theta_0)] \le -\varepsilon\} \to 1$.

(C6) The prior has a density $\pi(\theta)$ with respect to Lebesgue measure, which is continuous and positive at θ_0. Furthermore, $\pi(\theta)$ satisfies $\int_{\theta \in \Theta} |\theta| \pi(\theta) d\theta \le \infty$.

These conditions guarantee the consistency and asymptotic normality of the posterior mean and are the same as the conditions in [9]. Please see [9] and [13] for other sets of conditions that can guarantee the consistency and asymptotic normality of the posterior mean.

Theorem 1 *Suppose $\{f_\theta|\ \theta \in \Theta\}$ satisfies Condition (C1)-(C5) and the prior distribution satisfies Condition (C6). Let $X_{1,k}, \cdots, X_{1,n_k}$ $(k = 1, \cdots, K)$ be random variables from the distribution f_{θ_0}, θ_{k,n_k}^* be the posterior mean (2) based on the random variables $X_{1,k}, \cdots, X_{1,n_k}$ and $\tilde{\theta}_a = n_a^{-1} \sum_{k=1}^K n_k \theta_{k,n_k}^*$ be the aggregated Bayesian estimate. Then we have*

$$\sqrt{n_a}(\tilde{\theta}_a - \theta_0) \xrightarrow{d} N(0, I_{\theta_0}^{-1}) \ \ \textit{as} \ \ m_K = \min\{n_1, \cdots, n_K\} \to \infty.$$

Proof. Since $\{f_\theta|\ \theta \in \Theta\}$ and $\pi(\theta)$ satisfy Condition (C1)-(C6), from Theorem 1.4.3 in [9] we have

$$\sqrt{n_k}(\theta_{k,n_k}^* - \theta_0) \xrightarrow{d} N(0, I_{\theta_0}^{-1}) \ \text{ as } \ n_k \to \infty.$$

Let $Z_{k,n_k} = \sqrt{n_k}(\theta_{k,n_k}^* - \theta_0)$ and $\phi_{k,n_k}(t) = E[e^{itZ_{k,n_k}}]$ be its characteristic function. Denote $v^2 = I_{\theta_0}^{-1}$. Then, by Levy's Continuity Theorem (see, for example, [7] and [21] among others), we have $\phi_{k,n_k}(t)$ converges to $\exp(-v^2t^2/2)$ uniformly in every finite interval, where $\exp(-v^2t^2/2)$ is the characteristic

function of the normal distribution $N(0, v^2)$. On the other hand, the characteristic function of the random variable $Z_{n_a} = \sqrt{n_a}(\tilde{\theta}_a - \theta_0)$ is

$$\begin{aligned}
\phi_{n_a}(t) &= E[\exp\{it\sqrt{n_a}(\tilde{\theta}_a - \theta_0)\}] \\
&= E[\exp\{it\sqrt{n_a}n_a^{-1}\sum_{k=1}^{K} n_k(\theta^*_{k,n_k} - \theta_0)\}] \\
&= \prod_{k=1}^{K} E[\exp\{itn_a^{-1/2}n_k(\theta^*_{k,n_k} - \theta_0)\}] \\
&= \prod_{k=1}^{K} \phi_{k,n_k}(n_k^{1/2}n_a^{-1/2}t).
\end{aligned}$$

Then, we have

$$\begin{aligned}
|\log[\phi_{n_a}(t)] + v^2t^2/2| &= \left|\sum_{k=1}^{K}\{\log[\phi_{k,n_k}(n_k^{1/2}n_a^{-1/2}t)] + \frac{n_k}{2n_a}v^2t^2\}\right| \\
&\leq \sum_{k=1}^{K}\left|\log[\phi_{k,n_k}(n_k^{1/2}n_a^{-1/2}t)] + \frac{1}{2}v^2\big(n_k^{1/2}n_a^{-1/2}t\big)^2\right|
\end{aligned}$$

Since $\phi_{k,n_k}(t)$ converges to $\exp(-v^2t^2/2)$ uniformly in every finite interval, $\log[\phi_{k,n_k}(t)]$ will converge to $-v^2t^2/2$ uniformly in every finite interval. Then for any $\varepsilon > 0$, there exists an $N_k(\varepsilon) > 0$ such that when $n_k > N_k(\varepsilon)$, we have $|\log[\phi_{k,n_k}(\tau)] + v^2\tau^2/2| \leq \varepsilon/K$ for all $|\tau| \leq |t|$. Take $M_K(\varepsilon) = \max\{N_1(\varepsilon), \cdots, N_K(\varepsilon)\}$. Since $|n_k^{1/2}n_a^{-1/2}t| \leq |t|$, we have

$$|\log[\phi_{n_a}(t)] + v^2t^2/2| \leq \sum_{k=1}^{K}\varepsilon/K = \varepsilon$$

for $m_K \geq M_K(\varepsilon)$. Therefore, $\phi_{n_a}(t)$ converges to $\exp(-v^2t^2/2)$ for all $t \in \mathbb{R}$ and so Theorem 1 was proved by Levy's Continuity Theorem again. □

From Theorem 1, the aggregated Bayesian estimate $\tilde{\theta}_{n_a}$ achieves the same efficiency as the Bayesian estimate $\theta^*_{n_a}$, i.e. their asymptotic variances are the same. Theorem 1 holds for multidimensional case, but conditions in Theorem 1 need modification accordingly. For example, Condition (C2) and (C3) would be about the partial derivatives with respect to the component of the parameter vector θ, and Condition (C4) should change to "the Fisher information I_{θ_0} is a positive definite matrix".

Corollary 1 *Under the conditions of Theorem 1, the difference between the estimates $\theta^*_{n_a}$ and $\tilde{\theta}_{n_a}$ approaches 0 in probability.*

Proof. From Theorem 1, $\tilde{\theta}_{n_a}$ approaches θ_0 in probability as m_k goes to infinity. The Bayesian estimate $\theta^*_{n_a}$ also approaches θ_0 in probability. Therefore, the difference between $\theta^*_{n_a}$ and $\tilde{\theta}_{n_a}$ converges to 0 in probability. □

Corollary 1 means that the difference between $\theta^*_{n_a}$ and $\tilde{\theta}_{n_a}$ will become smaller and smaller as we have more and more data. Henceforth, the estimate $\tilde{\theta}_{n_a}$ is a good approximation to $\theta^*_{n_a}$ when the data set is large.

4.3 Detection of Non-homogeneous Data

Theorem 1 and Corollary 1 rely on the assumption that the data from different subcubes come from the same probability model, i.e. the data are homogeneous. Aggregation of non-homogenous data can lead to misleading results and Simpson's paradox [1] may occur. Therefore, it is important to develop tools of testing non-homogeneity. The test of non-homogeneity should be able to support the OLAP analysis and hence it should only depend on the compressed measures or the ALCRs of subcubes. The ALCR defined in Section 4.1 is insufficient for the test of non-homogeneity and one additional measure is needed. Let v_{k,n_k} be the posterior variance matrix based on the observations in the k^{th} component cells, i.e.

$$v_{k,n_k} = \Big(\int_{\theta\in\Theta} \prod_{j=1}^{n_k} f(x_{k,j}|\theta)\pi(\theta)d\theta \Big)^{-1} \int_{\theta\in\Theta} (\theta - \theta^*_{k,n_k})(\theta - \theta^*_{k,n_k})^T \prod_{j=1}^{n_k} f(x_{k,j}|\theta)\pi(\theta)d\theta. \qquad (6)$$

If the parameter θ is of p-dimensional, the measure v_{k,n_k} is a $p \times p$ matrix. We propose the following modified compression and aggregation scheme.

- **Compression into ALCR**. For each *base* cell c_k, $k = 1, \cdots, K$, at the lowest level of the data cube, calculate the Bayesian estimate θ^*_{k,n_k} using (5) and the posterior variance v_{k,n_k} using (6). Save

  $$\text{ALCR}=(\theta^*_{k,n_k}, v_{k,n_k}, n_k)$$

  in each component cell c_k.

- **Aggregation of ALCR**. Calculate the aggregated ALCR $(\tilde{\theta}_a, \tilde{v}_a, n_a)$ using the following formula

  $$n_a = \sum_{k=1}^{K} n_k, \quad \tilde{\theta}_a = n_a^{-1} \sum_{k=1}^{K} n_k \theta^*_{k,n_k}, \quad \tilde{v}_a = n_a^{-2} \sum_{k=1}^{K} n_k^2 v_{k,n_k}$$

  For any non-base cell, $\tilde{\theta}$ and $\tilde{v}_a$ is used in place of θ^*_{k,n_k} and v_{k,n_k} in its ALCR.

Suppose that c_1 and c_2 are two subcubes and $(\tilde{\theta}_1, \tilde{v}_1, n_1)$ and $(\tilde{\theta}_2, \tilde{v}_2, n_2)$ are their ALCRs respectively. By Theorem 1, $\sqrt{n_k}(\tilde{\theta}_k - \theta_0)$ approximately follows the normal distribution $N(0, I_{\theta_0}^{-1})$, or $\tilde{\theta}_k - \theta_0$ $(k = 1, 2)$ approximately follows the normal distribution $N(0, n_k^{-1} I_{\theta_0}^{-1})$ $(k = 1, 2)$. Use $\tilde{v}_k$ as the estimate of $n_k^{-1} I_{\theta_0}^{-1}$, and it follows that $t = (\tilde{\theta}_1 - \tilde{\theta}_2)^T (\tilde{v}_1 + \tilde{v}_2)^{-1} (\tilde{\theta}_1 - \tilde{\theta}_2)$ approximately follows χ^2_p distribution. Then, we use the statistic t to test the non-homogeneity.

We use the kidney stone data as considered in [23] as an example of the test of non-homogeneity. The data are from a medical study [3, 12] comparing the success rates of two treatments for kidney stones. The two treatments are open surgery (treatment A) and percutaneous nephrolithotomy (treatment B). Table 1 shows the effects of both treatments under different conditions. It reveals that treatment A has a higher success rates than treatment B for both small stone and large stone groups. However, after aggregating over the two groups, treatment A has a lower success rate than treatment B.

Let S be the binary random variable that indicate whether the treatment succeeds or not, and T be the type of treatment that a patient receives. Denote by p_A and p_B be the success rate of treatment A

Table 1: Success rates for different groups of stone size.

	Treatment A	Treatment B
Small Stone	93%(81/87)	87%(234/270)
Large Stone	73%(192/263)	69%(55/80)
Both	78%(273/350)	83%(289/350)

and B, and α_A be the probability that a patient receives treatment A. Then, we have the probability model

$$Pr(S,T|p_A,p_B,\alpha_A) = \left[p_A^S(1-p_A)^{1-S}\alpha_A\right]^{I(T=A)}\left[p_B^S(1-p_B)^{1-S}(1-\alpha_A)\right]^{I(T=B)},$$

where $I(\cdot)$ is the indicator function. Set priors for p_A, p_B, α_A as the noninformative prior Beta$(1,1)$. Given observations $D = \{(s_1,t_1),\cdots,(s_n,t_n)\}$, the posterior distribution of (p_A,p_B,α_A) is

$$f(p_A,p_B,\alpha_A|D) \propto p_A^{n_{As}}(1-p_A)^{n_{Af}}p_B^{n_{Bs}}(1-p_B)^{n_{Bf}}\alpha_A^{n_A}(1-\alpha_A)^{n_B},$$

where $n_{As} = \sum_{i=1}^n s_i I(t_i = A)$, $n_{Af} = \sum_{i=1}^n (1-s_i) I(t_i = A)$, $n_{Bs} = \sum_{i=1}^n s_i I(t_i = B)$, $n_{Bf} = \sum_{i=1}^n (1-s_i) I(t_i = B)$, $n_A = \sum_{i=1}^n I(t_i = A)$ and $n_B = \sum_{i=1}^n I(t_i = B)$. Therefore, the posterior distribution is the product of three independent Beta distributions.

Denote $\theta = (p_A, p_B, \alpha_A)$. Based on the small stone group data, the Bayesian estimate is $\theta_s^* = (0.92, 0.86, 0.24)$ and the corresponding posterior variance is $v_s = diag\{0.00080, 0.00043, 0.00051\}$. Based on the large stone group, the Bayesian estimate is $\theta_l^* = (0.73, 0.68, 0.77)$ and the corresponding posterior variance is $v_l = diag\{0.00075, 0.0026, 0.00052\}$. Using these results, the test statistic is $t = (\theta_s^* - \theta_l^*)^T(v_s + v_l)^{-1}(\theta_s^* - \theta_l^*) = 296.63$, which is highly significant. Therefore, it is highly possible that the two data sets are inhomogeneous and we should not aggregate them together.

5 Experimental Studies

We perform experimental studies on synthetic data to validate our method. The Bayesian model under consideration is the mixture of transition models. Mixture of transition models have been used to model user visiting website [2, 18, 19], unsupervised training of robots [16] and the dynamics of a military scenario [20].

Transition models are useful in describing time series which have only finite number of states. The observations of Transition models are finite state Markov chains of finite length. For example, the sequence (A, B, A, C, B, B, C) could be a realization of a 3-state first order Markov chain, where the transition probability at time t only depends on the state of the Markov chain at time t but not on the previous history. If all the observations are realizations from the same transition model, one can readily get a closed form of the posterior mean of the parameters. However, the set of sequences may be heterogeneous and the sequences may come from several different transition models, in which case the mixture of transition models is useful in estimating the transition matrices and clustering the observational sequences.

Consider a data set of N sequences, $D = \{x_1, \cdots, x_N\}$, that are realizations from some s-state discrete first order Markov processes. The sequences are possibly of different length. Assume that each

sequence comes from one of m transition models. Let ${}_{(l)}P_{ij}$ be the (i,j) element of the lth probability transition matrix, or the transition probability from state i to state j for a process in cluster l. Let ${}_{(l)}p_i$ be the ith element of the initial state distribution of processes from cluster l. Further assume that α_l is the probability that a process is from cluster l. Denote x_k^0 as the initial state of the sequence x_k and $n_{ij}^{(k)}$ be the number of times the process x_k transitioned from state i to state j. Then the mixture of transition model is

$$f(x_k|\theta) = \sum_{l=1}^{m} \alpha_l \prod_{i=1}^{s} {}_{(l)}p_i^{I(x_k^0=i)} \prod_{i=1}^{s}\prod_{j=1}^{s} {}_{(l)}P_{ij}^{n_{ij}^{(k)}},$$

where θ is the parameter vector consisting of ${}_{(l)}P_{ij}$, ${}_{(l)}p_i$ and α_l as its elements, and $I(\cdot)$ is the indicator function. The prior distribution for the parameter vectors $\alpha = (\alpha_1, \cdots, \alpha_m)$, ${}_{(l)}p = ({}_{(l)}p_1, \cdots, {}_{(l)}p_s)$ and ${}_{(l)}P_{i\cdot} = ({}_{(l)}P_{i1}, \cdots, {}_{(l)}P_{is})$ are Dirichlet priors with all parameters as 1. The Dirichlet priors used here are non-informative priors. The posterior mean has no closed form for this Bayesian model. However, by introducing a "missing data" $\delta_l^{(k)}$, the 0/1 unobserved indicator that process k belongs to cluster l, one can readily develop a Gibbs sampler [18, 19].

5.1 Quality of compression and aggregation scheme

We perform simulation studies for the mixture of transition models. In the simulation, the number of clusters are set as three and the Markov chains are 2-state chains. We generated $10,000$ chains from the mixture of transition model and each chain is of length 30. The underlying true parameters are set as

1. initial probabilities:

$$ {}_{(1)}p = (0.2,\ 0.8), \quad {}_{(2)}p = (0.9,\ 0.1), \quad {}_{(3)}p = (0.4,\ 0.6); $$

2. transition matrices

$$ {}_{(1)}P = \begin{pmatrix} 0.9 & 0.1 \\ 0.8 & 0.2 \end{pmatrix}, \quad {}_{(2)}P = \begin{pmatrix} 0.3 & 0.7 \\ 0.1 & 0.9 \end{pmatrix}, \quad {}_{(3)}P = \begin{pmatrix} 0.7 & 0.3 \\ 0.9 & 0.1 \end{pmatrix} $$

3. the probability vector $\alpha = (0.2,\ 0.5,\ 0.3)$.

We partition the entire data set into $K = 1, 10, 20 \cdots, 100$ cells with equal number of observations and then use the aggregation algorithm to approximate the posterior mean for the entire data set. We run the Gibbs sampler $11,000$ iterations and set the burn-in iterations as $1,000$. Note that the estimate corresponding to $K = 1$ is just the posterior mean. Suppose ${}_{(l)}\tilde{p}$, ${}_{(l)}\tilde{P}$ and ${}_{(l)}\tilde{\alpha}$ be the estimates of ${}_{(l)}p$, ${}_{(l)}P$ and α $(l = 1, 2, 3)$. Then we define the maximum absolute deviances (MAD) as $D(\tilde{p}, p) = \max\{|{}_{(l)}\tilde{p}_i - {}_{(l)}p_i| :\ l = 1, 2, 3,\ i = 1, 2\}$, $D(\tilde{P}, P) = \max\{|{}_{(l)}\tilde{P}_{ij} - {}_{(l)}P_{ij}| :\ l = 1, 2, 3,\ i, j = 1, 2\}$ and $D(\tilde{\alpha}, \alpha) = \max\{|\tilde{\alpha}_l - \alpha_l| :\ l = 1, 2, 3\}$. Figure 1 shows the MAD of the aggregated estimates from different partitions. The solid line is for the maximum absolute deviance $D(\tilde{p}, p)$, the blue dashed line is for $D(\tilde{P}, P)$ and the red dotted line is for $D(\tilde{\alpha}, \alpha)$. It is clear that estimates from all different partitions behave alike and they are all close to the underlying true parameter. The simulation study shows that the accuracy of the aggregated estimates is as good as the accuracy of the original Bayesian estimates.

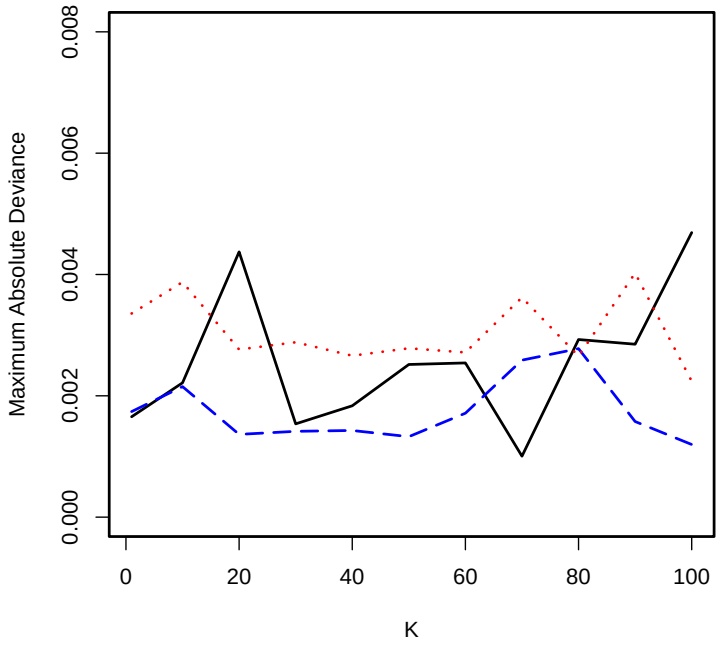


Figure 1: Maximum absolute deviance of the aggregated estimates with a varying number of partitions

5.2 Simulation study in data cube

In this simulation, we study the efficiency and quality of the compression and aggregation scheme for aggregated cells in data cube. The Bayesian model under consideration is again the mixture of transition model. Two dimensions are time and location. We have 20 months' records in time dimension and 50 states in location dimension. In practice, the data could be the records of a website that records user's visit to the website. The location dimension could just be the IP address of the user. For each state in each month, we have 500 observations, i.e. we have 500 users' records. Hence, we have 500,000 observations in total. The observations are sequences that record users' visiting path in the website. As in Section 5.1, the number of clusters are set as three and the Markov chains are 2-state chains. The underlying true parameters are also set as in Section 5.1.

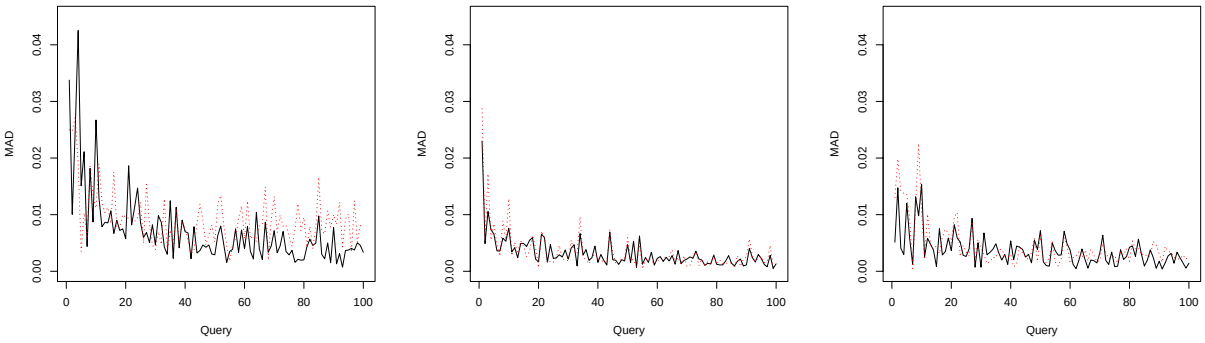
We compare our ALCR method to the direct Bayesian estimation method, which directly uses raw data to calculate the Bayesian estimates of the parameters, by comparing their computing time for handling 100 randomly generated queries. To save the computing time of the direct Bayesian estimation method, the aggregated cells that the queries ask for can have at most 200 base cells. More specifically, More specifically, to generate a query, we first randomly select a number D from $\{1, \cdots, 200\}$, and then we randomly select D cells from the 1000 base cells (t_i, l_j) $(i = 1, \cdots, 20,$ $j = 1, \cdots, 50)$. For simplicity, we can the number D the size of the query. We run Gibbs samplers $6,000$ iterations and the buin-in iterations are set as $1,000$.

Table 2 shows the time with and without using compression, respectively. The first row shows the computational time for compression and the second row shows the aggregation time for all these 100

Table 2: Comparison of the computational time used for answering 100 queries.

	ACLR compression	no compression
Preprocessing (compression)	2,281 minutes	NA
Query processing (aggregation)	1 minute	22,858 minutes

Figure 2: Comparison of the ALCR method (red dotted lines) and direct method (dark solid lines).



(a) MAD for initial probabilities (b) MAD for transition matrices (c) MAD for probabilities α.

queries. Without using ACLR compression, the aggregation time is the time to compute Bayesian estimates directly from the raw data of these selected cells. It is obvious that our method saves huge amount of computational time when handling OLAP queries in a data cube.

Figure 2 compares MADs of estimates for each query from the ALCR method and the direct method. The red dotted lines are for the ALCR methods and the dark solid lines are for the direct method. Figure 2 (a), (b) and (c) are MADs of estimates for initial probabilities p, transition matrices P and probabilities α. The queries are ordered according to queries' sizes. Figure 2 shows that estimates from the ALCR method tend to have larger MAD than estimates from the direct method when the size of query is large, especially for the estimates of initial probabilities, but in general MADs for both methods are very close.

References

[1] A Agresti. *Categorical Data Analysis*. John Wiley and Sons, New Jersey, 2nd edition, 2002.

[2] I. Cadez, D. Heckerman, P. Smyth C. Meek, and S. White. Visualization of navigation patterns on a web site using model-based clustering. Technical report, Microsoft Research. MSR-TR-00-18.

[3] C. R. Charig, D. R. Webb, S. R. Payne, and O. E. Wickham. Comparison of treatment of renal calculi by operative surgery, percutaneous nephrolithotomy, and extracorporeal shock wave lithotripsy. *British Medical Journal*, 292:897–882, 1986.

[4] B. Chen, L. Chen, Y. Lin, and R Ramakrishnan. Prediction cubes. In *Proceedings of the 31st VLDB Conference*, pages 982–993, 2005.

[5] Y. Chen, G. Dong, J. Han, J. Pei, B. Wah, , and J. Wang. Regression cubes with lossless compression and aggregation. *IEEE Transactions on Knowledge and Data Engineering*, 18:1585–1599, 2006.

[6] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multi-dimensional regression analysis of time-series data streams. *Proc. Int. Conf. on Very Large Data Bases*, pages 323–334, 2002.

[7] K. L. Chung. *A Course in Probability Theory*. Elsevier, San Diego, California, 3rd edition, 2001.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Ser. B*, 39:1–38, 1977.

[9] J. K. Ghosh and R. V. Ramamoorthi. *Bayesian Nonprametrics*. Springer, New Jersey, 2002.

[10] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. *Data Mining and Knowledge Discovery*, 1:29–54, 1997.

[11] J. Han, Y. Chen, G. Dong, J. Pei, B. W. Wah, J. Wang, and Y. Cai. Stream cube: An architecture for multi-dimensional analysis of data streams. *Distributed and Parallel Databases*, 18(2):173–197, 2005.

[12] S. A. Julious and M. A. Mullee. Confounding and simpson's paradox. *British Medical Journal*, 309:1480–1481, 1994.

[13] E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer, New Jersey, 2nd edition, 1998.

[14] C. Liu, M. Zhang, M. Zheng, and Y. Chen. Step-by-step regression: A more efficient alternative for polynomial multiple linear regression in stream cube. In *Proc. the Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 437–448, 2003.

[15] S. Pang, S. Ozawa, and N. Kasabov. Incremental linear discriminant analysis for classification of data streams. *IEEE Trans. Sys. Man. Cybernetics, Part B*, 35(5):905–14, 2005.

[16] M. Ramoni, P. Sebastiani, and P. Cohen. Bayesian clustering by dynamics. *Machine Learning*, 47(1):99–121, 2002.

[17] C. R. Rao. *Linear Statistical Inference and Its Applications*. John Wiley, New York, 1973.

[18] G. Ridgeway. Finite discrete markov process clustering. Technical report, Microsoft Research. MSR-TR-97-24.

[19] G. Ridgeway and S. Altschuler. Clustering finite discrete markov chains. In *Proceedings of the Section on Physical and Engineering Sciences*, pages 228–229, 1998.

[20] P. Sebastiani, M. Ramonni, P. Cohen, J. Warwick, and J. Davis. Discovering dynamics using bayesian clustering. In *Advances in Intelligent Data Analysis*, Lecture Notes in Computer Science, pages 395–406. Springer, 1999.

[21] A. N. Shiryaev. *Probability*. Springer, New Jersey, 2nd edition, 1995.

[22] M. A. Tanner and W. H. Wong. The calculation of posterior distribution by data augmentation. *Journal of the American Statistical Association*, 82:528–540, 1987.

[23] R. Xi, N. Lin, and Y. Chen. Compression and aggregation for logistic regression analysis in data cubes. *IEEE Transactions on Knowledge and Data Engineering*, 2008. in press.