

Compression and Aggregation for Logistic Regression Analysis in Data Cubes

Ruibin Xi, Nan Lin, and Yixin Chen

Abstract—Logistic regression is an important technique for analyzing and predicting data with categorical attributes. In this paper, we consider supporting online analytical processing (OLAP) of logistic regression analysis for multi-dimensional data in a data cube where it is expensive in time and space to build logistic regression models for each cell from the raw data. We propose a novel scheme to compress the data in such a way that we can reconstruct logistic regression models to answer any OLAP query without accessing the raw data.

Based on a first-order approximation to the maximum likelihood estimating equations, we develop a compression scheme that compresses each base cell into a small compressed data block with essential information to support the aggregation of logistic regression models. Aggregation formulae for deriving high-level logistic regression models from lower level component cells are given. We prove that the compression is asymptotically lossless in the sense that the aggregated estimator deviates from the true model by an error that is bounded and approaches to zero when the data size increases. The results show that the proposed compression and aggregation scheme can make feasible OLAP of logistic regression in a data cube. Further, it supports real-time logistic regression analysis of stream data which can only be scanned once and cannot be permanently retained. Experimental results validate our theoretical analysis and demonstrate that our method can dramatically save time and space costs with almost no degradation of the modelling accuracy.

Index Terms—data cubes, online analytical processing, logistic regression, compression, aggregation

I. INTRODUCTION

LOGISTIC regression is an important statistical method for modelling and predicting categorical data. When we conduct logistic regression analysis in real-world data mining applications, we often encounter the difficulty of not having the complete set of data in advance. It is often demanded to recover logistic regression models of a large data set with access not to the raw data but to only sketchy information of divided chunks of the data set.

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 [13] 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, such as regression analysis, are desired to be supported in OLAP. Moreover, there are lots of applications with stream data generated continuously in a dynamic environment, with huge volume, infinite flow, and fast changing behavior. When collected, such data are almost always at a rather low level, consisting of various kinds of detailed temporal and other features. To find interesting patterns, it is necessary to perform statistical analysis at a higher and more meaningful abstraction level [8].

Recently, there has been active research on aggregating advanced statistical measures in multi-dimensional data cubes from partitioned subsets of data. Previous statistical measures studied under this paradigm include parametric models such as linear regression [9], [14], general multiple linear regression [7], [18], and predictive filters [7], as well as nonparametric statistical models such as naive Bayesian classifiers [5] and linear discriminant analysis [21]. Along this line, in this paper, we propose schemes to support logistic regression analysis in data cubes.

Example 1: Suppose a nation-wide bank wants to study the likelihood of customers to apply for a new credit card. Suppose that for each day, for each regional branch of the bank, there is a data set containing $(y_1, x_{11}, x_{12}), (y_2, x_{21}, x_{22}) \dots, (y_n, x_{n1}, x_{n2})$, where n is the number of customers, x_{i1} represents the age of the i^{th} customer, x_{i2} represents the account balance of the i^{th} customer, and y_i is a binary indicator of whether the customer applied for the new credit card (0 for no and 1 for yes). To model the relationship between credit card application and user information, the bank manager can assume that the probability of a customer applying for the new credit card, p , depends on the customer age x_1 and account balance x_2 as follows.

$$\text{logit}(p) = \log \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \quad (1)$$

The above model (1) is called a *logistic regression* model [1]. After the logit transformation, $\text{logit}(p)$ ranges over the entire real line and makes it reasonable to be modelled as a linear function of $\mathbf{x} = (x_1, x_2)^T$. The regression coefficients, $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^T$, are often estimated using maximum likelihood.

To perform multi-dimensional analysis, it may be required to aggregate the models along multiple dimensions. For example, in the *location* dimension, we may have computed a logistic regression model for each *city*. Now if we want to roll up to the *state* level along the *location* dimension, we want to compute the logistic regression model over the data set containing all the cities in the state. We can also aggregate

Manuscript received February 25, 2008; revised July 16, 2008.

R. Xi and N. Lin are with the Department of Mathematics, Washington University in St. Louis, One Brookings Drive, St. Louis, MO 63130.

Y. Chen is with the Computer Science Department, Washington University in St. Louis, One Brookings Dr, St. Louis, MO 63130, chen@cse.wustl.edu.

along other dimensions such as *time*.

A key issue for such aggregation operations is: can we generate the high-level models without accessing the raw data? Since computing the logistic regression model requires performing a nonlinear numerical optimization problem, solving such a problem from scratch over a large aggregated data set for each roll-up operation is computationally very expensive. It is far more desirable to derive high-level logistic regression models from low-level model parameters without accessing the raw data. It is also expected that such aggregation computation is faster than estimating the logistic regression model parameters from scratch. ■

In addition to data cubing and OLAP, mining data streams is another motivation for the proposed work. In analyzing real-time data streams, it is typically impossible to store the raw data due to space limitations. One-scan algorithms are required for such applications. Therefore, in order to *incrementally* perform logistic regression on data streams, we are required to be able to generate logistic regression models over two parts of data: the old data that are already in the system and the new chunk of data. The key challenge is that previous raw data cannot be retained. We need to compress the existing raw data in such a way that we can incrementally re-construct the logistic regression model as new stream data flow in.

In this paper, we propose a compression scheme and its associated theory to support high-quality aggregation of logistic regression models 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 logistic models from partitioned segments with a small approximation error. The error is theoretically bounded and asymptotically convergent to zero as the sample size grows.

The paper is organized as follows. We define the basic concepts and problem statement in Section 2. We review the basics of logistic regression in Section 3. Then, we present our main technical approach in Section 4. We present experimental results in Section 5, discuss related work in Section 6, and conclude the paper in Section 7.

II. CONCEPTS AND PROBLEM DEFINITION

We develop our theory and algorithms in the context of data cubes and OLAP, although it should be understood that our results can be used for other settings such as incremental mining of stream data. In this section, we introduce the basic concepts related to regression analysis in data cubes and define our research problem.

A. 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 regression related measures to be studied here.

Example 2: In Example 1, the dimensional attributes may include time, location, customer name, account balance, and the measure attribute can be an indicator on the application of the credit card. ■

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.

B. 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()`.

If we characterize the logistic regression measure using the above classification, it seems to be a holistic measure because it requires the information of all the data points in an aggregated cell in order to compute the regression model. It is impossible to compute the regression model distributively and compose the high-level regression model merely from the low-level models. Thus the requirement for multi-level, multidimensional online analysis of advanced statistical measures, though desirable, raises a challenging research issue: “*Is it feasible to perform logistic regression in OLAP on huge volumes of data since a data cube is usually much bigger than the original data set, and its construction may take multiple database scans?*”

Our main idea in this paper is to compress the raw data for each cell, store only a minimum number of measures that are just sufficient to support the on-line analysis, and compute high-level measures from the corresponding low-level cells without accessing the raw data. Such a compression technique has led to the definition of compressible measures [7]. 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.

Unlike distributive and algebraic measures which require only the measure itself or few other distributive measures to support the aggregation, compressible measures cannot be aggregated if only the measure itself and some distributive measures are stored in each data cell. Additional information derived from the raw data is needed to support lossless or asymptotically lossless aggregation of compressible measures.

In this paper, we will show that logistic regression model parameters are compressible measures.

C. Previous work on regression cubes

Previously, a framework called the regression cube has been developed [5] for OLAPing of linear regression models in data cubes. The regression cube uses a *nonlinear compression representation* (NCR) [7] to support ordinary least squares (OLS) estimation. NCR is a lossless compression technique that can be used in linear models, such as linear regression models and autoregressive filters, to compute the OLS estimates of parameters. It has been shown that OLS estimates are lossless compressible measures [7].

One important limitation of previous regression cubes is that the measures in linear regression have to be numerical and not categorical. In this paper, we develop a compression scheme to support OLAP for logistic regression analysis, which can handle categorical data.

Although linear models have been studied, it remains a challenge to develop similar compression techniques to support **nonlinear** models such as logistic regression in data cubes. Logistic regression is widely used to predict a discrete outcome, such as group membership, from a set of variables that may be continuous, discrete, dichotomous, or a mix of any of these.

III. LOGISTIC REGRESSION ANALYSIS

In this section, we review the basic definitions of logistic regression.

A. Logistic regression model

Suppose we have n independent observations $(y_1, \mathbf{x}_1), \dots, (y_n, \mathbf{x}_n)$, where y_i is a binary variable assumed to have a Bernoulli distribution with parameter $p_i = P(y_i = 1)$ and $\mathbf{x}_i \in \mathbb{R}^d$ are some explanatory variables. An intercept term can be easily included by setting the first element of \mathbf{x}_i to be 1. Fitting a linear regression model for a binary response can give predicted values beyond $(0, 1)$ that are theoretically inadmissible. Logistic regression models are widely used to model binary responses using the following formulation:

$$\log \frac{p_i}{1 - p_i} = \boldsymbol{\beta}^T \mathbf{x}_i, \quad (2)$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ are some unknown regression coefficients often estimated using maximum likelihood. The **maximum likelihood estimates (MLE)** $\hat{\boldsymbol{\beta}}$ are so chosen as to maximize the following **likelihood function**:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{1 - y_i} \quad (3)$$

Intuitively, we see that (3) represents the joint probability of the given observation sequence $\{y_1, \dots, y_n\}$, since $p_i^{y_i} (1 - p_i)^{1 - y_i}$ models the probability that y_i takes the observed value for each $i = 1, \dots, n$.

According to (2), we have:

$$p_i = \frac{\exp(\boldsymbol{\beta}^T \mathbf{x}_i)}{1 + \exp(\boldsymbol{\beta}^T \mathbf{x}_i)}. \quad (4)$$

Thus, the likelihood function can be written as:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^n [\mu(\boldsymbol{\beta}^T \mathbf{x}_i)]^{y_i} [1 - \mu(\boldsymbol{\beta}^T \mathbf{x}_i)]^{1 - y_i}, \quad (5)$$

where

$$\mu(t) = \frac{e^t}{1 + e^t}. \quad (6)$$

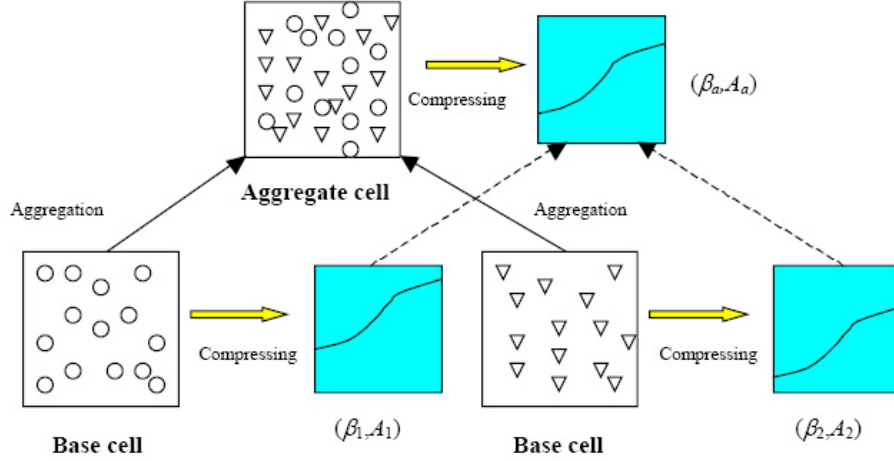
Maximizing $L(\boldsymbol{\beta})$ is equivalent to maximizing the **log-likelihood function**

$$l(\boldsymbol{\beta}) = \sum_{i=1}^n \left[y_i \boldsymbol{\beta}^T \mathbf{x}_i + \log(1 - \mu(\boldsymbol{\beta}^T \mathbf{x}_i)) \right]. \quad (7)$$

According to the optimality condition of unconstrained optimization, the parameter $\hat{\boldsymbol{\beta}}$ that maximizes $l(\boldsymbol{\beta})$ is the solution to the following **likelihood equation**:

$$l'(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^n [y_i - \mu(\boldsymbol{\beta}^T \mathbf{x}_i)] \mathbf{x}_i = 0. \quad (8)$$

Fig. 1. Schematic illustration of the compression and aggregation approach in a data cube.



In general, there are no closed-form analytical solutions to equation (8) and numerical optimization methods such as the Newton-Raphson method are needed to compute $\hat{\beta}$. Chen et al. [6] proved that $\hat{\beta}$ is strongly consistent under certain regularity conditions. *Strong consistency* means that $\hat{\beta}$ converges to the true parameter β with probability one when n goes to infinity.

Example 3: Consider a data set where each data record contains (y_i, x_i) , where y_i is a binary measure, and x_i is an explanatory variable. Suppose that we have the following ten data records $(0, 2)$, $(1, 3)$, $(0, 12)$, $(1, 4)$, $(0, 5)$, $(0, 8)$, $(0, 12)$, $(0, 3)$, $(1, 9)$, $(1, 5)$. Then, the log-likelihood function is

$$l(\beta) = [0 \times (\beta_0 + 2\beta_1) + \log(1 - \mu(\beta_0 + 2\beta_1))] + \\ \vdots \\ + [1 \times (\beta_0 + 5\beta_1) + \log(1 - \mu(\beta_0 + 5\beta_1))].$$

Taking derivatives with respect to $\beta = (\beta_0, \beta_1)^T$, we get the likelihood equation

$$l'(\beta) = [0 - \mu(\beta_0 + 2\beta_1)] \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \\ \vdots \\ + [1 - \mu(\beta_0 + 5\beta_1)] \begin{pmatrix} 1 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (9)$$

Using the Newton-Raphson method, we solve (9) and get the MLE $\hat{\beta} = (0.513515, -0.151184)^T$.

B. Proposed logistic regression cube

In this paper, in a data cube space, we consider aggregating cells along a dimension hierarchy. Note that an aggregation along multiple dimension hierarchies can be decomposed to multiple aggregations, each along a single dimension hierarchy. The aggregated cell c_a is given by the component cells c_1, \dots, c_k as $c_a = \bigcup_{i=1}^k c_i$. For the bank account data cube in Example 1, we may aggregate cells by merging the records at different locations or different times together.

Example 4: Continuing from Example 3, suppose we have component cells c_1 and c_2 that contain transactions in different

months, where $c_1 = \{(0,8), (1,10), (0,3), (1,10), (0,13), (0,8), (0,7), (1,15), (1,6), (0,3)\}$ and $c_2 = \{(0,5), (1,2), (1,5), (1,13), (0,4), (1,11), (1,10), (0,12), (0,1), (0,2)\}$.

The research challenge is, given the MLEs of the logistic regression model in c_1 and c_2 and possibly a few other quantities, we need to derive the aggregated model for c_a without scanning the raw data. The problem is schematically shown in Figure 1. In the figure, the circles and triangles denote the raw data. However, we cannot retain the raw data. Instead, for base cells c_1 and c_2 , we can retain the logistic regression model coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$, respectively. In addition, we may retain a small amount (independent of the number of tuple in a cell) of compressed measures A_1 and A_2 for the base cells. The question is, whether we can derive the logistic regression model for the aggregated cell c_a using only $\hat{\beta}_1, A_1$ and $\hat{\beta}_2, A_2$.

IV. ASYMPTOTICALLY LOSSLESS COMPRESSION AND AGGREGATION FOR LOGISTIC REGRESSION

Comparing to the OLS estimate of regression coefficients in linear regression models, the MLE of logistic regression coefficients are more difficult to compress and aggregate. The OLS estimates are solutions to linear equations and they have closed forms. However, the likelihood equations for the MLE in logistic regression are nonlinear equations that are not even polynomials, and the MLEs do not have closed-form representations and can only be obtained from nonlinear optimization procedures. As a result, it is much more difficult to aggregate the MLEs in logistic regression models.

We propose an asymptotically lossless compression technique to support efficient computation of the MLEs for logistic regression models in data cubes. We elaborate the notion of asymptotically lossless as follows.

Definition 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:

$$g(c) = \mathbf{v}, \text{ for any data cell } c \quad (10)$$

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

It is easily seen that any estimate of a set of fixed-length parameters, such as the MLE of the regression coefficients in logistic regression models, is a cell function.

Suppose that we have a logistic regression model $y = f(\mathbf{x}, \boldsymbol{\theta})$, where y and \mathbf{x} are attributes and $\boldsymbol{\theta}$ are coefficients. Suppose c_a is a cell aggregated from the component cells c_1, \dots, c_k . We define a cell function g_2 to obtain $\mathbf{m}_i = g_2(c_i)$, $i = 1, \dots, k$ and use an aggregation function g_1 to obtain an estimate of the regression coefficients for c_a by

$$\hat{\boldsymbol{\theta}} = g_1(\mathbf{m}_1, \dots, \mathbf{m}_k). \quad (11)$$

We say $\hat{\boldsymbol{\theta}}$, an estimate of $\boldsymbol{\theta}$, is an asymptotically losslessly compressible measure if

- the difference between $\tilde{\boldsymbol{\theta}} = g_1(\mathbf{m}_1, \dots, \mathbf{m}_k)$ and $\hat{\boldsymbol{\theta}}(c_a)$ tends to zero with probability one as the number of tuples in c_a goes to infinity;
- $\hat{\boldsymbol{\theta}}(c_a) = g_1(g_2(c_a))$; and
- the dimension of \mathbf{m}_i is independent of the number of tuples in c_i .

We call \mathbf{m}_i an **asymptotically lossless compression representation (ALCR)** of the cell $c_i, i = 1, \dots, k$. In the following, we develop an ALCR for logistic regression models by linearizing the likelihood equation. We show that the difference between the estimates obtained from aggregating the linearized equations in component cells and the MLE in the aggregated cell approaches zero when the number of tuples in the component cells is sufficiently large. Further, the space complexity of ALCR is independent of the number of tuples. As a result, our work shows that the MLEs for logistic regression are asymptotically losslessly compressible measures.

A. Compression and aggregation scheme

Consider aggregating K cells at a lower level into one aggregated cell at a higher level. For simplicity, we assume that each component cell contains n observations. This condition is not necessary for our theory, but it simplifies our notation. The results can be readily generalized to situations where the component cells have different sizes.

Denote the observations in the k^{th} component cell c_k by $\{(y_{k1}, \mathbf{x}_{k1}), \dots, (y_{kn}, \mathbf{x}_{kn})\}$, where each y_{ki} is a binary categorical attribute, and \mathbf{x}_{ki} is a d -dimensional vector of explanatory variables. For the logistic regression model in (2), we denote by $\hat{\boldsymbol{\beta}}_k$ the MLE of $\boldsymbol{\beta}$ in (2) based on the data cell c_k . Therefore, $\hat{\boldsymbol{\beta}}_k$ is the solution to the likelihood equation

$$l'_k(\boldsymbol{\beta}) = \frac{\partial l_k(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{j=1}^n [y_{kj} - \mu(\boldsymbol{\beta}^T \mathbf{x}_{kj})] \mathbf{x}_{kj} = 0.$$

Its Taylor's expansion at $\hat{\boldsymbol{\beta}}_k$ is given by

$$l'_k(\boldsymbol{\beta}) = - \sum_{j=1}^n \left[\dot{\mu}(\hat{\boldsymbol{\beta}}_k^T \mathbf{x}_{kj}) (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_k)^T \mathbf{x}_{kj} \right] \mathbf{x}_{kj} - \sum_{j=1}^n \left[\frac{1}{2} \ddot{\mu}(\hat{\boldsymbol{\beta}}_k^T \mathbf{x}_{kj}) ((\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_k)^T \mathbf{x}_{kj})^2 \right] \mathbf{x}_{kj}, \quad (12)$$

where the second equality comes from the fact $l'_k(\hat{\boldsymbol{\beta}}_k) = 0$, and $\hat{\boldsymbol{\beta}}_k^*$ is some vector between $\boldsymbol{\beta}$ and $\hat{\boldsymbol{\beta}}_k$ from Taylor's

theorem. Let $F_k(\boldsymbol{\beta})$ be the first-order approximation to the likelihood equation $l'_k(\boldsymbol{\beta})$, it follows from (12) that

$$\begin{aligned} F_k(\boldsymbol{\beta}) &= - \sum_{j=1}^n \left[\dot{\mu}(\hat{\boldsymbol{\beta}}_k^T \mathbf{x}_{kj}) (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_k)^T \mathbf{x}_{kj} \right] \mathbf{x}_{kj} \\ &= -A_k \boldsymbol{\beta} + A_k \hat{\boldsymbol{\beta}}_k, \end{aligned}$$

where

$$A_k = \sum_{j=1}^n \left[\dot{\mu}(\hat{\boldsymbol{\beta}}_k^T \mathbf{x}_{kj}) \mathbf{x}_{kj} \mathbf{x}_{kj}^T \right] = \sum_{j=1}^n \left[\frac{(e^{\hat{\boldsymbol{\beta}}_k^T \mathbf{x}_{kj}}) \mathbf{x}_{kj} \mathbf{x}_{kj}^T}{(1 + e^{\hat{\boldsymbol{\beta}}_k^T \mathbf{x}_{kj}})^2} \right]. \quad (13)$$

It is clear that $F_k(\boldsymbol{\beta})$ only depends on A_k and $\hat{\boldsymbol{\beta}}_k$, whose dimensions are independent of the number of data records in c_k . The size of $\hat{\boldsymbol{\beta}}_k$ is d , the number of coefficients in the logistic regression model, while A_k is a $d \times d$ matrix. Since $F_k(\boldsymbol{\beta})$ approximates $l'_k(\boldsymbol{\beta})$ by the first order, we can save $(\hat{\boldsymbol{\beta}}_k, A_k)$ as a compressed synopsis of each data cell, and solve the linearized equation

$$F_a(\boldsymbol{\beta}) = \sum_{k=1}^K F_k(\boldsymbol{\beta}) = 0. \quad (14)$$

in the aggregated cell instead of $\sum_k l'_k(\boldsymbol{\beta}) = 0$. By saving only $(\hat{\boldsymbol{\beta}}_k, A_k)$ in cell c_k , we can compute $\hat{\boldsymbol{\beta}}_a$, the solution to (14).

Using this compression scheme, we can approximate the MLE of $\boldsymbol{\beta}$ in c_a using the solution to

$$F_a(\boldsymbol{\beta}) = \sum_{k=1}^K F_k(\boldsymbol{\beta}) = \sum_{k=1}^K \left(-A_k \boldsymbol{\beta} + A_k \hat{\boldsymbol{\beta}}_k \right) = 0, \quad (15)$$

which leads to:

$$\tilde{\boldsymbol{\beta}}_a = \left(\sum_{k=1}^K A_k \right)^{-1} \sum_{k=1}^K A_k \hat{\boldsymbol{\beta}}_k. \quad (16)$$

In addition, in the aggregated cell c_a , we can also obtain its A_a from the A_k of components cells by observing:

$$A_a = \sum_{k=1}^K \sum_{j=1}^n \left[\dot{\mu}(\hat{\boldsymbol{\beta}}_k^T \mathbf{x}_{kj}) \mathbf{x}_{kj} \mathbf{x}_{kj}^T \right] = \sum_{k=1}^K A_k \quad (17)$$

To summarize, our asymptotically lossless compression technique can be described as the following.

- Compression into ALCR.** For each *base* cell $c_k, k = 1, \dots, K$, at the lowest level of the data cube, calculate the MLEs $\hat{\boldsymbol{\beta}}_k$ using numerical methods and A_k using (13). Save

$$\text{ALCR} = (\hat{\boldsymbol{\beta}}_k, A_k) \quad (18)$$

in each component cell c_k .

- Aggregation of ALCR.** Calculate the aggregated ALCR $(\hat{\boldsymbol{\beta}}_a, A_a)$ using (16) and (17). 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, $\tilde{\boldsymbol{\beta}}$ is used in place of $\hat{\boldsymbol{\beta}}$ in its ALCR.

B. Compressibility of logistic regression

We now show that $(\hat{\beta}_k, A_k)$ is an ALCR. In the sequel, we denote the MLE of coefficients for the aggregated cell to be $\hat{\beta}_a$, whereas the corresponding estimates derived from ALCR compression and aggregation to be $\tilde{\beta}_a$. We denote the underlying true logistic model for the data set to have coefficients β_0 . We show that the difference between $\tilde{\beta}_a$ and $\hat{\beta}_a$ approaches zero asymptotically as the number of data records in base cells increases.

The results are based on the following assumptions (for more discussions of these conditions, see [6], [10]).

- (C1) The minimum eigenvalue λ_k of $\sum_{j=1}^n \mathbf{x}_{kj} \mathbf{x}_{kj}^T$ goes to ∞ as $n \rightarrow \infty$ for each $k = 1, \dots, K$.
- (C2) $\sum_{j=1}^{\infty} c_j \varepsilon_{kj}$ converges a.s.¹ for any sequence of constants $\{c_j\}$ satisfying $\sum_{j=1}^{\infty} c_j^2 < \infty$, where $\varepsilon_{kj} = y_{kj} - \mu(\beta_0^T \mathbf{x}_{kj})$, for each $k = 1, \dots, K$.
- (C3) $\frac{1}{n} \lambda_k$ has a uniform lower positive bound for all $k = 1, \dots, K$.

These three mild assumptions are used to rule out some very special cases and are satisfied for most real data sets.

Condition (C3) actually implies Condition (C1). If (C3) is true, there is a constant $C_1 > 0$ such that $\frac{1}{n} \lambda_k > C_1$, i.e. $\lambda_k > C_1 n$. Therefore λ_k tends to infinity as n tends to infinity and Condition (C1) is fulfilled.

In practice, Condition (C3) can be easily satisfied. It is usually proper to assume that \mathbf{x}_{kj} 's come from a common distribution \mathcal{D} . Without loss of generality, let us assume that the distribution \mathcal{D} has a zero expectation. In this case $\sum_{j=1}^n \mathbf{x}_{kj} \mathbf{x}_{kj}^T$ tends to the covariance matrix Σ of the distribution \mathcal{D} with probability one, and thus $\frac{1}{n} \lambda_k$ will be arbitrarily close to the minimum eigenvalue of Σ which is greater than zero when n is large enough. Therefore, when n is large, $\frac{1}{n} \lambda_k$ is greater than one half of the minimum eigenvalue of Σ and Condition (C3) is fulfilled. On the other hand, when \mathbf{x}_{kj} 's are not random, Condition (C3) is typically also satisfied. For example, suppose \mathbf{x}_{kj} 's are some given fixed real numbers satisfying $|\mathbf{x}_{kj}| > C$ for $C > 0$, then Condition (C3) will be true with a lower bound C^2 .

Condition (C2) is also mild. It is actually implied by the independence of ε_{kj} 's [6], [10], which in turn is implied by the independence of y_{kj} 's if \mathbf{x}_{kj} 's are not random. Finiteness of \mathbf{x}_{kj} 's and independence of y_{kj} 's can also guarantee Condition (C2).

In order to establish our results, we first need to define the norm of a matrix.

Definition 2: Suppose A is a $d \times d$ matrix, the **norm** of A , denoted as $\|A\|$, is defined as

$$\|A\| = \sup_{\mathbf{v} \in \mathbb{R}^d, \mathbf{v} \neq \mathbf{0}} \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|}$$

The norm of a vector can be any type of norm, and we usually use the Euclidean norm which is the square root of the sum of squares of vector elements.

¹a.s. is a term in probability theory which denotes "almost sure" [10]. Intuitively, it means that an event will happen with probability one.

We need the following two results on matrix norms in order to bound the error of the estimator based on the proposed compression scheme.

Lemma 4.1: Suppose A is a $d \times d$ positive definite matrix, if λ is the smallest eigenvalue of A , we have

$$\mathbf{v}^T A \mathbf{v} \geq \lambda \mathbf{v}^T \mathbf{v} = \lambda \|\mathbf{v}\|^2$$

for any vector $\mathbf{v} \in \mathbb{R}^d$. In addition, if there exists a constant $C > 0$ such that $\mathbf{v}^T A \mathbf{v} \geq C \|\mathbf{v}\|^2$ for any vector $\mathbf{v} \in \mathbb{R}^d$, then we have $\lambda \geq C$.

Proof: Since A is positive definite matrix, there exists a $d \times d$ unitary matrix U and a $d \times d$ diagonal matrix D such that $A = U^T D U$ and D 's nonzero elements are the eigenvalues of A . Take any $\mathbf{v} \in \mathbb{R}^d$, and denote $\mathbf{u} = U\mathbf{v}$. Then we have:

$$\mathbf{v}^T A \mathbf{v} = \mathbf{v}^T U^T D U \mathbf{v} = \mathbf{u}^T D \mathbf{u}. \quad (19)$$

Since D is diagonal and λ is the smallest element of D 's main diagonal elements, we have

$$\mathbf{u}^T D \mathbf{u} \geq \lambda \mathbf{u}^T \mathbf{u}. \quad (20)$$

Since U is a unitary matrix, $\mathbf{u}^T \mathbf{u} = \mathbf{v}^T U^T U \mathbf{v} = \mathbf{v}^T \mathbf{v}$. Therefore, combining (19) and (20) leads to

$$\mathbf{v}^T A \mathbf{v} \geq \lambda \|\mathbf{v}\|^2.$$

To show the second part, suppose $\mathbf{v}^T A \mathbf{v} \geq C \|\mathbf{v}\|^2$ for some $C > 0$. We can find an eigenvector $\mathbf{v}_\lambda \neq \mathbf{0}$ corresponding to the eigenvalue λ , i.e. $A \mathbf{v}_\lambda = \lambda \mathbf{v}_\lambda$. For this particular vector \mathbf{v}_λ , we have

$$\mathbf{v}_\lambda^T A \mathbf{v}_\lambda = \lambda \|\mathbf{v}_\lambda\|^2.$$

The second part of the Lemma follows from the assumption $\mathbf{v}_\lambda^T A \mathbf{v}_\lambda \geq C \|\mathbf{v}_\lambda\|^2$. ■

Lemma 4.2: Let A be a $d \times d$ positive definite matrix and λ be the smallest eigenvalue of A . If $\lambda \geq C > 0$ for some constant C , we have $\|A^{-1}\| \leq \frac{1}{C}$.

Proof: Since A is a positive definite matrix, A^{-1} is also a positive definite matrix and the reciprocals of the eigenvalues of A are the eigenvalues of A^{-1} . Therefore, λ^{-1} must be the largest eigenvalue of A^{-1} . Hence, for any $\mathbf{v} \in \mathbb{R}^d$, we have $\|A^{-1} \mathbf{v}\| \leq \lambda^{-1} \|\mathbf{v}\|$. It follows that $\|A^{-1}\| \leq \lambda^{-1} \leq \frac{1}{C}$. ■

Now we develop our key results to bound the estimation error and show that the proposed compression scheme is asymptotically lossless.

Theorem 4.1: For a logistic regression model, suppose that conditions (C1) and (C2) are satisfied and $\sup_{k,j} \|\mathbf{x}_{kj}\| < \infty$. Then, $\hat{\beta}_k \rightarrow \beta_0$ a.s. as n goes to infinity. Specifically, for any $\delta > 0$, when n is large enough we have,

$$\|\tilde{\beta}_a - \beta_0\| \leq K \|\hat{\beta}_{k_0} - \beta_0\| = o\left(K \{[\log(\lambda_{k_0})]^{1+\delta} / \lambda_{k_0}\}^{1/2}\right),$$

where $k_0 = \operatorname{argmax}_{1 \leq k \leq K} \{\|\hat{\beta}_k - \beta_0\|\}$. If in addition the condition (C3) is satisfied, we have

$$\|\tilde{\beta}_a - \beta_0\| = o\left(\{[\log(\lambda_{k_0})]^{1+\delta} / \lambda_{k_0}\}^{1/2}\right).$$

Proof: Our proof strategy starts by noticing that $\tilde{\beta}_a$ is obtained from the aggregation equation (16). So we can get $\|\tilde{\beta}_a - \beta_0\|$ by subtracting β_0 from both sides of (16). We can

then get the first part by the triangle inequality and Theorem 1 in [6]. For the second part, we can use condition (C3) and get $\|(\sum_k A_k)^{-1} \cdot A_k\| \leq \frac{M}{K}$ for some constant M . We then use the triangle inequality again to reach the desired result. Below are the details.

The derivative of the function μ is

$$\dot{\mu}(t) = \frac{e^t}{(1+e^t)^2} > 0. \quad (21)$$

From Theorem 1 in [6] and conditions (C1) and (C2), $\hat{\beta}_k$'s are close to $\hat{\beta}_0$ when n is large enough. Since \mathbf{x}_{kj} 's are bounded, we can find a constant $c > 0$ for all k and j such that $|\hat{\beta}_k^T \mathbf{x}_{kj}| < c$. Since the function $\dot{\mu}(t) > 0$ is continuous, there exists a constant $a > 0$ such that $\dot{\mu}(t) \geq a$ for all $|t| \leq c$. The following inequality holds

$$A_k = \sum_{j=1}^n \left[\dot{\mu}(\hat{\beta}_k^T \mathbf{x}_{kj}) \mathbf{x}_{kj} \mathbf{x}_{kj}^T \right] \geq a \sum_{j=1}^n \mathbf{x}_{kj} \mathbf{x}_{kj}^T.$$

So A_k is positive definite when n is large enough for each $k = 1, \dots, K$. Thus, $\sum_{k=1}^K A_k$ is a positive definite matrix when n is large enough. In particular, $(\sum_{k=1}^K A_k)^{-1}$ exists and Equation (16) is meaningful. Subtracting β_0 from both sides of (16), we get

$$\tilde{\beta}_a - \beta_0 = \left[\left(\sum_{k=1}^K A_k \right)^{-1} \sum_{k=1}^K A_k \hat{\beta}_k \right] - \beta_0,$$

which implies

$$\tilde{\beta}_a - \beta_0 = \left(\sum_{k=1}^K A_k \right)^{-1} \left[\sum_{k=1}^K A_k (\hat{\beta}_k - \beta_0) \right].$$

Using the triangle inequality, we have

$$\begin{aligned} \|\tilde{\beta}_a - \beta_0\| &\leq \sum_{k=1}^K \left\| \left(\sum_{k=1}^K A_k \right)^{-1} A_k (\hat{\beta}_k - \beta_0) \right\| \quad (22) \\ &\leq \sum_{k=1}^K \|\hat{\beta}_k - \beta_0\|. \quad (23) \end{aligned}$$

The second inequality comes from the fact

$$\left\| \left(\sum_{k=1}^K A_k \right)^{-1} A_k \right\| \leq 1, \quad \forall k = 1, \dots, K.$$

By Theorem 1 in [6], we prove the first part.

If furthermore condition (C3) is satisfied, we have a constant C_1 such that $\frac{1}{n} \lambda_k > C_1$ for all $k = 1, \dots, K$. By Lemma 4.1, for any $\mathbf{v} \in \mathbb{R}^d$,

$$\mathbf{v}^T \left(\sum_{j=1}^n \mathbf{x}_{kj} \mathbf{x}_{kj}^T \right) \mathbf{v} \geq \lambda_k \|\mathbf{v}\|^2 \quad (24)$$

On the other hand, when n is large enough, we can find a constant α such that $1 > \dot{\mu}(\hat{\beta}_k^T \mathbf{x}_{kj}) > \alpha > 0$ for any k and j , since $\sup_{k,j} \|\mathbf{x}_{kj}\| < C_2$ for some constant C_2 and $\hat{\beta}_k$'s are

strongly consistent by Theorem 1 in [6]. Therefore, for any $\mathbf{v} \in \mathbb{R}^d$,

$$\mathbf{v}^T \left(\sum_{j=1}^n \mathbf{x}_{kj} \mathbf{x}_{kj}^T \right) \mathbf{v} \geq \mathbf{v}^T A_k \mathbf{v} \geq \alpha \mathbf{v}^T \left(\sum_{j=1}^n \mathbf{x}_{kj} \mathbf{x}_{kj}^T \right) \mathbf{v}.$$

Hence, we have:

$$\begin{aligned} \mathbf{v}^T \left(\frac{1}{nK} \sum_{k=1}^K A_k \right) \mathbf{v} &\geq \alpha \mathbf{v}^T \left(\frac{1}{nK} \sum_{k=1}^K \sum_{j=1}^n \mathbf{x}_{kj} \mathbf{x}_{kj}^T \right) \mathbf{v} \\ &\geq C_1 \alpha \|\mathbf{v}\|^2, \end{aligned}$$

where the second inequality is from (24) and the condition (C3). By Lemma 4.1, the smallest eigenvalue of $\frac{1}{nK} \sum_{k=1}^K A_k$ is greater than $C_1 \alpha$, and then, by Lemma 4.2, we know:

$$\left\| \left(\frac{1}{nK} \sum_{k=1}^K A_k \right)^{-1} \right\| \leq \frac{1}{C_1 \alpha}. \quad (25)$$

From the above, we find that:

$$\begin{aligned} \left\| \left(\sum_{k=1}^K A_k \right)^{-1} \cdot A_k \right\| &\leq \left\| \left(\frac{1}{nK} \sum_{k=1}^K A_k \right)^{-1} \right\| \cdot \left\| \frac{1}{nK} A_k \right\| \\ &\leq \frac{1}{C_1 \alpha n K} \sum_{j=1}^n \left\| \dot{\mu}(\hat{\beta}_k^T \mathbf{x}_{kj}) \mathbf{x}_{kj} \mathbf{x}_{kj}^T \right\| \\ &\leq \frac{M}{K}, \end{aligned}$$

where $M = \frac{C_2^2}{C_1 \alpha}$. At last, by Equation (22), definition of k_0 , and Theorem 1 in [6], we get:

$$\begin{aligned} \|\tilde{\beta}_a - \beta_0\| &\leq \sum_{k=1}^K \frac{M}{K} \|\hat{\beta}_k - \beta_0\| \\ &= o\left(\{(\log \lambda_{k_0})^{1+\delta} / \lambda_{k_0}\}^{1/2}\right), \end{aligned}$$

which proves the second part. \blacksquare

Theorem 4.1 establishes that our proposed aggregated estimator is asymptotically converging to the true model as the size of data sets n increases. Note that $n \rightarrow \infty$ implies $\lambda_{k_0} \rightarrow \infty$ by condition (C1), which in turn implies $\|\tilde{\beta}_a - \beta_0\| \rightarrow 0$.

The following theorem shows that, the difference between $\tilde{\beta}_a$ and $\hat{\beta}_a$ diminishes as the size of data increases.

Theorem 4.2: If conditions (C1), (C2), and (C3) are satisfied, then for any $\delta > 0$, when n is large enough we have

$$\|\tilde{\beta}_a - \hat{\beta}_a\| \leq o\left([\log \lambda_{k_0}]^{1+\delta} / \lambda_{k_0}\right). \quad (26)$$

Proof: Since $\hat{\beta}_a$ is the MLE based on all the data in the aggregated cell, we use the fact that $\hat{\beta}_a$ is the solution to the equation

$$\sum_{k=1}^K l'_k(\beta) = 0. \quad (27)$$

and calculate the difference between (14) and (27) to show the result.

Since $\sum_{k=1}^K F_k(\tilde{\beta}_a) = 0$ and $\sum_{k=1}^K l'_k(\hat{\beta}_a) = 0$, we have

$$\begin{aligned} 0 &= \sum_{k=1}^K (F_k(\tilde{\beta}_a) - l'_k(\hat{\beta}_a)) \\ &= -\sum_{k=1}^K A_k(\tilde{\beta}_a - \hat{\beta}_k) + \sum_{k=1}^K A_k(\hat{\beta}_a - \hat{\beta}_k) + \\ &\quad \frac{1}{2} \left[\sum_{k,j=1}^{K,n} \ddot{\mu}(\hat{\beta}_k^{*T} \mathbf{x}_{kj}) \mathbf{x}_{kj} \mathbf{x}_{kj}^T (\hat{\beta}_a - \hat{\beta}_k) (\tilde{\beta}_a - \hat{\beta}_k)^T \mathbf{x}_{kj} \right], \end{aligned}$$

where $\hat{\beta}_k^*$ is some vector between $\hat{\beta}_a$ and $\hat{\beta}_k$, as in the Taylor expansion in (12). Hence,

$$\begin{aligned} \tilde{\beta}_a - \hat{\beta}_a &= \frac{1}{2} \left(\sum_{k=1}^K A_k \right)^{-1} \sum_{k,j=1}^{K,n} \left[\ddot{\mu}(\hat{\beta}_k^{*T} \mathbf{x}_{kj}) \right. \\ &\quad \left. \mathbf{x}_{kj} \mathbf{x}_{kj}^T (\hat{\beta}_a - \hat{\beta}_k) (\hat{\beta}_a - \hat{\beta}_k)^T \mathbf{x}_{kj} \right]. \end{aligned}$$

As n goes to infinity, $\hat{\beta}_a$ and $\hat{\beta}_k$ tend to the true parameter β_0 . So $\hat{\beta}_k^*$ also tends to β_0 when n goes to infinity. Therefore, we can find a constant C_3 such that $|\ddot{\mu}(\hat{\beta}_k^{*T} \mathbf{x}_{kj})| \leq C_3$ when n is large enough. Using the same technique in the proof for the second part of Theorem 4.1 to quantify $\tilde{\beta}_a - \hat{\beta}_a$, we can get:

$$\|\tilde{\beta}_a - \hat{\beta}_a\| \leq \frac{1}{K} C_1^{-1} \alpha^{-1} C_2^3 C_3 \sum_{k=1}^K \|\hat{\beta}_a - \hat{\beta}_{k_0}\|^2.$$

Since the smallest eigenvalue of $\sum_{k,j} \mathbf{x}_{kj} \mathbf{x}_{kj}^T$ must be larger than $\lambda = \sum_k \lambda_k$, we have

$$\|\hat{\beta}_a - \beta_0\| = o\left(\{(\log \lambda)^{1+\delta}/\lambda\}^{1/2}\right).$$

Then again by Theorem 1 in [6] and the triangle inequality,

$$\begin{aligned} \|\hat{\beta}_a - \hat{\beta}_{k_0}\| &\leq \|\hat{\beta}_a - \beta_0\| + \|\hat{\beta}_{k_0} - \beta_0\| \\ &= o\left(\{(\log \lambda_{k_0})^{1+\delta}/\lambda_{k_0}\}^{1/2}\right). \end{aligned}$$

Therefore, we have

$$\|\tilde{\beta}_a - \hat{\beta}_a\| = o\left(\{(\log \lambda_{k_0})^{1+\delta}/\lambda_{k_0}\}\right).$$

The result is arrived. \blacksquare

Theorem 4.2 implies that the MLE $\hat{\beta}_a$ is asymptotically losslessly compressible since $\tilde{\beta}_a$ approaches $\hat{\beta}_a$ as n approaches infinity.

In the proof to Theorem 4.2, since the minimum eigenvalue λ_k satisfies $\lambda_k/n > C_1$ and $\sup_{k,j} \|\mathbf{x}_{kj}\| < C_2$ for some constants C_1 and C_2 , we have $(\log \lambda_{k_0})^{1+\delta}/\lambda_{k_0} = O((\log n)^{1+\delta}/n)$ and $\|\tilde{\beta}_a - \hat{\beta}_a\| = o((\log n)^{1+\delta}/n)$ from Theorem 4.2. Therefore, the difference between $\tilde{\beta}_a$ and $\hat{\beta}_a$ is roughly bounded by the reciprocal of the number of tuples in the base cell. If the sizes of the base cells are different, we have $\|\tilde{\beta}_a - \hat{\beta}_a\| = o((\log n_0)^{1+\delta}/n_0)$, where n_0 is the number of tuples in the smallest cell. Hence, it seems that the difference between $\tilde{\beta}_a$ and $\hat{\beta}_a$ would be large if the sizes

of base cells are very different. However, the estimate of the difference between $\tilde{\beta}_a$ and $\hat{\beta}_a$ in Theorem 4.2 is very conservative and the difference may still be small even if the sizes of base cells are very different. For instance, suppose we have two base cells c_1 and c_2 . The cell c_1 has 1 million tuples, but the cell c_2 has only 20 tuples. Then using (16), $\tilde{\beta}_a = (A_1 + A_2)^{-1}(A_1\hat{\beta}_1 + A_2\hat{\beta}_2)$, where $(A_k, \hat{\beta}_k)$ is the ALCR of the cell c_k ($k = 1, 2$). That is, $\tilde{\beta}_a$ is a weighted sum of $\hat{\beta}_1$ and $\hat{\beta}_2$ and we put much more weight on $\hat{\beta}_1$ than $\hat{\beta}_2$ because there are far more tuples in cell c_1 than cell c_2 . With 1 million tuples in the cell c_1 , $\hat{\beta}_1$ should be very close to the true parameter β_0 . Therefore, $\tilde{\beta}_a$ should also be very close to the true parameter β_0 . On the other hand, $\hat{\beta}_a$ is also close to β_0 and hence the difference between $\tilde{\beta}_a$ and $\hat{\beta}_a$ should be small.

Example 5: Continuing from Example 4, suppose that the transactions in c_1 are all the transactions of the first month and those in c_2 are of the second month. From the first month's transactions we can calculate the MLE $\hat{\beta}_1 = (0.513515, -0.151184)^T$, and by (13) we get

$$A_1 = \begin{pmatrix} 2.25748 & 13.1585 \\ 13.1585 & 101.465 \end{pmatrix}.$$

Similarly, from the second month's transactions we get $\hat{\beta}_2 = (0.732276, -0.0343601)^T$ and

$$A_2 = \begin{pmatrix} 2.38056 & 22.9297 \\ 22.9297 & 289.063 \end{pmatrix}.$$

So the aggregated maximum likelihood equation after linearization is

$$F_a(\beta) = -(A_1 + A_2)\beta + A_1\hat{\beta}_1 + A_2\hat{\beta}_2 = 0,$$

Solving the above equation, we get the aggregated MLE $\tilde{\beta}_a = (0.218388, -0.0245961)^T$. If we use all the data from the two months to directly calculate the MLE, we get the MLE $\hat{\beta}_a = (0.235159, -0.0299553)^T$. Notice that even though the sample size is pretty small, our aggregated estimate $\tilde{\beta}_a$ is close to the MLE $\hat{\beta}_a$ with relative difference

$$\|\tilde{\beta}_a - \hat{\beta}_a\|/\|\hat{\beta}_a\| \simeq 0.074.$$

This shows that our aggregated estimation approximates the MLE fairly accurately even when very limited amount of data (only 20 samples) are available. According to our theory, the approximation error will diminish as more data become available.

C. Extension to multi-valued response variables

To simplify the notations, we have developed the theory for binary response. The proposed scheme can also be extended to support logistic regressions for multi-valued response variables following a similar first-order approximation scheme. Suppose the response variable Y takes $m + 1$ possible values for $m > 1$. In general, one can create m binary variables and then fit m separate logistic regression models to study the relationship. However, this method needs a large number of parameters, especially when m is large. Oftentimes, the multiple values of the response variable have a natural order. In this

case, the response variable Y is called polytomous data with ordinal scales, which can be modelled by the proportional-odds model [19]. Without loss of generality, we assume the $m + 1$ values are $1, 2, \dots, m, m + 1$. Let $\gamma_j = P(Y \leq j)$. The proportional-odds model describes the dependence of the response variable Y on the predictor \mathbf{x} as

$$\text{logit}(\gamma_j) = \log[\gamma_j/(1 - \gamma_j)] = \alpha_j + \beta^T \mathbf{x}$$

for $j = 1, \dots, m + 1$. The data likelihood based on this model is given by

$$L(y) = \begin{cases} \mu(\alpha_1 + \beta^T \mathbf{x}) & y = 1 \\ \mu(\alpha_y + \beta^T \mathbf{x}) - \mu(\alpha_{y-1} + \beta^T \mathbf{x}) & 1 < y \leq m \\ 1 - \mu(\alpha_k + \beta^T \mathbf{x}) & y = m + 1. \end{cases}$$

Let $\xi = (\alpha_1, \dots, \alpha_m, \beta^T)^T$ and $l(y) = \log(L(y))$. Suppose that the data set has n observations $(y_1, \mathbf{x}_1), \dots, (y_n, \mathbf{x}_n)$, we can obtain the MLE $\hat{\xi}$ of ξ by solving the following estimating equation

$$\sum_{i=1}^n \frac{\partial l(y_i)}{\partial \xi} = \sum_{i=1}^n \frac{1}{L(y_i)} \frac{\partial L(y_i)}{\partial \xi} = \mathbf{0}. \quad (28)$$

Let $\mathbf{e}_1 = (1, 0, \dots, 0)^T$, $\mathbf{e}_2 = (0, 1, \dots, 0)^T$, \dots , $\mathbf{e}_k = (0, 0, \dots, 1)^T \in \mathbb{R}^k$ and $\mathbf{z}_j(\mathbf{x}) = (\mathbf{e}_j^T, \mathbf{x}^T)^T$. The derivative of $L(y)$ can be written as

$$\frac{\partial L(y)}{\partial \xi} = \begin{cases} \dot{\mu}(\alpha_1 + \beta^T \mathbf{x}) \mathbf{z}_1(\mathbf{x}) & y = 1 \\ \dot{\mu}(\alpha_y + \beta^T \mathbf{x}) \mathbf{z}_y(\mathbf{x}) - \dot{\mu}(\alpha_{y-1} + \beta^T \mathbf{x}) \mathbf{z}_{y-1}(\mathbf{x}) & 1 < y \leq m \\ -\dot{\mu}(\alpha_k + \beta^T \mathbf{x}) \mathbf{z}_k(\mathbf{x}) & y = m + 1 \end{cases},$$

and the Hessian matrix of $L(y)$ is

$$\frac{\partial^2 L(y)}{\partial \xi^2} = \begin{cases} \ddot{\mu}(\alpha_1 + \beta^T \mathbf{x}) \mathbf{z}_1(\mathbf{x}) \mathbf{z}_1(\mathbf{x})^T & y = 1 \\ \ddot{\mu}(\alpha_y + \beta^T \mathbf{x}) \mathbf{z}_y(\mathbf{x}) \mathbf{z}_y(\mathbf{x})^T - \ddot{\mu}(\alpha_{y-1} + \beta^T \mathbf{x}) \mathbf{z}_{y-1}(\mathbf{x}) \mathbf{z}_{y-1}(\mathbf{x})^T & 1 < y \leq m \\ -\ddot{\mu}(\alpha_k + \beta^T \mathbf{x}) \mathbf{z}_k(\mathbf{x}) \mathbf{z}_k(\mathbf{x})^T & y = m + 1 \end{cases}.$$

Differentiate (28) with respect to ξ , we get the Hessian matrix of the log likelihood function

$$H(\xi) = \sum_{i=1}^n \frac{\partial^2 l(y_i)}{\partial \xi^2} = \sum_{i=1}^n \left[\frac{1}{L(y_i)} \frac{\partial^2 L(y_i)}{\partial \xi^2} - \frac{1}{L^2(y_i)} \frac{\partial L(y_i)}{\partial \xi} \left(\frac{\partial L(y_i)}{\partial \xi} \right)^T \right].$$

Taylor expansion at the MLE ξ gives the following approximated estimating equation

$$H(\hat{\xi})(\xi - \hat{\xi}) = \mathbf{0}. \quad (29)$$

Let $\hat{\xi}_k$ be the MLE in the k^{th} cell and $A_k = H(\hat{\xi}_k)$. We can save $(\hat{\xi}_k, A_k)$ as the ALCR for each cell and get the estimate $\hat{\xi}_a = (\sum_{k=1}^K A_k)^{-1} \sum_{k=1}^K (A_k \hat{\xi}_k)$ for the aggregated data set. Compressibility of the proposed scheme can be shown following the same proof strategy as that used for the binary response case.

TABLE I
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)

D. Detection of non-homogenous data

Our compression theory relies on the assumption that the subcubes are homogeneous in the sense that a same logistic regression model holds in all the subcubes. When we have non-homogeneous data, it is well known that aggregation over non-homogeneous subsets can cause misleading results in categorical data analysis. A famous example is the Simpson's paradox [2], which can be illustrated by a medical study [4], [16] comparing the success rates of two treatments for kidney stones. The two treatments are open surgery (treatment A) and percutaneous nephrolithotomy (treatment B).

Table I 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 $y = 1$ (*resp.* $y = 0$) denote the treatment is successful (*resp.* unsuccessful), and $x = 1$ (*resp.* $x = 0$) denote that treatment A (*resp.* B) is applied. We can model the success rate p by the logistic regression model

$$\text{logit}(p) = \beta_0 + \beta_1 x.$$

Using the data from the small stone group, we get the MLE estimate $(\hat{\beta}_{0s}, \hat{\beta}_{1s}) = (1.87, 0.73)$, and the corresponding A_s is,

$$A_s = \begin{pmatrix} 36.79 & 5.59 \\ 5.59 & 5.59 \end{pmatrix}.$$

The estimates of β_0 and β_1 can then be used to estimate the success rates of treatment A and treatment B. Let p_A be the success rates of treatment A and p_B the success rate of treatment B, we have the estimates $p_A = 0.93$ and $p_B = 0.87$.

Similarly, for the large stone group, we get $(\hat{\beta}_{0l}, \hat{\beta}_{1l}) = (0.79, 0.21)$ and the corresponding A_l is,

$$A_l = \begin{pmatrix} 69.02 & 51.83 \\ 51.83 & 51.83 \end{pmatrix}.$$

Then we get another estimates of success rates, $p_A = 0.73$ and $p_B = 0.69$. Therefore, treatment A shows a higher success rate than treatment B in both situations.

However, if we use the entire data set to estimate the parameter, we will get $(\hat{\beta}_{0e}, \hat{\beta}_{1e}) = (1.56, -0.29)$, $p_A = 0.78$ and $p_B = 0.83$, which shows that, contrary to the previous estimates, treatment A has a lower success rate than treatment B. The occurrence of Simpson's paradox is largely due to the non-homogeneity of the data sets. The estimates of the parameters for the two groups are largely different, which implies that it is very likely that the two data sets are not homogenous.

Our compression technique can actually serve as a device to detect nonhomogeneity in a data set. Let $\hat{\beta}_i$ and A_i be the ALCR of cell c_i ($i = 1, 2$), respectively. It is proved that $A_i^{1/2}(\hat{\beta}_i - \beta_{i0})$ asymptotically follows a standard normal

TABLE II
PERFORMANCE OF THE AGGREGATED ESTIMATOR $\tilde{\beta}_a$.

	K=1000	K=100	K=10	K=1 (MLE)
$\tilde{\beta}_a$	(0.3110, -0.1819)	(0.3116, -0.1826)	(0.3116, -0.1826)	(0.3116, -0.1826)
Compression Time (Sec.)	366	347	377	636
Aggregation Time (Sec.)	<0.1	<0.1	<0.1	0
$\frac{\ \tilde{\beta}_a - \beta_0\ }{\ \beta_0\ }$	0.9%	0.8%	0.8%	0.8%
Memory	8,000	20,600	200,060	2,000,000

distribution [11], where β_{i0} is the underlying true parameter for the cell c_i ($i = 1, 2$). Following the standard multivariate normal distribution theory, if the two cells c_1 and c_2 are homogeneous, we know that the statistic $\chi = (\hat{\beta}_1 - \hat{\beta}_2)^T (A_1^{-1} + A_2^{-1})^{-1} (\hat{\beta}_1 - \hat{\beta}_2)$ is asymptotically χ_p^2 -distributed, where p is the dimension of β_{i0} . Hence we can use the statistic χ to construct a chi-square test to test the homogeneity of the two cells. For the kidney stone example, we have $\chi = 26.04$ which is highly significant. This indicates that the two kidney stone groups are very likely non-homogeneous and not suitable for aggregation.

Sometimes, the class-label distributions in some cells may be skewed. For example, some cell may have much more positive examples than negative ones. The strong skewness of the class-label is a known problem in logistic regression, not specific to the data cube context. The maximum likelihood estimate may not exist in such a circumstance. If only a few cells show the skewness of the class-label in a data cube, it is likely that these cells and other cells are not homogenous. Aggregating these cells with other cells may not be appropriate.

V. EXPERIMENTAL RESULTS

We perform experimental studies on synthetic and real data to validate our method. We study the aggregation accuracy, the space usage, and the computational efficiency of the proposed scheme. There are a number of advantages of the proposed scheme we aim to validate through the experiments. First, we show that the error of the compression is small. Second, we show that using the compression and aggregation can tremendously reduce the computational complexity for OLAPing queries. Third, we show that, although we assume homogeneity of component cells, aggregation is meaningful as it can improve the model accuracy over subsampling. Finally, we show that the compression allows efficient incremental aggregation for mining stream data.

A. Quality and efficiency of compression and aggregation in data cubes

We run several experiments to evaluate various properties of the proposed scheme.

a) *Experiment 1.*: In the first simulation, we consider a logistic regression model with one predictor and one intercept term. The true regression coefficients are set as $\beta_0 = (0.314, 0.181)^T$ and the total sample size is $N = 1,000,000$. We generate the predictors x_1, \dots, x_N independently from the standard normal distribution. Then y_i is simulated from the Bernoulli distribution with parameter

$$p_i = \mu[\beta_0^T \cdot (1, x_i^T)^T].$$

We partition the entire data set into $K = 10^3, 10^2$, or 10 cells with equal number of observations. The original N observations can be viewed as an aggregated cell from the K base cells. We then apply our aggregation algorithm to approximate the MLE for the entire data set. In the compression step, we use the Newton-Raphson algorithm to find the MLE β_k in each base cell c_k .

Table II shows our aggregated estimates $\tilde{\beta}_a$ for different values of K and compares them to the MLE $\hat{\beta}_a$. Note that in the last column of Table II where $K = 1$, $\tilde{\beta}_a$ is exactly $\hat{\beta}_a$. The second row shows the computational time to finish the compression step for different K . The third row shows the time to obtain $\tilde{\beta}_a$ by aggregating the ALCRs. The fourth row shows the relative bias of $\tilde{\beta}_a$, which is quite stable for different values of K . The last row shows the space complexity for different K . For example, in column 1 of row 4, when $K = 1000$, the number 8000 means we only need to store 1000×2 data points (x_i, y_i) for the current cell and the 1000×6 numbers for the 1000 ALCRs.

The simulation results show that the ALCR compression method can achieve almost the same accuracy as the MLE based on all the original observations, while it significantly saves computational time and space.

b) *Experiment 2.*: In the second simulation, we study the efficiency to generate logistic regression models for aggregated data cells in a data cube. Two dimensions are time and location, and the other three measure dimensions are y, x_1 and x_2 . We have 50 months' record in the time dimension and 20 cities in the location dimension. We first generate (x_1, x_2) from a two-dimensional standard normal distribution and then generate y from the Bernoulli distribution with parameter $p = \mu(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$, where $(\beta_0, \beta_1, \beta_2)^T = (1, 2, 3)^T$. For each city in each month we have 1,000 tuples (y, x_1, x_2) . Then we randomly generate 100 queries. More specifically, we first randomly select a number D from $\{1, \dots, 1000\}$, and then we randomly select D cells from the 1000 base cells (t_i, l_j) ($i = 1, \dots, 50, j = 1, \dots, 20$). Each query asks to estimate the logistic regression model in the aggregated cell composed of the selected base cells.

We compare our ALCR aggregation method to the direct MLE method by their computing time for handling these 100 queries. Table III 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 queries. Without using ALCR compression, the aggregation time is the time to compute MLE 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.

TABLE III

COMPARISON OF THE COMPUTATIONAL TIME USED FOR ANSWERING 100 QUERIES.

	ALCR compression	no compression
Preprocessing (compression)	953 seconds	0 second
Query processing (aggregation)	0.47 seconds	8 hours

c) *Experiment 3.*: In this simulation, we consider a logistic regression model with 5 predictors and one intercept term. The true regression coefficients are set as $\beta = (\beta_0, \dots, \beta_5)^T = (1, 2, 3, 4, 5, 6)^T$ and the total sample size is $N = 500,000$. We generate the predictors $\mathbf{x}_1, \dots, \mathbf{x}_5$ independently from the standard normal distribution. We simulate y_i from a Bernoulli distribution with the probability parameter

$$p_i = \frac{\exp\left(\beta_0 + \sum_{j=1}^5 \beta_j x_{ij}\right)}{1 + \exp\left(\beta_0 + \sum_{j=1}^5 \beta_j x_{ij}\right)}.$$

We partition the entire data set into $K = 20, 40, \dots, 680$ cells with equal number of observations (except the last cell in some cases). We then compute both the MLE $\hat{\beta}_a$ and the aggregated estimate $\tilde{\beta}_a$ for the entire data set. In the compression step of our aggregation method, we use the Newton-Raphson algorithm to find the MLE $\hat{\beta}_k$ in each base cell c_k . Figure 2a plots the relative bias, $\|\hat{\beta}_a - \beta\|/\|\beta\|$, of $\hat{\beta}_a$ against the partition number K . Figure 2b plots the relative difference $\|\tilde{\beta}_a - \hat{\beta}_a\|/\|\tilde{\beta}_a\|$ against the partition number K . It shows that both the relative bias and the relative difference are increasing with respect to the partition number K . This phenomenon can be well explained by our theory. Since the size of the entire data set is held fixed in this simulation, as the partition number K increases, we will have less number of observations in each base cell, and the smallest eigenvalue λ_k of the matrix $\sum_{j=1}^n \mathbf{x}_{kj} \mathbf{x}_{kj}^T$ in the k th cell becomes smaller. According to Theorem 4.2, the difference between the aggregated estimate and the MLE will be larger. This shows that for a data set of fixed size, the accuracy of our aggregated estimate lowers as the partition number increases. However, the small relative difference indicates that our aggregated estimates are still very accurate comparing to the MLE. When the size of the entire data set goes to infinity, our asymptotic theory shows that the aggregated estimate will have the same performance as the MLE.

Next, we compare our aggregated estimate $\tilde{\beta}_a$ with the estimate from subsampling. The K parameter estimates $\hat{\beta}_k$ ($k = 1, \dots, K$) from the K cells can be viewed as estimates from subsampling. For each $K = 20, 40, \dots, 680$, we calculated the bias ratios, $\|\hat{\beta}_k - \beta\|/\|\tilde{\beta}_a - \beta\|$, for $k = 1, \dots, K$. In Figure 3, we plot base-10 logarithm of different quantiles (the minimum, the 5th percentiles, the 25th percentiles, the 50th percentiles (medians), the 75th percentiles, the 95th percentiles and the maximum) of the bias ratios for each $K = 20, 40, \dots, 680$. For example, for $K = 200$, the 0.25 quantile is 0.28. It means that, among the 200 subsampling estimates $\hat{\beta}_k$, $k = 1, \dots, 200$, only 25% has $\|\hat{\beta}_k - \beta\|/\|\tilde{\beta}_a - \beta\| < 10^{0.28} = 1.9$. In other words, for $K = 200$, 75% of all $\hat{\beta}_k$ has an error 1.9 times larger than the aggregated estimate $\tilde{\beta}_a$.

From the curve of the maximums bias ratio (the top curve), we see that for all K , the maximum bias of $\hat{\beta}_k$ is about ten

Fig. 2. Relative difference of the aggregated estimate with a varying number of partitions.

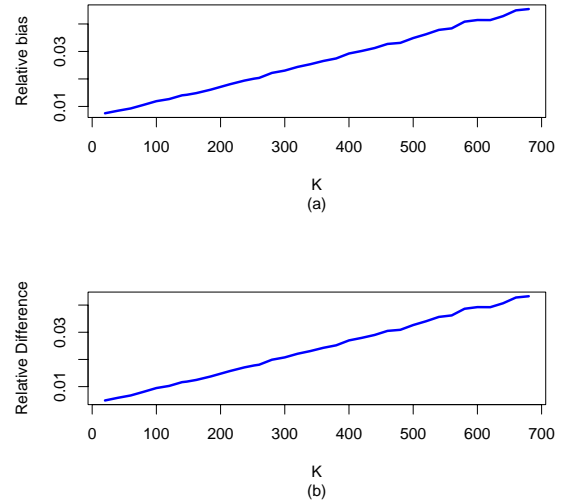
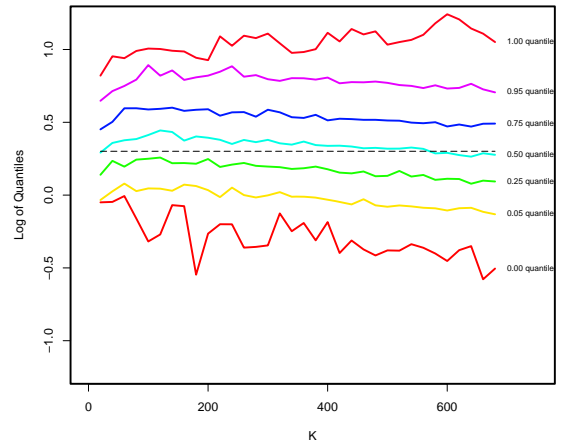


Fig. 3. Comparison of the performance of ALCR aggregation and subsampling.

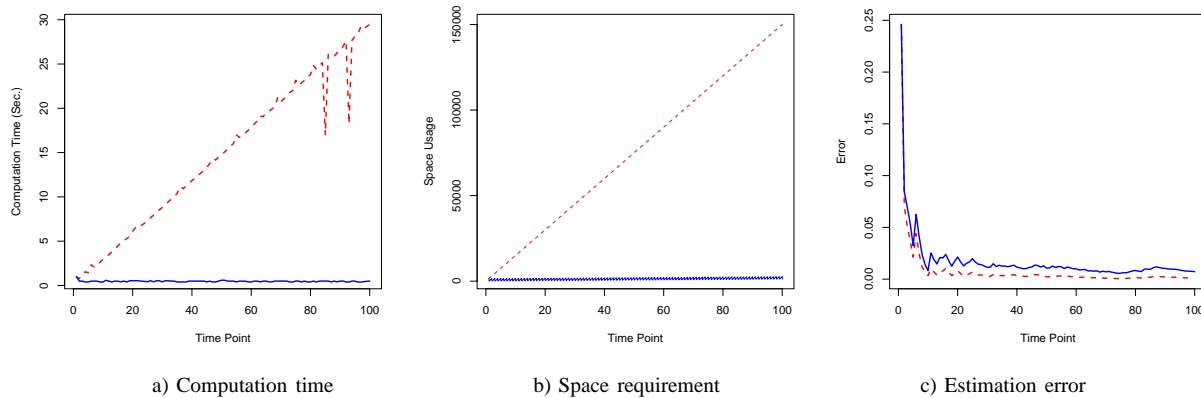


times larger than the bias of $\tilde{\beta}_a$. Similarly, from the curve of the medians of the bias ratio (the fourth curve), we see that a half of the estimates $\hat{\beta}_k$ ($k = 1, \dots, K$) have biases about twice of that of the aggregated estimates for all K (the dashed horizontal line is $y = \log_{10} 2$). Also the curve of the 0.25 quantiles shows that more than 75% of $\hat{\beta}_k$'s have larger biases than the aggregated estimates, since the 0.25 quantiles are all above the 0.0 line. Therefore, Figure 3 shows that the aggregated estimate performs better than the estimates from subsampling. Although the data is homogeneous, aggregation is necessary as the aggregated estimates are significantly more accurate than the individual estimates from each component cell.

B. Application to data streams

For many stream data applications, the size of the data is too large to be stored permanently. We are often more interested

Fig. 4. Comparison of the ALCR method (blue lines) and naive method (red lines) for logistic regression analysis of stream data.



in analyzing and predicting the data instead of the values of the raw data. We need to continuously update our model while receiving data. Therefore, even if we can store all the historical raw data, we may not be able to perform online updating in a reasonable amount of time because of the large volume of the stream data and the time costs for computing a model from large data sets. The proposed compression and aggregation method provides a solution.

We apply our method to estimate logistic regression models for stream data. The data set we used is the 50 month's data of the first city in the second simulation. We update our model for every 500 new data records. In our methods, whenever we receive 500 new data records, we compute its ACLR, update the logistic model by aggregating the ACLR with previous ACLRs, and discard the raw data. We compare the performance of our method to a "naive" method, which stores all the stream data and uses the raw data to update the model for every 500 new data records.

Figure 4 shows the computation time, space usage, and estimation error of our method and the "naive" method at different update times. As before, the error is defined as the Euclidean distance between the estimates and true parameters. In the three graphs, the solid blue lines represent our aggregation method and the dashed red lines represent the naive method.

From Figure 4a, we see that the computation time of our aggregation method roughly maintains at a constant level at different update times, while the computation time of the naive method increases very quickly. When we use our aggregation method to update the model we only need to use the new coming data to calculate the ALCR ($\hat{\beta}$, \mathbf{A}) for the cell of new coming data and aggregate all the ALCRs together to get $\tilde{\beta}$. The computation time for getting ($\hat{\beta}$, \mathbf{A}) remains small since every time we only use 500 data points and the aggregation step is very fast. In contrast, for the naive method, every time we update the model, we have to use all the raw data to calculate $\hat{\beta}$, and the computation time increases linearly when the volume of raw data becomes larger. Similar argument is valid for the memory space usage in Figure 4b. In Figure 4b the y -axis is the amount of data we need to store at different update times.

From Figure 4c we see that at the beginning the two

methods actually have the same accuracy. As the data stream flows in, the error of our method becomes larger than that of the naive method. However, errors of both methods diminish as the data stream progresses. Considering the tremendous time and space savings, our method is obviously a more preferable and scalable alternative to the naive method without compression.

C. Application to the behavioral risk factor surveillance system

We also apply our aggregation method to the Behavioral Risk Factor Surveillance System (BRFSS) survey data [12]. The BRFSS is an ongoing data collection program designed to measure behavioral risk factors in the adult population.

We use the survey data in 2004 to 2006. In the time dimension, we have three levels, year, month and day. In the location dimension, however, we have only the state level. The variable state can take 53 possible values including the 50 states and 3 districts.

In the BRFSS survey data, we are most interested in modelling the variable called `_RFHLTH`, which can take value 1 (fair or poor health) or 2 (good or better health). For simplicity, we denote it by Y . The explanatory variables are `EXERANY2`, `_EMPLOY`, `_RFSMOK3`, `DRNKANY4`, and `AGE`. The variable `EXERANY2` describes whether an interviewee has any kind of exercise during the past month, `_RFSMOK3` describes whether an interviewee is a smoker, `DRNKANY4` describes whether an interviewee has been drinking alcoholic beverages during the past month, and `_EMPLOY` describes whether one was employed or not. For all of the above variables, value 1 means yes and 2 means no. The variable `AGE` is the age of an interviewee. We consider the following model

$$\text{logit}(p) = \beta_0 + \beta_1 \text{EXERANY2} + \beta_2 \text{_EMPLOY} + \beta_3 \text{_RFSMOK3} + \beta_4 \text{DRNKANY4} + \beta_5 \text{AGE}.$$

We will use our aggregation method to estimate the parameters $\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^T$ and compare the aggregated estimates $\tilde{\beta}$ with the MLE $\hat{\beta}$. The measurement of the quality of the aggregated estimates is the relative difference $\|\tilde{\beta} - \hat{\beta}\| / \|\hat{\beta}\|$.

Fig. 5. Histogram of number of tuples in base cell.

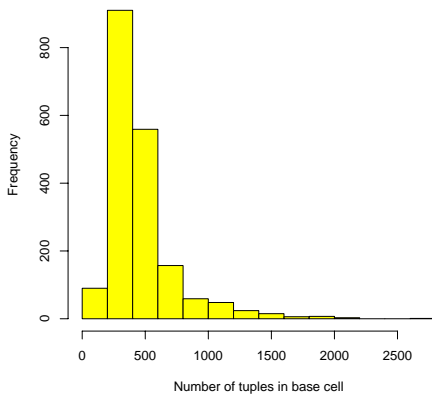
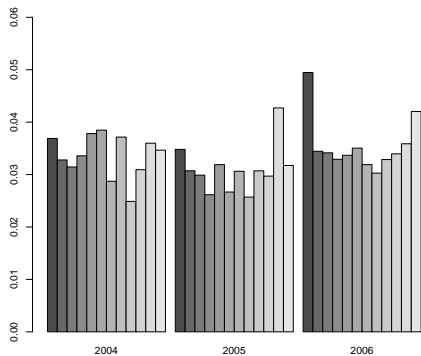


Fig. 6. Relative errors of the estimates aggregated over states for each month.



The data set in each year is partitioned into 12 subsets by month, which can be further partitioned by the 53 states. Figure 5 shows the histogram of the number of tuples in the base cells. We compute the ALCR for each state in each month and then aggregate the ALCRs. For the data in each month, we can get the aggregated estimates over the states. Figure 6 shows the relative difference of these aggregated estimates. We see that all the relative differences in Figure 6 are less than 0.05. This result on a real data set suggests that our aggregation method provides much faster computation at the cost of negligible loss of accuracy.

VI. DISCUSSION OF RELATED WORK

In this section, we compare our study with some related work and point out their differences from our work.

In general, statistical models can be put into two categories, parametric models such as linear regression and logistic regression, and nonparametric models such as probability based ensembles, naive Bayesian classifier and kernel-density-based classifiers. In parametric models, emphasis is often put on parameter estimation, such as how accurate an estimator is. On the other hand, prediction accuracy is more important in evaluating the performance of a nonparametric model. The framework of regression cube [9], [7] develops a lossless

compression and aggregation scheme to support OLAP of linear regression, a parametric model, in a data cube setting.

Another highly related work is the tool of prediction cube [5], which supports OLAP of prediction models including probability based ensemble, naive Bayesian classifier, and kernel-density classifier. The prediction cubes bear similar ideas as regression cubes in that both of them aim at deriving high-level models from lower-level models instead of accessing the raw data and rebuilding the models from scratch. A key difference is that, the prediction cube only supports models that are distributively decomposable or algebraically decomposable [5], whereas the regression models in our study are not. Also, the prediction cubes deal with the prediction accuracy of nonparametric statistical models, whereas our compression theory is developed for parameter reconstruction of logistic regression models.

The above developments all focus on lossless computation for data cubes. Alternatively, asymptotically lossless computation that provides good approximations to the desired results is also acceptable in many applications when efficient storage and computation is attainable. An approximation technique called quasi-cube uses the loglinear model, a parametric model, to characterize regions of a data cube [3]. Efficient storage and fast computation are achieved by storing the parameters of the loglinear models instead of the original data. In quasi-cubes, the desired computation is done based on approximations to the original data provided by the loglinear model. However, it is difficult to quantify the approximation errors in a quasi-cube.

Our paper considers aggregation operations without accessing the raw data. Palpanas, Koudas, and Mendelzon [20] have considered the reverse problem, which is to derive original raw data from the aggregates. An approximative estimation algorithm based on maximum information entropy is proposed [20]. It will be interesting to study the interactions of these two complimentary approaches.

Dimension hierarchies, cubes, and cube operations are formally introduced by Vassiliadis [23]. Lenz and Thalheim [17] proposed to classify OLAP aggregation functions into distributive, algebraic, and holistic ones. In data warehousing and OLAP, much progress has been made on the efficient support of standard and advanced OLAP queries in data cubes, including selective cube materialization [15] and intelligent roll-up [22]. However, the measures studied in previous OLAP systems are usually single values, not regression models.

VII. CONCLUSIONS

In this paper, we have proposed an asymptotically lossless compression technique to support efficient logistic regression analysis in a data cube environment. We have developed a compression scheme that compresses a data cell into a compressed representation whose size is independent of the size of the cell. Under regularity conditions, we have proved that the aggregated estimator is strongly consistent and asymptotically error-free.

The proposed technique allows us to quickly perform OLAP operations and generate logistic regression models at any

level in a data cube without retrieving or storing the raw data. Our experimental studies show that our compression and aggregation method can significantly save computing time with little loss of accuracy. Moreover, the aggregation error diminishes as the size of the data cube increases. We are currently extending the technique to more general situations, such as quasi-likelihood estimation for generalized statistical models.

ACKNOWLEDGMENT

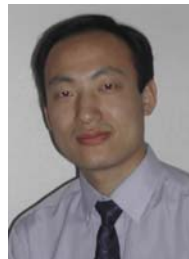
The authors would like to thank the reviewers for their comments to improve this paper. This work is supported by an NSF grant IIS-0713109, a Microsoft Research New Faculty Fellowship, and a Department of Energy ECPI grant.

REFERENCES

- [1] A. Agresti. *An Introduction to Categorical Data Analysis*. Wiley, 1996.
- [2] A. Agresti. *Categorical Data Analysis*. John Wiley and Sons, New Jersey, 2nd edition, 2002.
- [3] D. Barbara and X. Wu. Loglinear-based quasi cubes. *Journal of Intelligent Information Systems*, 16:255–276, 2001.
- [4] 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.
- [5] B. Chen, L. Chen, Y. Lin, and R. Ramakrishnan. Prediction cubes. In *Proceedings of the 31st VLDB Conference*, pages 982–993, 2005.
- [6] K. Chen, I. Hu, and Z. Ying. Strong consistency of maximum quasi-likelihood estimators in generalized linear models with fixed and adaptive designs. *The Annals of Statistics*, 27:1155–1163, 1999.
- [7] 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.
- [8] Y. Chen, G. Dong, J. Han, J. Pei, B. W. Wah, and J. Wang. Olaping stream data: Is it feasible? In *Proc. Workshop on Research Issues in Data Mining and Knowledge Discovery, ACM SIGMOD*, pages 53–58, 2002.
- [9] 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.
- [10] Y. Chow and H. Teicher. *Probability Theory*. Springer, New York, 2nd edition, 1988.
- [11] L. Fahrmeir and H. Kaufmann. Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *The Annals of Statistics*, 13:342–368, 1985.
- [12] Centers for Disease Control and Prevention. *Behavioral Risk Factor Surveillance System Survey Data*. U.S. Department of Health and Human Services, Centers for Disease Control and Prevention, 2006.
- [13] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatarao, 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.
- [14] 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.
- [15] V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In *Proc. ACM-SIGMOD Int. Conf. Management of Data*, pages 205–216, 1996.
- [16] S. A. Julious and M. A. Mullee. Confounding and simpson’s paradox. *British Medical Journal*, 309:1480–1481, 1994.
- [17] H. Lenz and B. Thalheim. OLAP databases and aggregation functions. In *Proceedings of the 13th International Conference on Scientific and Statistical Database Management*, pages 91–100, 2001.
- [18] 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.
- [19] P. McCullagh and J. A. Nelder. *Generalized linear models*. Chapman and Hall, London, 2nd edition, 1989.
- [20] T. Palpanas, N. Koudas, and A. O. Mendelzon. Using datacube aggregates for approximate querying and deviation detection. *IEEE Trans. Knowl. Data Eng.*, 17(11):1465–1477, 2005.
- [21] 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.
- [22] G. Sathe and S. Sarawagi. Intelligent rollups in multidimensional OLAP data. In *Proc. Int. Conf. on Very Large Data Bases*, pages 531–540, 2001.
- [23] P. Vassiliadis. Modeling multidimensional databases, cubes and cube operations. In *Proc. 10th International Conference on Scientific and Statistical Database Management*, pages 53–62, 1998.



Ruibin Xi received the master’s degree in Mathematics from Washington University in St. Louis. He is currently working toward the PhD degree in the Department of Mathematics, Washington University in St. Louis. His research interests include statistical computing, massive data analysis, Bayesian statistics and data mining.



Nan Lin is an Assistant Professor of Mathematics and Biostatistics at the Washington University in St. Louis. He received the Ph.D. in Statistics from University of Illinois at Urbana-Champaign in 2003. He has also worked as a Postdoctoral Associate at Yale University School of Medicine from 2003 to 2004. His research interest includes statistical computing, massive data analysis, robust statistics, bioinformatics and psychometrics. He is a member of the American Statistical Association and the International Chinese Statistical Association.



Yixin Chen is an Assistant Professor of Computer Science at the Washington University in St. Louis. He received the Ph.D. in Computing Science from University of Illinois at Urbana-Champaign in 2005. His research interests include nonlinear optimization, constrained search, planning and scheduling, data mining, and data warehousing. His work on constraint partitioning and planning has won First Prizes in optimal and satisficing tracks in the International Planning Competitions (2004 & 2006) and the Best Paper Award at the International Conference on Tools for AI (2005). His work on data clustering has won the Best Paper Award at the International Conference on Machine Learning and Cybernetics (2004) and the Best Paper nomination at the International Conference on Intelligent Agent Technology (2004). He has received an Early Career Principal Investigator Award from the Department of Energy (2006) and a Microsoft Research New Faculty Fellowship (2007).