Distributed Multivariate Regression Based on Influential Observations

Hang Yu
School of Computing
National University of Singapore
yuhang@comp.nus.edu.sg

Ee-Chien Chang
School of Computing
National University of Singapore
changec@comp.nus.edu.sg

ABSTRACT

Large-scale data sets are sometimes logically and physically distributed in separate databases. The issues of mining these data sets are not just their sizes, but also the distributed nature. The complication is that communicating all the data to a central database would be too slow. To reduce communication costs, one could compress the data during transmission. Another method is random sampling. We propose an approach for distributed multivariate regression based on sampling and discuss its relationship with the compression method. The central idea is motivated by the observation that, although communication is limited, each individual site can still scan and process all the data it holds. Thus it is possible for the site to communicate only influential samples without seeing data in other sites. We exploit this observation and derive a method that provides tradeoff between communication cost and accuracy. Experimental results show that it is better than the compression method and random sampling.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*

Keywords

Distributed data mining, multivariate linear regression, learning curve, sampling.

1. INTRODUCTION

Recently, Distributed Data Mining (DDM) has emerged as a popular means of extracting statistical information in a distributed environment. Several systems have been developed for DDM. Here are some examples: the JAM system developed by Stolfo et.al.[11], the Kensington system developed by Guo et.al.[5], and BODHI developed by Kargupta et.al.[7]. The detailed description of these systems can be found in Bailey et.al.[1].

Multivariate regression is widely used to analyze data in social and natural sciences. Sometimes, these data are logically and physically distributed. In the model studied in this paper, the data are vertically distributed in individual site as shown in Figure 1.

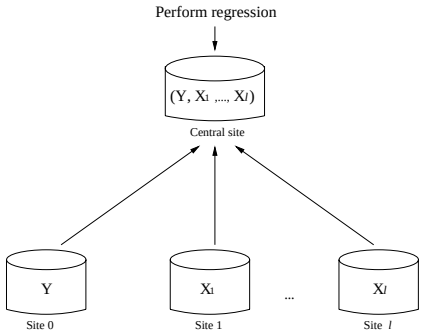

Figure 1: Distributed Data Model

Although multivariate regression, in particular linear regression, is well-studied, the additional constraints imposed in the distributed environment pose a different challenge. One of the concerns in DDM is the cost of communication[2], which is the focus of this paper.

In the non-distributed environment, a method to tradeoff accuracy with computing resources is by reducing the samples size[8]. Because the sample points are not known until they are observed, thus, random sampling is the only option. In DDM, although random sampling can be applied, it does not exploit the fact that each local site has access to all its data. In other words, because each site can see and analyze all the data stored in its site, it is possible for the sites to selectively transmit data to the central site, so as to achieve a overall better estimate given the constraint on communication bandwidth. The above observation motivates our proposed scheme.

Outline

The rest of this paper is organized as follows. We first discuss the related work. In section 3 we give an overview of multivariate regression(MR) and our model. The main observation is given in Section 4. Section 5 gives the proposed algorithm. Section 6 gives the communication overhead

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGKDD '03 Washington, DC, USA
Copyright 2003 ACM 1-58113-737-0/03/0008 ...$5.00.

analysis. Section 7 presents the experimental results showing comparison of the proposed algorithm with two known methods. Comparison with compression method is given in Section 8. Section 9 concludes the paper.

2. RELATED WORK

Existing DDM work may be grouped into five basic categories, Meta-learning and Stacking, Collective Data Mining, Distributed Association Rule Learning, Distributed Clustering[12] and other DDM techniques[6]. However there is not much work on Distributed Regression.

Our work is most similar to that given by Daryl et. al. [6], who gave a wavelet compression-based method. They proposed to compress each vertical data set using wavelet transform. Specifically, one dimensional wavelet transform is first applied to the vertical data; next, only the large absolute value coefficients are kept, and the remainings are set to zero. Our proposed method is very similar to the above. A main difference is the motivation. Instead of compression, our method is guided by sampling. Based on the estimation theory, we argue that large absolute value sample points are "good" sample points and they give better estimate than random samples. This leads to the improvement which selects the "most influential" sample points in each vertical data set, and a better overall performance. Because wavelet transform is linear, our method complements the compression-based method in the sense that it can be applied to the compressed data.

3. BACKGROUND AND MODEL

Background on multivariate regression can be found in[9, 10].Here are some definitions on the regression model with fixed independent variables. Let X represent the independent variables and Y the real-valued response variable. Y is a $n \times 1$ matrix of sample values, X=$[X_0, X_1, ...X_\ell]$ is a $n \times (\ell+1)$ matrix where each column represents the sample data for one independent variable and the first column X_0 consists of the constant ones. Thus, each row contains the set of observed values of the independent features for one sample. Using matrix notation,we can express the function relation between Y and X as,

$$Y = XB + \epsilon, \tag{1}$$

where B=$[\beta_0, \beta_1, ...\beta_\ell]^T$ is a $(\ell+1) \times 1$ column vector of regression coefficients to be estimated. The ϵ is the error in the measured values of Y and usually assumed to follow a multivariate normal distribution,

$$\epsilon \sim N(0, \sigma^2 I). \tag{2}$$

The standard approach to estimate B is accomplished using the least square estimator. If the matrix X^TX is invertible then the least square estimator of the regression coefficients is

$$\hat{B} = (X^TX)^{-1}X^TY. \tag{3}$$

The property of $\hat{B}$ that we are interested in is as follow:

- The covariance matrix for $\hat{B}$ is given by $\sigma^2(X^TX)^{-1}$.

Distribution Model

We assume that there are $\ell+1$ local sites $L_0, L_1, \ldots, L_\ell$ storing the distributed data sets and one central site C computing the estimator $\hat{B}$. The local site L_0 stores the response variable Y. Each other local site stores the independent variable, specifically, L_i stores the column vector X_i for $i = 1, 2, \ldots, \ell$. The estimator is to be computed in the central site C. Thus, information on X and Y has to be sent to C. We want to find a method that gives a good estimate of B, while using some fixed amount of communication cost. The performance of the estimator is measured by its variance (that is, the expected $\|B - \hat{B}\|_2^2$). Details on the accounting of communication cost will be given in Section 6.

4. INFERENCE ON ESTIMATOR $\hat{B}$

In this section, we describe the main observation that motivates our algorithm: the influential samples occur at the two extreme ends of the data set. In the previous section, we have $cov(\hat{B}) = \sigma^2(X^TX)^{-1}$. Thus the configuration of X^TX is important in the estimation of the $\beta_j's$. The slope coefficient $\hat{\beta}_j (j \neq 0)$ has sampling variance

$$V(\hat{\beta}_j) = \left(\frac{1}{1-R_j^2}\right)\left(\frac{\sigma^2}{\sum_{i=1}^n (X_{ij} - \bar{X}_j)^2}\right) \tag{4}$$

where X_{ij} is the ith sample value in X_j, R_j^2 is the multiple correlation from the regression of X_j on all the other $X's$[3]. Typically, hypotheses about $\hat{\beta}_0$ are of less interest than those of the slope coefficient $\hat{\beta}_j (j \neq 0)$, since our first priority is to determine whether there is a linear relationship between Y and X.

Now we investigate Eq.(4) and identify terms that can be improved by using local sites. The first factor in Eq.(4) is called the *variance-inflation factor*. It is small when X_j is not strongly correlated with other $X's$. Note that its value depends on the sample values of all the other independent variables. However the second factor depends solely on the sample values in X_j. According to the second factor, a large variance of X_j gives a smaller variance in estimation of $\hat{\beta}_j (j \neq 0)$.

In the non-distributed environment, information on X and Y is obtained only after they are observed. Thus, the only way to reduce the number of samples is by random selection. However in the distributed environment, each local site L_j can see all the values of X_{ij}, for all i. Thus it is possible to reduce the second factor.

Now we ask the following question: given a set of real values K=$\{x_1, x_2, ...x_\ell\}$, which subset $K^{'}$ of k values has maximum "variance" $V(K^{'})$,

$$V(K^{'}) = \sum_{x \in K^{'}} (x - \bar{K^{'}})^2. \tag{5}$$

First, note that among all random real-valued variables having range $[a, b]$, the one with the maximum variance is the random variable that equals a with probability 1/2 and equals b with probability 1/2 [9].

In our model, the sample values $X's$ are fixed. We need to choose influential subsets from the samples to achieve minimum variance of the $\beta_j's$ when the size of the subset is given. Observations whose inclusion or exclusion results in substantial changes in the fitted model (coefficients, fitted values) are usually said to be influential. Here, we call the set of samples that gives lower variance of $\hat{B}$, *influential*. Based on the following theorem, we can purposively choose a given

size of influential samples and achieve better regression performance than random sampling method.

Theorem 1

For an ascending-ordered set $K = \{x_1, x_2, ..., x_\ell\}$ with ℓ real values in the range $[a, b]$. Let $K_{i,k}$=$\{x_1, ..., x_i, x_{\ell-k+i+1}, ..., x_\ell\}$ be a set of size k where $1 \leq i < \ell$, and $0 < k < \ell$
Among all possible subsets of K with a given size k, the subset $K^{'}$ which gives the largest $V(K^{'})$ is $K_{i,k}$ for some i. Furthermore, for any fixed k, the function $V(K_{i,k})$ with respect to i is convex.

The above theorem states that the influential samples occur at the two extreme ends. Because the function $V(K_{i,k})$ with respect to i is convex, we can find the influential samples $K_{i,k}$ efficiently using binary search.

5. PROPOSED ALGORITHM

In this section, we present the algorithm to distributed multivariate linear regression based on influential samples. Besides that, we also describe two other methods, random sampling and compression-based method, which are used for performance comparison.

Algorithm 1

---

1. Each local site L_i, where $i = 1, 2, \ldots, \ell$ performs:
   (a) Determine the k influential samples in X_i using Theorem 1. Let $\mathcal{I}_i$ be the indices of these k samples.
   (b) Send $\mathcal{I}_i$ to the central site C.
2. After receiving all the $\mathcal{I}_i$'s, the central site C performs:
   (a) Determine $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2 \ldots \cup \mathcal{I}_\ell$.
   (b) Send $\mathcal{I}$ to L_0.
   Send $(\mathcal{I} - \mathcal{I}_i)$ to L_i for $i = 1, 2, \ldots, \ell$.
3. Each local site L_i, where $i = 0, 1, \ldots, \ell$, sends the samples whose indices are in $\mathcal{I}$ to the central site C.
4. The central site C computes the estimator $\hat{B}$.

The communication cost will be described in Section 6. The computational cost in step 1(a) is linear with respect to the number of samples n. If the samples are already sorted, then, using binary search, the influential samples can be found in $O(\log k)$ time, and in $O(k)$ if we include the reporting time. If the samples are not sorted, we can use the selection algorithm to list out the $2k$ largest absolute values. This can be done in linear $O(n)$ time. With the $2k$ largest absolute values, the influential samples can be similarly found using binary search.

We compare our method with the following two methods.

1. **Compression-based**. This is the method proposed in [6]. Each local site first performs 1-d wavelet-packet decomposition on the samples it holds. Next, it sends a predefined number of coefficients with the largest absolute values to the central site. For example, L_1 performs 1-d wavelet on X_1, and sends the k coefficients with the largest absolute value to C, where k is a predefined constant. After receiving the large coefficients from all local sites, the central site C computes the estimator. Coefficients that C does not receive are set to zero.
2. **Random sampling**. This simple method randomly chooses a few samples from each site and sends them to the central site for processing. In other words, first, a few rows from X are randomly chosen. The chosen rows, together with the respective samples in Y are then used to compute the estimator.

6. COMMUNICATION OVERHEAD

In this section, we analyze the communication requirements. Let the size of one sample be δ, and the communication cost to transmit a sample be δ. Thus, the size of Y is $n\delta$ and the size of X is $n\ell\delta$. Although X is a $n \times (\ell+1)$ matrix, the column X_0 consists of the fixed constant one's and is not required to be explicitly stored. Therefore, if all the samples are used for the regression, then the communication cost is $n(\ell+1)\delta$. Besides the actual samples, in some methods, their locations may be required to be explicitly stored. Each sample set, i.e. each row of the X can be identified by an index or the row number. Let γ be the size required to represent the indices. For example, if $n = 2^8$, then $\gamma = 8$ bits.

To reduce the number of samples, additional communication is required to synchronize all the sites so that all use the same set of samples. We call the size of the data, other than the samples, that are transmitted for synchronization the *overhead cost*. We ignore the constant overhead required during hand shaking and communication housekeeping, which are required by all methods.

1. Random sampling. This method incurs the lowest overhead. The local sites and the central site first agree on a random seed. Based on this seed, a random sequence of indices is generated in each site. Because the seed is the same, so are the generated sequences in all the sites. Next, each local site sends the corresponding samples to the central site. Only constant overhead is incurred.
2. Compression-based. Suppose that each local site sends the k largest absolute value coefficients to the central site. Besides the value of the sample points, the indices of these coefficients are required to be sent. Thus, the communication cost is $k(\ell+1)(\delta+\gamma)$, and the overhead is $k(\ell+1)\gamma$.
3. The proposed method. There are three rounds of communication. (1) Each local site sends the indices of the influential samples to the central cite; (2) The central site sends to each local site the indices of additional samples it requires from the local site; (3) Each local site sends the samples to the central site. Suppose the number of all influential samples is $\widetilde{k}$ (i.e. the total number of elements in $\mathcal{I}$ in Algorithm 1, not to be confused with the k in step 1), then the total number of indices sent to and from the central site is $(\ell+1)\widetilde{k}$. The number of samples sent in round 3 is also $(\ell+1)\widetilde{k}$. Thus, the communication cost is $\widetilde{k}(\ell+1)(\delta+\gamma)$, which is same as that of compression-based method, and the overhead is $\widetilde{k}(\ell+1)\gamma$.

Depending on the data type, the ratio of δ and γ varies. The size γ is arguably small. For example, the samples value might be a 32-bit single precision floating point, and the number of samples is 2^{16}, which gives $\gamma = 16$ bits. In the experimental results, we take $\delta = 1$ and $\gamma = 0$. For non-zeros value of γ, the performance can be easily inferred from the graphs.

7. EXPERIMENTAL RESULTS

We use both synthetic and benchmark data sets to compare the resulting model statistics with random sampling and compression-based method.

7.1 Synthetic data

The synthetic data are generated according to following procedure:

1. Randomly generate data X which follows a multivariate normal distribution. Recall that X_0 consists of the constant one's.
2. Randomly generate $B = [\beta_0, \beta_1, ..., \beta_l]^T$.
3. Randomly generate error ϵ which follows a multivariate normal distribution.
4. Compute Y as $Y = XB + \epsilon$.

The experimental results are presented in Figure 2 and Figure 3. The size of the data is $n = 512$ and the number of variables is $\ell = 7$. We repeat the experiment 40 times. Each time we generate different X, B, ϵ and Y, i.e. 40 sets of data are generated. From these 40 sets of data, we compute the average to estimate the variance of estimator. Figure 2 shows the log-scaled variance of estimator as the communication cost increases. Recall that the variance of estimator is the expected $\|B - \hat{B}\|_2^2$. The communication cost is computed using the size of a sample $\delta = 1$ and ignoring the size of index $\gamma = 0$. The figure shows that proposed method performs better than compression-based method and random sampling. Figure 3 shows the log-scaled residual sum of squares($\|Y - \hat{Y}\|_2^2$) as the communication cost increases. The proposed method performs the best.

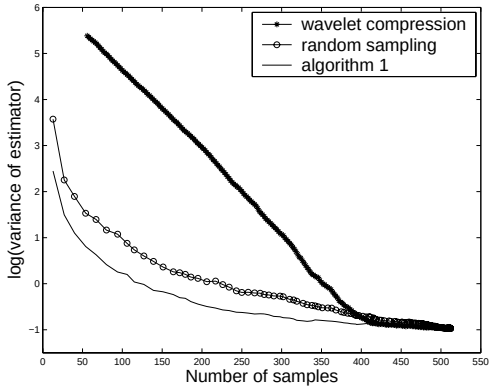

Figure 2: Performance comparison for synthetic data using variance of estimator.

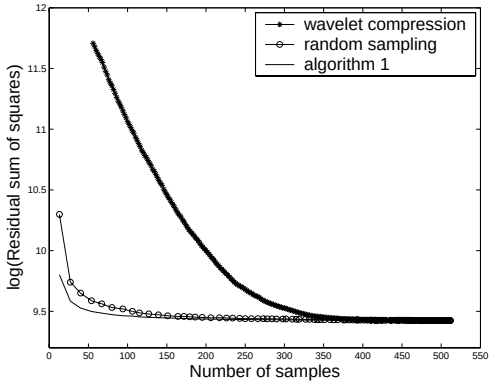

Figure 3: Performance comparison for synthetic data using residual sum of squares.

Y	X_1	X_2
100.08	48.83	63.42
89.97	41.09	60.11
113.22	51.83	70.12
101.09	34.82	55.38
82.22	22.54	52.36
105.13	40.58	60.37
105.81	41.02	57.56
91.32	34.04	50.86
97.38	48.65	63.20
97.38	47.80	56.35

Table 1: Energy expenditure data. The data consists of 104 samples. This table shows part of the data.

7.2 Benchmark data

In order to test the validity of our method, we apply it to benchmark data set that has been analyzed using standard MR techniques. For real life data, the true B is unknown, we use the value estimated from **all** the samples as the ground truth.

The data are from an investigation[4] concerning the energy expenditure for human subjects at a given physical activity and for a given time period. We choose the largest set of the data from the investigation, consisting of 104 women. The variables measured for the i-th subject were total energy expenditure at rest for a 24 hour period Y, mass of fat tissue X_1 and mass of fat-free tissue X_2. Table 1 shows a subset of the data. The experimental results are presented in Figure 4 and Figure 5. Figure 4 shows the log-scaled variance of estimator as the communication cost increases. Figure 5 shows the log-scaled residual sum of squares as the communication cost increases. The proposed method performs the best.

8. COMPARISON WITH COMPRESSION METHOD

The main difference between compression-based method and the proposed method is the motivation: the proposed method is guided by sampling. Yet, in a certain way, both

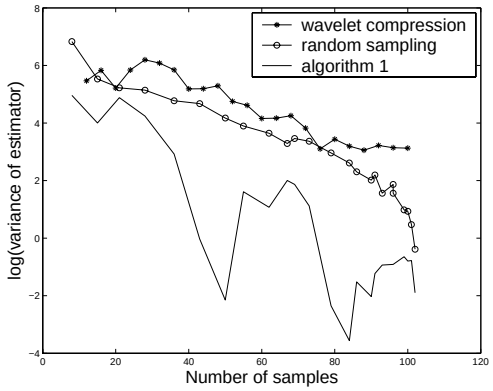

Figure 4: Performance comparison for energy expenditure data using variance of estimator.

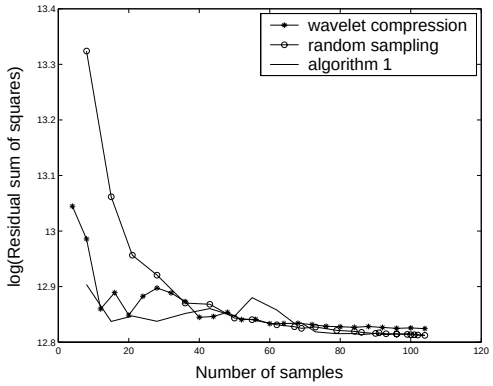

Figure 5: Performance comparison for energy expenditure data using residual sum of squares.

methods are similar. Because wavelet transform is linear, Algorithm 1 can be applied in the transformed domain. This is simply done by treating the wavelet coefficients as samples and applying Algorithm 1 on them. If the distribution of the wavelet coefficients follows a zero mean normal distribution, then very likely, most of the influential samples (or coefficients) are the coefficients of k largest absolute value, where k is the number of samples to be used. In other words, the influential samples are likely to be the same samples picked by the compression-based method.

Note, however, that Algorithm 1 and the compression-based method are not equivalent even if the k influential samples are the k largest absolute values. Under the compression-based method, samples with small absolute value are set to zeros. For example, consider a row $(1, x_1, x_2, \ldots, x_\ell)$ in X. Suppose only $|x_2|$ is large, then the values used in the estimation are $(1, 0, x_2, 0, \ldots 0)$. On the other hand, under Algorithm 1, if x_2 is the only influential sample, all other values in the same row are requested by the central site and are used in the estimation.

The compression-based method exploits the compressibility of each feature, i.e. the column vector X_i. As such, it performs poorly if there are not coherence within each column X_i. For example, in the synthetic data (Figure 2), the samples are independently generated, and thus the performance of the compression-based method is relatively poor. In some real-life data, there might be coherence among the samples. For example, in Figure 4, the performance of the compression-based method improves significantly. In this case, our method is complementary in the sense that it can also make use of the coherence.

9. CONCLUSIONS AND REMARKS

In this paper we present a method for distributed multivariate regression using influential samples. With the intuition that, although communication is limited, each individual site can still scan and process all the data it holds, we propose a technique for the site to communicate only influential samples without seeing data in other sites. Based on the estimation theory, we argue that large and small sample points are influential sample points and they give better estimate than random samples. We also discuss its relationship with the compression method in[6]. Experimental results show that our method performs better.

There are many possible extensions of this work. Currently, our method attempts to minimize the second factor in (4). It may be possible to minimize the first factor using information gathered in a few rounds of communication. Another possible extension is to adapt the method to clustered samples. In is also interesting to study whether incorporating quantization of the sample values will further reduce communication cost.

ACKNOWLEDGEMENTS

We would like to thank all anonymous reviewers for detailed and insightful comments.

10. REFERENCES

[1] Stuart M. Bailey, Robert L. Grossman, Harinath Sivakumar, and Andrei L. Turinsky. Papyrus: A system for data mining over local and wide area clusters and super-clusters. *Proceedings of Supercomputing*, 1999.

[2] Chris Clifton. Privacy preserving distributed data mining. *Purdue Research Foundation*, August 2002 through July 2003.

[3] John Fox. *Applied Regression Analysis, Linear Models, and Related Methods*. Sage Publications.

[4] L. Garby, J.S. Garrow, B. Jorgensen, O. Lammert, K. Madsen, P. Sorensen, and J. Webster. Relation between energy expenditure and body composition in man: specific energy expenditure in vivo of fat and fat-free mass. *European Journal of Clinical Nutrition*, pages 301–305, 1988.

[5] Y. Guo, S. M. Rueger, J. Sutiwaraphun, and J. Forbes-Millott. Meta-learnig for parallel data mining. *Seventh Parallel Computing Workshop*, 1997.

[6] Daryl E. Hershberger and Hillol Kargupta. Distributed multivariate regression using wavelet-based collective data mining. *Journal of Parallel and Distributed Computing*, 61(3):372–400, 2001.

[7] H. Kargupta, I. Hamzaoglu, and B. Stafford. Scalable distributed data mining using an agent based architecture. *3rd International Conference on the Knowledge Discovery and Data Mining*, 1997.

[8] Foster J. Provost, David Jensen, and Tim Oates. Efficient progressive sampling. *5th ACM SIGKDD*, pages 23–32, 1999.

[9] Alvin C. Rencher. *Linear models in statistics.* Wiley, 2000.

[10] Alvin C. Rencher. *Methods of multivariate analysis.* Wiley-Interscience, 2002.

[11] S. Stolfo, A. L. Prodromidis, , and P. K. Chan. Jam: Java agents for meta-learning over distributed databases. *3rd International Conference on Knowledge Discovery and Data Mining*, 1997.

[12] Bin Zhang and George Forman. Distributed data clustering can be efficient and exact. *SIGKDD Explorations*, 2000.