M-004

Preserving-preserving Multi-party Statistical Computation

蘇　春華†　　櫻井幸一‡
Chunhua Su　Kouichi Sakurai

Abstract

The rapid development of Internet provides us tremendous opportunities for cooperative computations. As a new technology, Data Mining can extract valuable knowledge from large among of data. And statistical computation is a major tool used in data mining. However, the collected data may contain the sensitive Information or privacy of individual or company. This privacy concern may prevent them to share their data for the cooperation. In paper, we proposed some protocols for privacy-preserving statistical computation over the distributed databases in the network environment. Our protocols are based on the data perturbation and cryptographic techniques.

1. Introduction

The statistical computation is very frequently used for analysis of the data. In modern world, many organizations or companies have to do joint work to get mutual benefit of their individual data. The growth of the Internet opens up tremendous opportunities for such cooperations, Most data mining tools operate by gathering all data into a central site, then running an algorithm on those gathered data. However, in this real world, many organizations and the individual will have concern about their own privacy, they may be reluctant to share their own data to go on data mining unless there is privacy preserving data mining technology to make sure their privacy won't be violated or misused by other parties.

Foe example, two company A and B each have their own database about their own costumers' privacy, or two hospital want to do joint analysis on some disease research while preserving the patients' private in the database records. Under such conditions, we have to provide a secure environment to preserve the privacy of all the participants.

In this paper, we use some cryptographic techniques to construct some secure multi-party computation protocols to do the statistical computation in the distributed network environments.

Section 2 is a brief review of the relative work and section 3 is about some preliminaries. We will give our protocols in section 4 and discuss their complexity in section 5. Finally, we make conclusions and claim thefuture works in section 5

2. Relative Works

Secure multi-party computation (MPC) protocols allow a set of n players to securely compute any target function on their private inputs, where the following properties must be satisfied: ***privacy***, meaning that the corrupted players do not learn any information about the other players' inputs (except for what they can infer from the function output), and ***correctness***, meaning that the protocol outputs the correct function value, even when the malicious players misbehave.

The problem that we have described above is a case of secure two-party computation. This notion was first investigated by Yao who proposed a general solution [1]. The two-party case was subsequently generalized to the multi-party case [2, 3]. In all of these results, they just provide general solutions without telling us how to solve the application problem.

3. Preliminaries

3.1 Secure Multi-party Computation

In Secure Multi-party Computation, we always assume that *semi-honest* model exists. *Semi-honest* model means all the parties will faithfully follow the protocol and they may infer the other party's private information from the intermediate message during the execution.

In Secure Multi-party Computation, we always assume that *semi-honest* model exists. *Semi-honest* model means all the parties will faithfully follow the protocol and they may infer the other party's private information from the intermediate message during the execution. We say that π privately computes a function f if there exist probabilistic, polynomial-time algorithms S_1 and S_2 such that(The views of Party 1 and Party 2 during an execution of $\pi(x_1, x_2)$ is denoted by $view_1^{\pi}(x_1, x_2)$ and $view_2^{\pi}(x_1, x_2)$, respectively):

$$\{S_1(x_1, f_1(x_1, x_2)), f(x_1, x_2)\} \equiv \{view_1^{\pi}(x_1, x_2), output^{\pi}(x_1, x_2)\}$$

$$\{S_2(x_2, f_2(x_1, x_2)), f(x_1, x_2)\} \equiv \{view_2^{\pi}(x_1, x_2), output^{\pi}(x_1, x_2)\}$$

Where $\equiv$ denotes the computational indistinguishability.

3.2 Statistical Computation Background

In this paper, we focus on there kinds of statistical analysis computation:

- Mean Value: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Correlation Coefficient between x and y: It is a useful measure to analyze the linear relationship between x and y. We compute this correlation coefficient as following:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sqrt{\sum_{i=1}^{n} x_i^2 - n\bar{x}\sum_{i=1}^{n} y_i^2 - n\bar{y}}}$$

- Linear Regression Line analysis: This measure is used to get the closest line to the data.

$$y = bx + (\bar{y} - b\bar{x}), \quad where$$

$$b = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}$$

†九州大学大学院システム科学府, su@itslab.csce.kyushu-u.ac.jp

‡九州大学大学院システム科学研究院, sakurai@csce.kyushu-u.ac.jp

4. Secure Statistical Computation Protocols

4.1 Secure Mean Computation protocol

There are n clients want to interested in the sum of m selected numbers in the database (whose indices it is assumed to know,), but all the client does not wish to reveal his own database privacy. So we use a linear-communication solution that provides database privacy and client privacy using semantically secure homomorphic encry-ption. Many such systems exist in the works by Benaloh [4], Naccache and Paillier[5], to mention a few. A useful property of homomorphic encryption schemes is that an addition operation can be conducted based on the encrypted data without decrypting them:

$$\prod_{i=1}^{n} E(I_i)^{x_i} = E(\sum_{i=1}^{n} I_i x_i)$$

We can use this property to construct our Secure Mean Computation. For computing the mean value privately, The client encrypts its array of indices using the homomorphic cryptosystem and sends the encryptions $E(I_1), E(I_2), \ldots, E(I_n)$ to the server. After the decryption, we can get the final result by dividing the sum with m, which is the number of selected items.

4.2 Secure Scalar Product Protocol

Secure Scalar Product Protocol is a very important sub-protocol for the following computation. We use XY to denote the scalar product of two vectors X = ($x_1,\ldots, x_n$) and Y = ($y_1,\ldots,y_n$), $XY = \sum_{k=1}^{n} x_k y_k$. We use the proposal in [6] as a sub-protocol to construct our protocol.

Secure Scalar Product Protocol:

PRIVATE INPUTS: Private vectors $x, y \in \mathbb{Z}_m^N$.

PRIVATE OUTPUTS: Shares $s_A + s_B \equiv x \cdot y \mod m$

1. Setup phase. Alice does:
   Generate a private and public key pair (sk, pk).
   Send pk to Bob.
2. Alice does for $i \in \{1,\ldots,N\}$:
   Generate a random new string r_i.
   Send $c_i = \mathrm{Enc}_{\mathrm{pk}}(x_i; r_i)$ to Bob.
3. Bob does:
   Set $w \leftarrow \prod_{i=1}^{N} c_i^{y_i}$.
   Generate a random plaintext s_B and a random nonce r'.
   Send $w' = w \cdot \mathrm{Enc}_{\mathrm{pk}}(-s_B; r')$ to Alice.
4. Alice does: Compute $s_A = \mathrm{Dec}_{\mathrm{sk}}(w') = x \cdot y - s_B$.

4.3 Secure Protocol for Computing Correlation Coefficient and Linear Regression Line

We assume that there are two parties, Party A and Party B want to find out the correlation coefficient and linear regression linear in their joint database of D_A=($x_1,\ldots,x_n$) and D_B=($y_1,\ldots,y_n$) owned by Party A and B, respectively.

Correlation Coefficient:

Let $u = \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}$ and $v = \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}$. To compute the coefficient r, we can transform the formulation in 3.2 into:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$= \sum_{i=1}^{n} \frac{(x_i - \bar{x})}{u} \frac{(y_i - \bar{y})}{v}$$

$$= (\frac{x_1 - \bar{x}}{u}, \ldots, \frac{x_n - \bar{x}}{u}) \cdot (\frac{y_1 - \bar{y}}{v}, \ldots, \frac{y_n - \bar{y}}{v})$$

With this transformation, we can use the sub-protocol mentioned in 4.2 to compute the correlative coefficient.

Linear Regression Line: Let $w = \sum_{i=1}^{n} x_i^2 - n\bar{x}^2$, and this w can be computed by Party A alone. So what they need to compute

$$b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}$$

$$= (\frac{x_1 - \bar{x}}{w}, \ldots, \frac{x_n - \bar{x}}{w}) \cdot (\frac{y_1 - \bar{y}}{w}, \ldots, \frac{y_n - \bar{y}}{w})$$

So we can see obviously that the task of computing b is also reduced to a secure two-party scalar product computation using the protocol 4.2.

5. Complexity Analysis

We assume that d is the number of bits needed to represent any number in the inputs, the computation complexity is O(n*d). protocol 4.1. For the protocols 4.3, the complexity is relative to the communication complexity in protocol 4.2. The overhead is k'/μ, where k' is just the size of each ciphertext in bits and μ is set to be $\lfloor \sqrt{m/N} \rfloor$ where N denotes number each party's input vectors and m denotes the mod in Z_m. We can see the protocol is already communication-efficient for practical applications.

6. Conclusions and Future Works

In this paper, we have formally defined secure two/multi-party statistical analysis problems corresponding to the cooperation models, and have developed some protocols for those problems. We proposed a framework for the mean value, correlation coefficient and linear regression line. We show our protocols are practical. Our future direction would be to study more complicated statistical analysis computations, such as nonlinear regression, variance analysis and so on. Furthermore, we also want to extent our protocols to multi-party environment study, various data analysis computations such as data mining other than the statistical analysis.

References

[1] A.C Yao. *Protocols for Secure Computation*, In 23rd FOCS, 1982.

[2] D. Chanm, C. Crepeau, and I. Damgard. *Multiparty unconditionally secure protocols.* In 20th ACM Symposium on Theory of Computing, pages 11–19. ACM Press, 1988.

[3] O. Goldreich, S. Micali, and A. Wigderson. *How to play any mental game: A completeness theorem for protocols with honest majority.* In 19th ACM Symposium on Theory of Computing, ACM Press, 1997.

[4] J. Benaloh. *Dense probabilistic encryption.* In Procee-dings of the Workshop on Selected Areas of Cryptography, 1994.

[5] P. Paillier. *Public-key cryptosystems based on com-posite degree residue classes.* EUROCRYPT 99, 1999.

[6] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikainen. *On Secure Product Computation for Privacy-preserving Data Mining*, In 7th Annual International Conf. in Information Security and Cryptology, 2004.