

Private Collaborative Forecasting and Benchmarking

**Mikhail Atallah, Marina Bykova, Jiangtao Li,
Keith Frikken, and Mercan Topkara**

Department of Computer Sciences and CERIAS
Purdue University

ACM Workshop on Privacy in the Electronic
Society (WPES'04)
October 2004

Outline

- Motivation
- Background
 - Time-series techniques
 - Regression-based techniques
- Building Blocks
 - Secure split protocol
 - Secure division protocols
- The protocols

Motivation

- What is “private collaborative forecasting and benchmarking” ?
 - Example 1: Forecasting
 - small retailers with similar products cannot compete with giant stores in their forecasting capabilities
 - they decide to collaborate to better estimate future demand
 - such collaboration requires sharing of proprietary data
 - unacceptable unless secure computation is used
 - Example 2: Forecasting
 - an overseas manufacturer cannot accurately forecast demand for seasonal merchandise

Motivation

- Example 3: Benchmarking
 - hospitals in an area want to investigate correlation between mortality rate of certain surgeries and patients health conditions
 - each hospital doesn't have enough data to draw reliable conclusions
 - each hospital would like to know how it compares to others
 - sharing of patient data may result in law suits or loss of face

Introduction

- We provide *secure* and *efficient* protocols for performing
 - forecasting based on time-series techniques
 - regression-based benchmarking
- Types of malicious behavior we address
 - semi-honest users
 - colluding users
- The computation requires division
 - introduces floating-point arithmetic
 - must be done in privacy-preserving manner

Background

- Time-series techniques
 - Observations are taken at regular intervals, t is current interval, d_i is data for interval i .
 - *Moving average*: $F_t = \left(\sum_{i=0}^{n-1} d_{t-i} \right) / n$
 - n is number of intervals
 - F_t is forecasted value for interval $t + 1$
 - *Weighted moving average*: $F_t = \sum_{i=0}^{n-1} w_i d_{t-i}$
 - $\vec{w} = \{w_0, w_1, \dots, w_{n-1}\}$ is a weight vector such that $\sum_{i=0}^{n-1} w_i = 1$

Background

- Time-series techniques (cont.)
 - *Exponential smoothing*: $F_t = F_{t-1} + \alpha(d_{t-1} - F_{t-1})$
 - F_i is value forecasted in interval i
 - α is a smoothing constant
- Regression techniques
 - *Linear regression*
 - given n points (x_i, y_i) , find a and b such that the sum of deviations of all points from $y = ax + b$ is minimized

$$a = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}, \quad b = \frac{\sum y - a \sum x}{n}$$

Building Blocks

- Are essential part of this work
- Provide different degree of resilience to collusion
- Vary in their level of difficulty and computation
- Consist of
 - Secure split protocol
 - each player P_j holds private input $x^{(j)}$
 - at the end of protocol player P_j obtains $z^{(j)}$ such that
$$\sum_{j=1}^m x^{(j)} = \sum_{j=1}^m z^{(j)}$$
 - Secure division protocols

Secure Division Protocols

- Description
 - each player P_j has private inputs $x^{(j)}$ and $y^{(j)}$
 - at the end of protocol player P_j learns $\frac{x}{y}$ where $x = \sum_{j=1}^m x^{(j)}$ and $y = \sum_{j=1}^m y^{(j)}$
- Division protocol 1
 - one player, say P_k , is selected to perform division
 - other players randomize the data, P_k divides it, and others recover the result
 - fast approach but works only when $m \geq 3$
 - directly operates on floating point numbers
 - doesn't leak as long as P_k doesn't collude with others

Secure Division Protocols

- Division protocol 2 (2-party)
 - is adopted from Newton's method to compute reciprocal of y : $2^{2^\ell-1}/y$ (ℓ is the length of y in bits)
 - uses secure 2-party multiplication and division by a constant protocols
 - secure but expensive
- Division protocol 3 (2-party)
 - uses homomorphic encryption
 - player P_1 hides the data, player P_2 performs the computation, P_1 recovers the result
 - should not be used when $x^{(1)} + x^{(2)}$ can be 0

Secure Division Protocols

- Division protocol 4
 - uses homomorphic encryption
 - each player selects a public-private key pair
 - each player participates in data hiding
 - advantages: resilient against collusion
 - disadvantages: doesn't scale well to large m 's

The Protocols

- Having these building blocks, development of full forecasting and benchmarking protocols is not hard
- Example: Moving average forecasting
 - Input: each player P_j has input data $d_{t-i}^{(j)}$ for n time intervals
 - Output: player P_j learns $\frac{F_t - d_t}{d_t}$, F_t is the forecasted moving average
 - The players need to find

$$\frac{F_t - d_t}{d_t} = \frac{d_{t-n+1} + \dots + d_{t-1} - (n-1)d_t}{nd_t}$$

The Protocols

- Example: Moving average forecasting

- Protocol steps:

1. Each player locally computes

$$x^{(j)} = d_{t-n+1}^{(j)} + \dots + d_{t-1}^{(j)} - (n-1)d_t^{(j)} \text{ and } y^{(j)} = nd_t^{(j)}$$

2. All players run secure division protocol on $x^{(j)}$'s and $y^{(j)}$'s and obtain $\frac{x}{y} = \frac{F_t - d_t}{d_t}$

Summary of Protocols

Protocol	Comm Rounds	Total Comm	Total Computation
Split	$O(1)$	$O(km)$	$O(km)$
Division with an Appointee	$O(1)$	$O(km)$	$O(km)$
2-party Division with Scaling	$O(\log \ell)$	$O(\log \ell)$	$O(\log \ell)$ enc's
2-party 2-key Division	$O(1)$	$O(1)$	$O(1)$ enc's
m-key Division	$O(1)$	$O(m^2)$	$O(m^2)$ enc's
Moving Average	same as div.	same as div.	same as div.
Exponential Smoothing	same as div.	same as div.	same as div.
Linear Regression	split + n div's	split + n div's	split + n div's

m - number of players, k - collusion threshold s.t. $1 \leq k < m$,
 ℓ - length of numbers, n - number of data points in regression.