Privacy-preservation for Stochastic Gradient Descent Method

Graduate School of System and Information Engineering
1st year of master course 201220724 Shuang Wu
Supervisor Jun Sakuma

2013.1.10

1 INTRODUCTION

The traditional paradigm in machine learning has been that given a data set, the goal is to learn a target function or decision model (such as a classifier) from it. Many techniques in data mining and machine learning follow a gradient descent paradigm in the iterative process of discovering this target function or decision model. For instance, Linear regression can be resolved through a gradient descent method that iteratively minimizes the error of the target function. Other gradient-descent-based methods include Bayesian networks induction, genetic algorithms, and simulated annealing. These traditional algorithms assume free access to data, either at a centralized location or in federated form. Increasingly, privacy and security concerns restrict this access.

Nowadays, data is often distributed among multiple parties: financial data is distributed across multiple banks and credit agencies; medical records are distributed across multiple hospitals and health care institutions and so on. secure multiparty computation and privacy preservation have attracted much attention in incorporating security into machine learning and data mining algorithms. A key issue in multiparty secure methods is to allow individual parties to preserve the privacy of its data, while contributing to the computation of a global result together with other parties. Many methods have been proposed to perform secure multiparty computation. For instance, the scalar product is a basic operations required in data mining. Recently, Li considered a privacy-preserving method [7] for Batch Gradient Descent (BGD) where each party learning a local classifier on its own data, and then gave two options on how to use the computed classifiers: all parties jointly compute the final classifier within a privacy-preserving way; another is using their local classifier to jointly perform on-demand prediction whenever an unknown sample's target value is required. The solution [7] proposed by Li is attractive, however, Batch Gradient Descent (BGD) method is costly for large scale data set problem and some randomized parameters in their method have not been declared in details. Another work [1] is a differentially private Stochastic Gradient Descent (SGD) algorithm for multiparty setting proposed by Rajkumar.

Here, we propose a privacy-preserving solution based on Stochastic Gradient Descent (SGD) procedure by using the paillier cryptosystem. Our proposal achieves more strict privacy than [7], in our protocol, each party knows nothing but their own data during the update process. And it is a different solution by using cryptography to ensure privacy rather than random disruption in [1].

2 RESEARCH TARGET

2.1 Overview

Classification comprises of two subtasks: learning a classifier from data with class labels often called a training data and predicting the class labels for unlabeled data using the learned classifier. As an example, consider about a classification task of detecting diseases. The training data comprises of medical records with labels " diseased " or " non-diseased ". More precisely, the training data is constructed by converting the classified medical records to word count vectors and then the classifier is learned from this training data. The learned classifier is used to predict which of the unlabeled medical records are deseased or non-diseased. As nowadays, more and more people pursue healthy life style, this kind of diseases prediction work is worth doing.

In considering about realizing this prediction work, there will be three parties involving in: patients (training data), hospital (learn a classifier) and individuals (predicting data) who need prediction. As patients come to hospital individually and unintentionally, it is appropriate for us to consider the data appear randomly and temporarily, which is exactly the idea of stochastic gradient descent. To the best of our knowledge, personal information is sensitive; and privacy concerns may prevent the parties from directly sharing

it. Hence, this prediction work should be conducted within a privacy-preserving way. Therefore, the problem can be seen as enable privacy preservation in gradient descent methods.

2.2 Decision model

Let us first consider a simple supervised learning setup. Each example is a pair $(\mathbf{x}, y)$. We consider a loss function $l(\hat{y}, y)$ that measures the cost of predicting $\hat{y}$ when the actual answer is y, and we consider the target function f parameterized by a weight vector $\mathbf{w}$ as $f(\mathbf{x}) = \varphi(\mathbf{w} \cdot \mathbf{x})$ which is differentiable function. The *empirical risk* which measures the training set performance is:

$$E_n(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(\mathbf{x}_i), y_i)$$

Batch Gradient Descent (BGD) method is often used to minimize this empirical risk, in which each iteration updates the weights $\mathbf{w}$ on the basis of the gradient of $E_n(f)$:

$$w_{t+1} = w_t - \eta_t \frac{1}{n} \nabla_w \sum_{i=1}^{n} l(f(\mathbf{x}_i), y_i)$$

As for Stochastic Gradient Descent (SGD) [4] , instead of computing the gradient of $E_n(f)$ exactly, each iteration estimates this gradient on the basis of a single randomly picked example $(\mathbf{x}_t, y_t)$:

$$w_{t+1} = w_t - \eta_t \nabla_w l(f(\mathbf{x}_i), y_i)$$

Table 1. illustrates Stochastic Gradient Descent (SGD) algorithms for a number of classic machine learning schemes.

Loss	Gradient
Perceptron	
$L = \max\{0, -y\mathbf{w}^T\varphi(\mathbf{x})\}$	$\begin{cases} -y\varphi(\mathbf{x}) & y(\mathbf{w}^T\varphi(\mathbf{x}) + b) \leqslant 0 \\ 0 & \text{otherwise} \end{cases}$
L-SVM	
$L = \max\{0, 1 - y\mathbf{w}^T\varphi(\mathbf{x})\}$	$\begin{cases} 0 & y(\mathbf{w}^T\varphi(\mathbf{x}) + b) \geqslant 1 \\ -y\varphi(\mathbf{x}) & \text{otherwise} \end{cases}$
Adaline	
$L = \frac{1}{2}(y - \mathbf{w}^T\varphi(\mathbf{x}))^2$	$(y - \mathbf{w}^T\varphi(\mathbf{x}))\varphi(\mathbf{x})$

Table 1. SGD algorithms for various learning system

3 TECHNICAL ELEMENT

3.1 Linear Support Vector Machine (L-SVM)

A standard L-SVM [2] takes the form:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

Given some training data D, a set of n points of the form:

$$D = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in R^m, y_i \in \{-1, 1\}\}_{i=1}^{N}$$

where the y_i is either 1 or -1, indicating if a patient is diseased or non-diseased. The optimal weight vector $\mathbf{w}$ and bias b are obtained by maximising the soft margin, which penalises each sample by the hinge loss:

$$argmin_{\mathbf{w},b} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\Sigma_{i=1}^{N} \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))$$

As for stochastic gradient descent, the gradient has the form:

$$\nabla_{\mathbf{w}} = \lambda\mathbf{w} + \begin{cases} 0 & y(\mathbf{w}^T\mathbf{x} + b) \geqslant 1 \\ -y\mathbf{x} & \text{otherwise} \end{cases}$$

The update process performs as the following:

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta_t \nabla_{\mathbf{w}}$$

3.2 PAILLIER CRYPTOSYSTEM

The Paillier Cryptosystem [5] is a modular, public key encryption scheme, created by Pascal Paillier, that can be used to conceal information, with its homomorphic properties.

- Homomorphic addition of plaintexts
  - $E_{pk}(m_1, r_1)E_{pk}(m_2, r_2) \bmod n^2$ $= E_{pk}(m_1 + m_2) \bmod n$
- Homomorphic multiplication of plaintexts
  - $E_{pk}(m_1, r)^{m_2} \bmod n^2 = E_{pk}(m_1 m_2) \bmod n$
  - $E_{pk}(m_1, r)^k \bmod n^2 = E_{pk}(km_1) \bmod n$

4 OUR PROPOSAL

In this section, we propose a protocol to perform privacy preserving stochastic gradient descent based on the Hospital-patient prediciton work (Figure 1). The following three conditions should be satisfied:

- Patients randomly appear in hospital.
- Patients are not able to know any thing during the update process except their own personal information.
- Hospital knows the final classifier "W_{final}" without seeing any updates during the training process. Thus, it is not able to infer any patients' personal information.

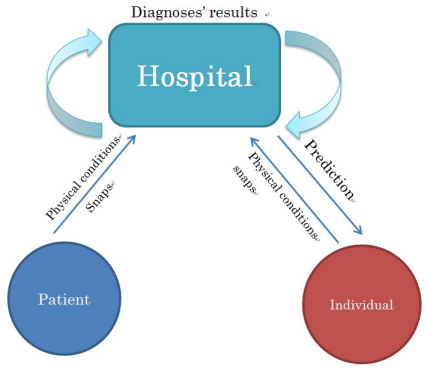

Figure 1.

4.1 Problem Statement

Given a data set M which includes N vectors: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, and target $\mathbf{y}$ as shown below:

$$M: \begin{array}{cccc} \mathbf{x}_1 = (x_{1,1} & x_{1,2} & \cdots & x_{1,m}) \\ \mathbf{x}_2 = (x_{2,1} & x_{2,2} & \cdots & x_{2,m}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_N = (x_{n,1} & x_{n,2} & \cdots & x_{n,m}) \end{array}$$

$$\mathbf{y}: [y_1, y_2, \cdots, y_n]^T$$

such that each vector $\mathbf{x}_i$ is associated with element y_i of $\mathbf{y}$. We say that y_i is $\mathbf{x}_i^{'}s$ observed target value.

Here the $(1 \times m)$ vector $\mathbf{x}_i$ holds the information of each patient and will be used for building the decision model. As mentioned above, hospital are able to acess part of patients' information (diagnosis records) which, can be viewed as each $\mathbf{x}_i$ is divided into two parts: $(1 \times m_1)$ vector $\mathbf{x}_i^p = (x_1^i, x_2^i, \ldots, x_{m_1}^i)$ and $(1 \times m_2)$ vector $\mathbf{x}_i^h = (x_{m_1+1}^i, x_{m_1+2}^i, \ldots, x_{m_1+m_2}^i)$ seperately known to patient P_i and Hospital. y_i is $\mathbf{x}_i$'s target value which is known to both patient P_i and Hospital.

4.2 Privacy-preserving Protocol for L-SVM

We propose the following protocol to perform secure stochastic gradient descent. The protocol executes as follows, and we assume that public key (pk) is public available, private key (sk) is known only by the generator.

Input. Patients randomly appear and each patient P_i holds $(1 \times m_1)$ vector $\mathbf{x}_i^p$, Hospital holds $(1 \times m_2)$ vector $\mathbf{x}_i^h$ corresponding to the patients. Both $\mathbf{x}_i^p$ and $\mathbf{x}_i^h$ are the same as that defined above. The target vector $\mathbf{y}$ is known to both Patients and Hospital.

Output. Classifier $\mathbf{w}_{final} = (w_1, w_2, \ldots, w_{m_1}, w_{m_1+1}, w_{m_1+1}, \ldots, w_{m_1+m_2})$

1 Patient P_1 and Hospital randomly generate $\mathbf{w}^h(1 \times m_2)$ and $\mathbf{w}^p(1 \times m_1)$ respectively. P_1 encrypts $E_p(w_i^h)$, Hospital encrypts $E_h(w_i^p)$ and sends to each other.

2 Patient P_1 does the encryption: $a_i = E_h(w_i^p)^{x_i^p} = E_h(w_i^p x_i^p)$, and so does hospital: $b_i = E_p(w_i^h)^{x_i^h} = E_p(w_i^h x_i^h)$. Then Patient P_1 does the multiplication of a_i: $A = \Pi_{1=1}^{m_1} a_i = E_h(\mathbf{w}^p \cdot \mathbf{x}^p)$, and so does Hospital :$B = \Pi_{1=1}^{m_2} b_i = E_p(\mathbf{w}^h \cdot \mathbf{x}^h)$

3 Patient P_1 and Hospital separately generates random number r^p and r^h.Then Patient P_1 computes: $A' = AE_h(r^p) = E_h(r^p + \mathbf{w}^p \cdot \mathbf{x}^p)$ and $A'' = E_p(-r^p)$, so does Hospital: $B' = BE_p(r^h) = E_p(r^h + \mathbf{w}^h \cdot \mathbf{x}^h)$ and $B'' = E_h(-r^h)$. After that, they send their values to each other.

4 Patient P_1 decrypts B' and re-encryts it by using the Hospital's public key in order to get the cyphertext $E_h(\mathbf{w}^h \cdot \mathbf{x}^h) = B''E_h(r^h + \mathbf{w}^h \cdot \mathbf{x}^h)$. Similar as Hospital: decrypts A' and gets $E_p(\mathbf{w}^p \cdot \mathbf{x}^p) = A''E_p(r^p + \mathbf{w}^p \cdot \mathbf{x}^p)$.

6 Patient P_1 computes $AE_h(\mathbf{w}^h \cdot \mathbf{x}^h) = E_h(\mathbf{w}^h \cdot \mathbf{x}^h + \mathbf{w}^p \cdot \mathbf{x}^p) = E_h(\mathbf{w} \cdot \mathbf{x})$. And Hospital computes $BE_p(\mathbf{w}^p \cdot \mathbf{x}^p) = E_p(\mathbf{w}^p \cdot \mathbf{x}^p + \mathbf{w}^h \cdot \mathbf{x}^h) = E_p(\mathbf{w} \cdot \mathbf{x})$.

7 Both of Patient P_1 and Hospital conduct Comparison Protocol [3] individually. For P_1: *COMPARE*$(E_h(y(\mathbf{w} \cdot \mathbf{x})), E_h(1))$; for Hospital: *COMPARE*$(E_p(y(\mathbf{w} \cdot \mathbf{x})), E_p(1))$. If the output of Comparison Protocol is "TRUE", They will update their $\mathbf{w}^p$ and $\mathbf{w}^h$ individually.
The update process is as following: For Patient $E_h(\mathbf{w}^{p'}) = E_h(\mathbf{w}^p)E_h(\mathbf{x}^p)^{y\eta} = E_h(\mathbf{w}^p + \eta y \mathbf{x}^p)$, and for Hospital: $E_p(\mathbf{w}^{h'}) = E_p(\mathbf{w}^h)E_p(\mathbf{x}^h)^{y\eta} = E_p(\mathbf{w}^h + \eta y \mathbf{x}^h)$

After finishing this update, Hospital will generate a random vector $\mathbf{R}^h = (r_1^h, r_2^h, \ldots, r_{m2}^h)$ in order to creates $c_i = E_p(w_i^{h'})E_p(r_i^h) = E_p(w_i^{h'} + r_i^h)$ and sends to Patient P_1. Patient P_1 does:(i) Decrypts c_i and uses the next patient's public key to re-encrypt $E_{p_{next}}(w_i^{h'} + r_i^h)$; (ii) Encrypts $E_{p_{next}}(-r_i^p)$; (iii) Generates a random vector $\mathbf{R}^p = (r_1^p, r_2^p, \ldots, r_{m1}^p)$ in order to encrypts $E_h(w_i^{p'})E_h(r_i^p) = E_h(w_i^{p'} + r_i^p)$. Then sends all these three items to Hospital.

When the next Patient P_2 comes, Hospital will send $E_h(w_i^{p'} + r_i^p)$ and $E_{p_{next}}(-r_i^p)$ to him. P_2 is able to use these information to compute $E_h(w_i^{p'} + r_i^p)E_h(-r_i^p) = E_h(w^{p'})$. From now on, the step 2 to 7 will be repeated.

After the protocol terminating, the last Patient directly sends $E_h(w_{final}^p)$ to Hospital without disturbing. Thus, Hospital may combine $\mathbf{w}_{final}^p$ and $\mathbf{w}_{final}^h$ into $\mathbf{w}_{final}$ which can be used for prediction.

4.3 Discussion

As mentioned in the protocol: after updating and before the next patient comes, the former patient should use the next patient's public key to re-encryt the partial classifier. Here discussing the two possible methods to deal with patients' public keys:
Method 1. After updating, the former patient uses all the pks to encrypt his partial classifier and sends to hospital. When the next patient comes, hospital is able to choose the ciphertext corresponding to this patient and abandon all the other useless ciphertexts. This method achieves strict privacy, however, it is costly in deal with large amount of data.
Method 2. In considering of hospital and patients, hospital, to some extent, has the information about who will come next. Thus, hospital could help the former patient judge which pk should be used. This method is appropriate in the real situation, and more flexible than the first one.

Not only for linear support vector machine, this protocol can be used for those problems in which the target function is linearly separable or has the form of $\mathbf{w}^T\phi(\mathbf{x})$.

5 EXPERIEMENT

5.1 The Pegasos Algorithm

The Pegasos Algorithm [6] is recently proposed by Joachims for linear SVMs. This experiment implement the original Pegasos and then try to implement it using the protocol form.

5.2 Dataset

Dataset SPECT, which is divided into SPECT.train and SPECT.test, is used in this experiment. This data set describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal.

	SPECT.train	SPECT.test
number of instances	80	187
number of attributes	23	23

5.3 Result

	Prediction Accuracy
Original Pegasos Algorithm	87.17
Pegasos of Protocol form	87.17

6 FUTURE WORK

- Implement this privacy-preserving protocol.
- Consider about the privacy-preservation protocol for non-linear problem.

References

[1] Rajkumar A. Shivani A. A differentially private stochastic gradient descent algorithm for multiparty classification. *Journal of Machine Learning Research - Proceedings Track*, 22:933–941, 2012.

[2] Bottou L. Lin C.-J. Support vector machine solvers. *Large scale kernel machines. MIT Press*, 2007.

[3] Philippe G. A private stable matching algorithm. In *Financial Cryptography*, pages 65–80, 2006.

[4] Bottou L. Large-scale machine learning with stochastic gradient descent. *In Lechevallier, Y. and Saporta, G., editors, Compstat 2010,*, 2010.

[5] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.

[6] Shai S. S. Yoram S. Nathan S. Pegasos: Primal estimated sub-gradient solver for svm. *ICML'07 Proceedings of the 24th International Conference on Machine Learning*, 2007.

[7] Wan L. Han S. G. Ng W. K. Lee V. C. S. Privacy-preservation for gradient descent methods. *Kdd-2007 Proceedings of the Thirteenth Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 2007.