Quantum Privacy Preserving Perceptron

Shenggang Ying

Centre for Quantum Software and Information, University of Technology Sydney
Joint work with Mingsheng Ying and Yuan Feng
arXiv:1707.09893

27/09/2017
1. Motivations

2. Problem and Classical Method

3. Our Quantum Protocol

4. Correctness

5. Privacy Analysis

6. Conclusion
Motivations

Develop a quantum algorithm/protocol, which
- is easy to implement,
- deals with data mining or machine learning problems,
- and is better than classical algorithms.
Motivations

Problem and Classical Method

Our Quantum Protocol

Correctness

Privacy Analysis

Conclusion
**Problem**

**Table:** Savings of 8 people. (Training examples.)

<table>
<thead>
<tr>
<th>ID ( (i) )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salary &gt; 50k ( (x_1) )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Number of Children ( (x_2) )</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Saving &gt; 100k ( (y) )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Problem: Machine learning**

Find a function \( f : (x_1, x_2) \mapsto y \).
Perceptron
Definition [Fr57, No62]


Definition
A perceptron is a linear classifier

\[
f(\vec{x}) = \begin{cases} 
1 & \vec{w} \cdot \vec{x} + b > 0, \\
0 & \text{otherwise.}
\end{cases}
\]
Perceptron
Algorithm [Fr57, No62]

- Initialize $\vec{w} = 0$ and $b = 0$.
- Repeat the following loop, until no update happens.
  - For all $i$, do the following two steps.
    - **Check.** Compute $d_i = f(\vec{x}_i)$.
    - **Update.** If $d_i \neq y_i$, update
      $$\vec{w} \leftarrow \vec{w} + (y_i - d_i)\vec{x}_i \quad \text{and} \quad b \leftarrow b + y_i - d_i.$$  
- Output $\vec{w}$ and $b$.

**Correctness**

**Perceptron Example**

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>x*w' + b</th>
<th>Update</th>
<th>w</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>(0, 2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(0, 0)</td>
<td>0</td>
</tr>
<tr>
<td>02</td>
<td>(1, 1)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(1, 1)</td>
<td>1</td>
</tr>
<tr>
<td>03</td>
<td>(0, 1)</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>(1, 1)</td>
<td>1</td>
</tr>
<tr>
<td>04</td>
<td>(1, 3)</td>
<td>0</td>
<td>5</td>
<td>-1</td>
<td>(0, -2)</td>
<td>0</td>
</tr>
<tr>
<td>05</td>
<td>(1, 3)</td>
<td>0</td>
<td>-6</td>
<td>0</td>
<td>(0, -2)</td>
<td>0</td>
</tr>
<tr>
<td>06</td>
<td>(1, 0)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(1, -2)</td>
<td>1</td>
</tr>
<tr>
<td>07</td>
<td>(0, 1)</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
<tr>
<td>08</td>
<td>(1, 0)</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>x*w' + b</th>
<th>Update</th>
<th>w</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>(0, 2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>(1, 1)</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>(0, 1)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>(1, 3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>(1, 3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>(1, 0)</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>(0, 1)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>(1, 0)</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>(1, -1)</td>
<td>2</td>
</tr>
</tbody>
</table>
Classical Method

Adding Noise and Reconstructing Distribution [AS00]

Suppose Alice holds the original training set $D = \langle (\vec{x}_i, y_i) \rangle$. Bob wants to compute $f$ on $D$, and Alice wants to preserve $D$.

1. Alice adds noise to $D$ to get $D' = \langle (\vec{x}'_i, y_i) \rangle$, where $\vec{x}'_i = \vec{x}_i + \vec{r}_i$ and $\vec{r}_i$ is a random vector with distribution $p(\vec{r})$.

2. Alice publishes $D'$ and $p(\vec{r})$.

3. Bob reconstructs the training set $\tilde{D}$ based on $D'$ and $p(\vec{r})$, and then computes $f$ on $\tilde{D}$.

$$g_X^{(j+1)}(a) = \frac{1}{N} \sum_{i=1}^{N} p(x'_i - a) g^{(j)}(a) / \sum_{z} p(x'_j - z) g^{(j)}(z)$$

Suppose a training vector $\vec{z} = (z_1, z_2, \cdots, z_k)$ has $k$ attributes.

- Reconstruct the distribution $g_i(z_i)$ of each attribute one by one, and then get

$$g(\vec{z}) = \prod g_i(z_i).$$

- Weakness: the accuracy is very low if the attributes are not independent of each other.
Classical Method
Reconstruction [AS00]

Suppose a training vector $\vec{z} = (z_1, z_2, \cdots, z_k)$ has $k$ attributes.

- Reconstruct the distribution $g_i(z_i)$ of each attribute one by one, and then get
  
  $$g(\vec{z}) = \prod g_i(z_i).$$

  - Weakness: the accuracy is very low if the attributes are not independent of each other.

- Reconstruct the distribution $g(\vec{z}) = g(z_1, z_2, \cdots, z_k)$ of all attributes directly.
  - Weakness: the computational cost is exponential on the number $k$ of attributes.
Motivations

Problem and Classical Method

Our Quantum Protocol

Correctness

Privacy Analysis

Conclusion
Our Quantum Protocol

Basic Idea

Quantum Check: Employing a quantum data system to compute \( f(\vec{x}_i) \).
This data system works by randomly sending a computational state or a test state to detect Bob’s cheat.

Classical Update: Adding noise into original training examples.
The distribution \( p(\vec{r}) \) of noise is kept secret.
Our Quantum Protocol

Quantum Check

To compute $f(\vec{x}_i)$, Alice sends 3 states to Bob, and asks Bob to send back these states.

- Random sequence of states. (c,t,t), (t,c,t), or (t,t,c).
- Computational State.

$$|+\rangle|\vec{x}_i\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{f(\vec{x}_i)}|1\rangle)|\vec{x}_i\rangle.$$

- Test State. Suppose $\vec{z} = 1z_2\,z_3\cdots\,z_n$. Totally $n$ bits.

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)|z_2\,z_3\cdots\,z_n\rangle \rightarrow \frac{1}{\sqrt{2}}(|00\rangle + (-1)^{f(\vec{z})}|11\rangle)|z_2\,z_3\cdots\,z_n\rangle.$$

Randomly choosen from $n2^{n+1}$ test states.
Our Quantum Protocol

Classical Update

- Classical update in our quantum protocol.

\[ \vec{w} \leftarrow \vec{w} + (y_i - d_i)\vec{x}' \quad \text{and} \quad b \leftarrow b + y_i - d_i, \]

where \( \vec{x}' = \vec{x}_i + \vec{r} \).

- Bob does not know \( \vec{x}_i \) and \( \vec{r} \).
- Bob does not know the distribution \( p(\vec{r}) \) of noise.
- Bob only knows \( \vec{x}' \) and the expected value of noise is 0.
Our Quantum Protocol

Classical Update

- Classical update in our quantum protocol.
  \[
  \vec{w} \leftarrow \vec{w} + (y_i - d_i)\vec{x}' \quad \text{and} \quad b \leftarrow b + y_i - d_i,
  \]
  where \( \vec{x}' = \vec{x}_i + \vec{r} \).
  - Bob does not know \( \vec{x}_i \) and \( \vec{r} \).
  - Bob does not know the distribution \( p(\vec{r}) \) of noise.
  - Bob only knows \( \vec{x}' \) and the expected value of noise is 0.

- Classical reconstruction method.
  \[
  \vec{w} \leftarrow \vec{w} + (y_i - d_i)\vec{x}_i \quad \text{and} \quad b \leftarrow b + y_i - d_i.
  \]
  - Bob does not know \( \vec{x}_i \) and \( \vec{r} \).
  - Bob knows the distribution \( p(\vec{r}) \) of noise.
### Our Quantum Protocol

#### Example

<table>
<thead>
<tr>
<th>014:</th>
<th>x=(1, 3)</th>
<th>y=0</th>
<th>x*w'+b=0</th>
<th>Update=0</th>
<th>x'= (2, 2)</th>
<th>w= (0, 0)</th>
<th>b= 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>023:</td>
<td>x=(0, 1)</td>
<td>y=1</td>
<td>x*w'+b=0</td>
<td>Update=1</td>
<td>x'= (-1, 0)</td>
<td>w= (-1, 0)</td>
<td>b= 1</td>
</tr>
<tr>
<td>032:</td>
<td>x=(1, 1)</td>
<td>y=1</td>
<td>x*w'+b=0</td>
<td>Update=1</td>
<td>x'= (0, 0)</td>
<td>w= (-1, 0)</td>
<td>b= 2</td>
</tr>
<tr>
<td>041:</td>
<td>x=(0, 2)</td>
<td>y=0</td>
<td>x*w'+b=2</td>
<td>Update=-1</td>
<td>x'= (0, 2)</td>
<td>w= (-1, -2)</td>
<td>b= 1</td>
</tr>
<tr>
<td>058:</td>
<td>x=(1, 0)</td>
<td>y=1</td>
<td>x*w'+b=0</td>
<td>Update=1</td>
<td>x'= (1, 0)</td>
<td>w= (0, -2)</td>
<td>b= 2</td>
</tr>
<tr>
<td>067:</td>
<td>x=(0, 1)</td>
<td>y=1</td>
<td>x*w'+b=0</td>
<td>Update=1</td>
<td>x'= (1, 0)</td>
<td>w= (1, -2)</td>
<td>b= 3</td>
</tr>
<tr>
<td>076:</td>
<td>x=(1, 0)</td>
<td>y=1</td>
<td>x*w'+b=4</td>
<td>Update=0</td>
<td>x'= (2, -1)</td>
<td>w= (1, -2)</td>
<td>b= 3</td>
</tr>
<tr>
<td>085:</td>
<td>x=(1, 3)</td>
<td>y=0</td>
<td>x*w'+b=-2</td>
<td>Update=0</td>
<td>x'= (2, 3)</td>
<td>w= (1, -2)</td>
<td>b= 3</td>
</tr>
</tbody>
</table>

| 115: | x=(1, 3) | y=0 | x*w'+b=-2| Update=0 | x'= (0, 2) | w= (1, -2) | b= 3 |
| 126: | x=(1, 0) | y=1 | x*w'+b=4 | Update=0 | x'= (2, 1) | w= (1, -2) | b= 3 |
| 137: | x=(0, 1) | y=1 | x*w'+b=1 | Update=0 | x'= (-1, 2) | w= (1, -2) | b= 3 |
| 148: | x=(1, 0) | y=1 | x*w'+b=4 | Update=0 | x'= (2, 1) | w= (1, -2) | b= 3 |
| 151: | x=(0, 2) | y=0 | x*w'+b=-1| Update=0 | x'= (-1, 1) | w= (1, -2) | b= 3 |
| 162: | x=(1, 1) | y=1 | x*w'+b=2 | Update=0 | x'= (0, 2) | w= (1, -2) | b= 3 |
| 173: | x=(0, 1) | y=1 | x*w'+b=1 | Update=0 | x'= (-1, 1) | w= (1, -2) | b= 3 |
| 184: | x=(1, 3) | y=0 | x*w'+b=-2| Update=0 | x'= (2, 3) | w= (1, -2) | b= 3 |
Correctness

Theorem 1

Suppose

1. there exists $\vec{w}^*$ and $b^*$ classifying training set $D$ correctly,
2. the expected value of noise in the protocol is 0.

Then the quantum protocol terminates and outputs a correct classifier with probability 1.

This theorem states that the correctness of our quantum protocol is independent of the specific form of the noise generator.
Correctness

- $R_0(\delta)$: Uniform distribution $[-\delta, \delta]$.
- $R_1(\delta)$: Uniform distribution $[-1.5\delta, -0.5\delta) \cup \{0\} \cup (0.5\delta, 1.5\delta]$.
- $R_2(\delta)$: $[-1.5\delta, -0.5\delta) \cup \{0\} \cup (0, 2\delta]$.
- $R_3(\delta)$: Normal distribution $(0, \delta)$
- $R_4(\delta)$: Half normal distribution (for negative part), and half uniform distribution (for positive part).
Motivations

Problem and Classical Method

Our Quantum Protocol

Correctness

Privacy Analysis

Conclusion
Privacy analysis for the check step.

Theorem 2 (Qualitative analysis)
If Bob’s operators are not equivalent to an identity operator or a controlled-Z gate, then it will be detected with nonzero probability.
Privacy analysis for the check step.

Theorem 2 (Qualitative analysis)

If Bob’s operators are not equivalent to an identity operator or a controlled-Z gate, then it will be detected with nonzero probability.

Quantitative analysis

Before being detected, Bob can expectedly perform measurements on $O(n)$ qubits with basis $\{|0\rangle, |1\rangle\}$.

Note $n$ is the length of bit string representing $\tilde{x}_i$, and is independent of the number $N$ of training examples.
Privacy analysis for the update step.

Methods for comparison
Privacy analysis for the update step.

Methods for comparison

- Quantum protocol with $R_0(\delta)$.
- Classical noise without reconstruction: Uniform distribution $[-\delta, \delta]$.
- Classical noise without reconstruction: Normal distribution $(0, 0.484\delta)$.
- Classical noise with reconstruction1D: Uniform distribution $[-\delta, \delta]$.
- Classical noise with reconstruction2D: Uniform distribution $[-\delta, \delta]$. 
Privacy analysis for the update step.

- Terminating probability: The algorithm terminates in 40000 outer loops.
- Success probability: the algorithm terminates in 40000 outer loops and outputs a correct classifier for the original training set $D$.

(i) $T$ probabilities on training set 1.

(j) $S$ probabilities on training set 1.
Privacy analysis for the update step.

Comparison

(k) T probabilities on training set 2.

(l) S probabilities on training set 2.

(m) T probabilities on training set 3.

(n) S probabilities on training set 3.
Motivations

Problem and Classical Method

Our Quantum Protocol

Correctness

Privacy Analysis

Conclusion
**Conclusion**

- **Easy to implement.**
  - No quantum database or Oracle.
  - Any quantum state only lasts for $O(n)$ two-qubit gates. It is independent of $N$.

- **Higher privacy level than classical methods.**

- **Generalizable.**
  - Other data mining and machine learning tasks.
  - Preserving both parties’ privacy.
Reference

Fr57 Frank Rosenblatt. The Perceptron—a perceiving and recognizing automaton. Report 85-460-1, Cornell Aeronautical Laboratory. (1957)


For other references, see arXiv:1707.09893.
Thank you!