Privacy Preserving Distributed ID3 Algorithm

Chen Yuechen
Fung Divine
Nan Meng

April 28, 2016

Overview

* Introduction
* Problem Definition
* Solution
* Result
* Conclusion

Privacy Preserving Data Mining

- Mining while protecting the privacy of data.

Figure : Lindell's definition

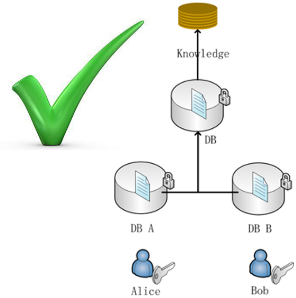
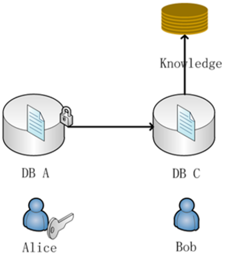


Figure : Agrawal's definition



ID3 Algorithm

- ID3 is an algorithm used to generate a decision tree from a dataset, and is typically used in the data mining.
  1. Calculate the **entropy** of every attribute using the data set S.
  2. Split the set S into subsets using the attribute for which entropy is minimum
  3. Make a decision tree node containing that attribute
  4. Recurse on subsets using remaining attributes.

Distributed ID3 Algorithm

Table : Play Golf Dataset

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes

$\Rightarrow$ **Alice**

$\Rightarrow$ **Bob**

Distributed ID3 Algorithm

- Data is distributed in two or more parties

Table : Alice

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Table : Bob

Outlook	Temp	Humidity	Windy	Play Golf
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes

- Combine data together and get a decision tree

Problem Definition

- However, data is **privacy**.

Table : Alice

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Table : Bob

Outlook	Temp	Humidity	Windy	Play Golf
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes

- How to share data in a safe way in distributed ID3 algorithm?

An Example of Distributed ID3 Algorithm

- Here we use a example of Distributed ID3 algorithm to clearly define the problem. For example, Compute the entropy of **Rainy**.

Table : Alice

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

3 records, 2 No, 1 Yes

Table : Bob

Outlook	Temp	Humidity	Windy	Play Golf
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes

2 records, 1 No, 1 Yes

$$Entropy(Rainy) = \underbrace{-\frac{2+1}{3+2}log_2(\frac{2+1}{3+2})}_{PlayGolf=No} - \underbrace{\frac{1+1}{3+2}log_2(\frac{1+1}{3+2})}_{PlayGolf=Yes}$$

$$= -\frac{3}{5}log_2(\frac{3}{5}) - \frac{2}{5}log_2(\frac{2}{5})$$

An Example of Distributed ID3 Algorithm

- For example, Compute the entropy of **Rainy**.

Table : Alice

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

3 records, 2 No, 1 Yes

Table : Bob

Outlook	Temp	Humidity	Windy	Play Golf
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes

2 records, 1 No, 1 Yes

$$Entropy(Rainy) = -\frac{2+1}{3+2}log_2(\frac{2+1}{3+2}) - \frac{1+1}{3+2}log_2(\frac{1+1}{3+2})$$

$$= -\frac{3}{5}log_2(\frac{3}{5}) - \frac{2}{5}log_2(\frac{2}{5})$$

An Example of Distributed ID3 Algorithm

- For example, Compute the entropy of **Rainy**.

Table : Alice

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Table : Bob

Outlook	Temp	Humidity	Windy	Play Golf
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes

3 records, 2 No, 1 Yes

2 records, 1 No, 1 Yes

$$-\frac{2+1}{3+2} log_2(\frac{2+1}{3+2})$$

An Example of Distributed ID3 Algorithm

- For example, Compute the entropy of **Rainy**.

Table : Alice

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Table : Bob

Outlook	Temp	Humidity	Windy	Play Golf
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes

3 records, 2 No, 1 Yes

2 records, 1 No, 1 Yes

$$-\frac{2+1}{3+2} log_2 \left(\frac{2+1}{3+2}\right)$$

An Example of Distributed ID3 Algorithm

- For example, Compute the entropy of **Rainy**.

Table : Alice

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

3 records, 2 No, 1 Yes

Table : Bob

Outlook	Temp	Humidity	Windy	Play Golf
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes

2 records, 1 No, 1 Yes

$$\frac{2+1}{3+2}$$

An Example of Distributed ID3 Algorithm

- For example, Compute the entropy of **Rainy**.

Table : Alice

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Table : Bob

Outlook	Temp	Humidity	Windy	Play Golf
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes

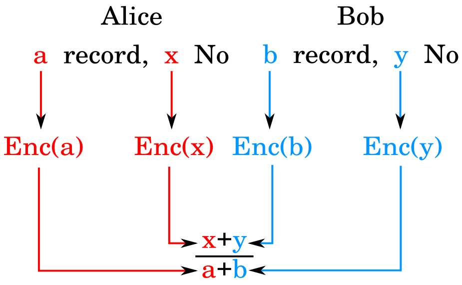
a records, x No, 1 Yes

b records, y No, 1 Yes

$$\frac{x+y}{a+b}$$

Problem Definition

- Compute $\frac{x+y}{a+b}$ without reveal a, x, b, y.
- Realize Privacy Preserving Distributed ID3 algorithm.



$Enc(\cdot)$ – Encryption Algorithm

Solution

PPWAP

- PPWAP: Privacy Preserving Weight Average Protocol

- In this project, we choose PPWAP by Pailier Encryption.

Pailier Encryption

- $KeyGeneration()$: Generate public key PK, and secret key SK.
- $Encryption(m, PK)$: Using PK to encrypt message m, output $Enc(m)$.
- $Decryption(Enc(m), SK)$: Using SK to decrypt $Enc(m)$, output m.

Pailier Encryption

- Property: Addition Homomorphism
- Given two messages $m1$ and $m2$, $Enc(m1 + m2) = Enc(m1) \cdot Enc(m2)$.
- The encryption of $m1 + m2$ can be computerd by $Enc(m1)$ and $Enc(m2)$.

PPWAP based on Pailier Encryption

Privacy Preserving Weighted Average Protocol

- Within the help of Paillier, build PPWAP scheme.

Alice　　　Bob

Enc(a)　Enc(x)　Enc(b)　Enc(y)

$$\frac{x+y}{a+b}$$

PPWAP

Alice		Bob
1. $KeyGeneration() : SK, PK$ $Encryption(a, PK) : Enc(a)$ $Encryption(x, PK) : Enc(x)$	$\Rightarrow$	$Enc(a)$ $Enc(x)$
2.		Random integer z $Enc(a)^z, Enc(x)^z$

PPWAP

Alice | Bob

1. $KeyGeneration() : SK, PK$
   $Encryption(a, PK) : Enc(a)$
   $Encryption(x, PK) : Enc(x)$

$\Rightarrow$ $Enc(a)$, $Enc(x)$

---

2. Random integer z
   $Enc(a)^z, Enc(x)^z$

$$\boxed{\mathrm{Enc(a)^z = Enc(a)...Enc(a) = Enc(a+a+...+a) = Enc(za)}}$$

PPWAP

Alice

Bob

1. $KeyGeneration() : SK, PK$
   $Encryption(a, PK) : Enc(a)$
   $Encryption(x, PK) : Enc(x)$

$\Longrightarrow$

$Enc(a)$
$Enc(x)$

2.

Random integer z
$Enc(a)^z, Enc(x)^z$
$Enc(za), Enc(zx)$

PPWAP

Alice

Bob

$$Encryption(b, PK) : Enc(b) \Rightarrow Enc(zb)$$
$$Encryption(y, PK) : Enc(y) \Rightarrow Enc(zy)$$
$$Enc(za + zb) = Enc(za)$$
$$Enc(zb)Enc(zx + zy) = Enc(zx)Enc(zy)$$

3. $Enc(za + zb)$

$Enc(zx + zy)$ $\Longleftarrow$

---

4. $Decryption(Enc(za + zb), SK) :$

$za + zb$ $\Longrightarrow$

$Decryption(Enc(zx + zy), SK) :$

$zx + zy$

$$\frac{zx+zy}{za+zb} = \frac{x+y}{a+b} \quad \Rightarrow \quad \frac{x+y}{a+b}$$

Algorithm

Algorithm 1 Two-party jointly decision tree algorithm

procedure PRIVACYID3($D, Attribute, transInfo, T$)
  $ct \leftarrow createNode()$
  $label(ct) = mostCommonClass(D, transInfo, T)$
  IF $\forall \langle \mathbf{x}, c(\mathbf{x}) \rangle \in D : c(\mathbf{x}) = c$ **THEN**
    $return(t)$
  ENDIF
  IF $Attributes = \varnothing$ **THEN**
    $return(t)$
  ENDIF
  $\widetilde{A} = \underset{A \in Attributes}{argmax} \; (InformationGain(D, A, transInfo))$
  for $a \in \widetilde{A}$ **do**
    $D_a = \left\{ (\mathbf{x}, c(\mathbf{x})) \in D : \mathbf{x} \mid_{\widetilde{A}} = a \right\}$
    IF $D = \varnothing$ **THEN**
      $ct' = createNode()$
      $label(ct') = mostCommonClass(D, transInfo, T)$
      $createEdge(ct, a, ct')$
    ELSE
    $transInfo^* = TransInfo(B, Attributes \backslash \{\widetilde{A}\}, T^*)$
    $createEdge(ct, a, PrivacyID3(D_a, Attributes \backslash \{\widetilde{A}\}$
      $, transInfo^*, T))$
    ENDIF
  return *combinedtree*

Figure : Two-party Jointly Decision Tree Algorithm.

Result

- Demo
- Efficiency: The runtime depend on 3 factors.
  - Dataset size
  - Length of Key in encryption algorithm
  - Number of parties

Algorithm Implement

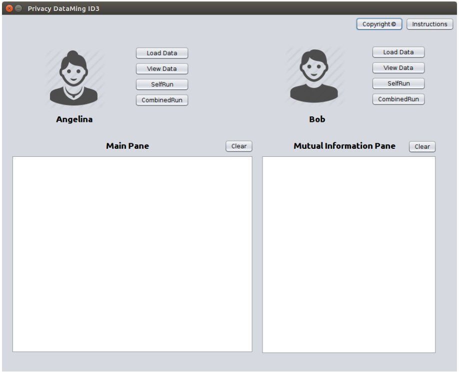


Figure : Welcome Graphical User Interface.

Algorithm Implement

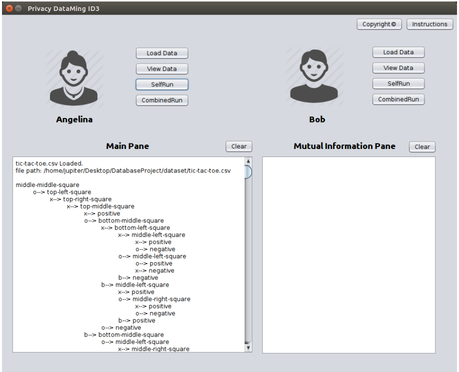


Figure : Result of single-party ID3 algorithm on tic-tac-toe2 dataset.

Conclusion

- The PPWAP scheme is purposed in 2005 in PP K-means.
  - PPWAP can be extend to multi-party, supports Multi-party distributed ID3 algorithm.
- Further research focus on improving the security level.
  - The scheme became safer and more complex.
- Current research focus on preventing malicious attack.

Conclusion

- Select two large primes, p and q.
- Calculate the product $n = p \times q$, such that $gcd(n, \Phi(n)) = 1$, where $\Phi(n)$ is $(p-1)(q-1)$.
- Choose a random number g, where g has order multiple of n or $gcd(L(g^{\lambda} mod\ n^2), n) = 1$, where $L(t) = (t-1)/n$ and $\lambda(n) = lcm(p-1, q-1)$.
- The public key is composed of (g, n), while the private key is composed of (p, q, λ).
- The Encryption of a message $m < n$ is given by:
  - $c = g^m \cdot r^n mod\ n^2$
- The Decryption of ciphertext c is given by: The Decryption of ciphertext c is given by:
  - $m = (L(g^{\lambda} mod\ n^2)/L(g^{\lambda} mod\ n^2)) mod\ n$

The End