Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee

Boxiang Dong  Ruilin Liu  Wendy Hui Wang

Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ

December 10, 2013
Data-mining-as-a-service (DMaS)

Data Mining as a Service:

- Weak client
- Computationally powerful service provider (e.g. cloud)
- Result integrity: are the returned mining results the same as if the computation were locally executed?

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Outsourcing Setting

• We focus on the problem of result integrity of outsourced frequent itemset mining.

• The architecture of outsourcing frequent itemset mining:

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Verification Goal

Given a transaction dataset $D$ and its correct frequent itemset mining result $F$, let $F^S$ be the erroneous mining result that the server returns.

- Integrity concerns:
  - **Completeness** no frequent itemset is missing in $F^S$.
  - **Correctness** all itemsets in $F^S$ are frequent.

- We propose an efficient approach to catch incorrect/incomplete mining result with 100% certainty.

----------------------------------------
Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Verification Framework

- The server constructs cryptographic proofs of the mining results.
  - We use the set intersection verification protocol [PTT11] to construct the proofs.
  - Use the proof to verify the true support of a frequent/infrequent itemset.

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Set Intersection Verification Protocol

Given a collection sets $S = \{S_1, \ldots, S_m\}$, an intersection result $Y = \{y_1, \ldots, y_\delta\}$, $Y = S_1 \cap S_2 \cap \cdots \cap S_m$ is the correct intersection of $S$ if and only if:

- $(Y \subseteq S_1) \land \cdots \land (Y \subseteq S_m)$ (subset condition);
- $(S_1 - Y) \cap \cdots \cap (S_m - Y) = \emptyset$ (completeness condition).
Set Intersection Verification Protocol

Given a collection sets \( S = \{S_1, \ldots, S_m\} \), an intersection result \( Y = \{y_1, \ldots, y_\delta\} \), \( Y = S_1 \cap S_2 \cap \cdots \cap S_m \) is the correct intersection of \( S \) if and only if:

- \((Y \subseteq S_1) \land \cdots \land (Y \subseteq S_m)\) (subset condition);
- \((S_1 - Y) \cap \cdots \cap (S_m - Y) = \emptyset\) (completeness condition).

<table>
<thead>
<tr>
<th>[PTT11] server prepares ( \Pi(Y) = {B, A, W, C} )</th>
<th>client checks</th>
</tr>
</thead>
<tbody>
<tr>
<td>coefficients ( B = {b_\delta, b_{\delta-1}, \ldots, b_0} ) of polynomial ((s + y_1)(s + y_2) \cdots (s + y_\delta))</td>
<td>( B = {b_0, \ldots, b_\delta} ) are correct.</td>
</tr>
<tr>
<td>accumulation values ( A = {\text{acc}(S_j) \mid \forall S_j \in S} ) where ( \text{acc}(S_j) = g \prod_{x \in S_j} (s + x) )</td>
<td>( A ) are correct</td>
</tr>
<tr>
<td>subset witness ( W = {W_j \mid \forall S_j \in S} ) where ( W_j = g^{P_j(s)} ), ( P_j(s) = \prod_{x \in S_j - Y} (x + s) )</td>
<td>( e(\prod_{k=0}^{\lvert Y \rvert} (g^s)^{b_k}, W_j) = e(\text{acc}(S_j), g) ) for ( j = 1, \ldots, m )</td>
</tr>
<tr>
<td>completeness witness ( C = {C_j \mid \forall S_j \in S} ) for each set ( S_j \in S ), ( C_j = g^{q_j(s)} ) s.t. ( q_1(s)P_1(s) + q_2(s)P_2(s) + \cdots + q_m(s)P_m(s) = 1 )</td>
<td>( \prod_{j=1}^{m} e(W_j, C_j) = e(g, g) )</td>
</tr>
</tbody>
</table>
Basic Solution

Given a dataset \( D \) that contains \( n \) unique items, the client does the following:

1. Build the item-based inverted index \( E \) that consists of \( n \) inverted lists \( \{ L_1, \ldots, L_n \} \).
2. Construct the Merkle hash tree \( T \) of the inverted index.

- Leaf \( l_j \) is assigned \( h_j = \text{hash}(\text{acc}(L_j)(s+j)) \).
- Internal node \( v \) with children \( c_1, \ldots, c_k \) is assigned \( h_v = \text{hash}(h_{c_1} || \ldots || h_{c_k}) \).

Mapping to the set intersection verification problem

Verifying whether any itemset \( I \) is included in a set of transactions \( T_I \) is equivalent to verifying whether \( T_I \) is the correct intersection of the inverted lists of all items in \( I \).
Basic Solution

Given a dataset $D$ that contains $n$ unique items, the client does the following:

1. **Build the item-based inverted index $E^I$** that consists of $n$ inverted lists $\{L_1, \ldots, L_n\}$.

2. **Construct the Merkle hash tree $\mathcal{T}$** of the inverted index.
   - Leaf $l_j$ is assigned $h_j = \text{hash}(\text{acc}(L_j)^{(s+j)})$.
   - Internal node $v$ with children $c_1, \ldots, c_k$ is assigned $h_v = \text{hash}(h_{c_1} \Vert \ldots \Vert h_{c_k})$.

---

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Basic Solution

Given a dataset $D$ that contains $n$ unique items, the client does the following:

1. **Build the item-based inverted index** $E^I$ that consists of $n$ inverted lists $\{L_1, \ldots, L_n\}$.

2. **Construct the Merkle hash tree** $T$ of the inverted index.
   - Leaf $l_j$ is assigned $h_j = \text{hash}(\text{acc}(L_j)^{s+j})$.
   - Internal node $v$ with children $c_1, \ldots, c_k$ is assigned $h_v = \text{hash}(h_{c_1} \parallel \ldots \parallel h_{c_k})$.

**Mapping to the set intersection verification problem**

Verifying whether any itemset $I$ is included in a set of transactions $T^I$ is equivalent to verifying whether $T^I$ is the correct intersection of the inverted lists of all items in $I$.

---

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Basic Solution

Drawbacks
- Total number of proofs is $2^n - 1$.
- Too much overhead.
Maximal frequent itemset (MFI) A subset of $F^S$ s.t. for each itemset $I \in MFI$, there does not exist any itemset $I' \in F^S$ s.t. $I \subseteq I'$.

Minimal infrequent itemset (MII) A set of itemsets that do not appear in $F^S$ s.t. for each itemset $I \in MII$, there does not exist any itemset $I' \not\in F^S$ s.t. $I' \subseteq I$.

(Itemsets in dotted rectangles are maximal frequent itemsets.)

**Advantage** $|MFI| + |MII| \ll |F^S| + |\overline{F^S}|$
Optimized Solution

Verification Preparation
• build $E^I$ of $D$
• build $T$ of $E^I$
• keep $\text{Sig}(E^I) = \text{root}(T)$

Proof Construction

Result Verification
• correctness verification with $MFI$
• completeness verification with $MII$

Security Analysis Our optimized solution provides the same security guarantee as the basic solution.

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Complexity

Proof construction at server side \( O(M\log^3 M + n^\epsilon \log n) \)
- \( M = \sum_{I \in M_{FI} \cup M_{II}} \sum_{i \in I} |L_i| \)
- \( n \) is the number of unique items of \( D \).
- \( \epsilon \in (0, 1) \)

Verification at client side \( O(N + F) \)
- \( N = \sum_{I \in M_{FI} \cup M_{II}} |I| \)
- \( F = \sum_{I \in M_{FI} \cup M_{II}} \text{sup}(I) \)

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Experiments

- Environment
  - Language: C++
  - Testbed: Macbook Pro, 2.4GHz CPU, 4 GB memory

- Dataset
<table>
<thead>
<tr>
<th>Dataset</th>
<th># of trans.</th>
<th># of items</th>
<th>Avg. trans. length</th>
<th>$min_{sup}$</th>
<th># of freq. itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$10^3$</td>
<td>49</td>
<td>10</td>
<td>250</td>
<td>36</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$10^4$</td>
<td>49</td>
<td>10</td>
<td>250</td>
<td>3854</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$10^5$</td>
<td>49</td>
<td>10</td>
<td>250</td>
<td>149744</td>
</tr>
<tr>
<td>$S_4$</td>
<td>$10^6$</td>
<td>49</td>
<td>10</td>
<td>250</td>
<td>3074610</td>
</tr>
<tr>
<td>$R$</td>
<td>500</td>
<td>100</td>
<td>2.4</td>
<td>5</td>
<td>97</td>
</tr>
</tbody>
</table>

- Simulation of malicious actions
  - Error ratio: $r = 1\%, 2\%, 5\%, 10\%, 20\%$
  - Incomplete: Randomly delete $r$ percent mining result.
  - Incorrect: Randomly insert $r$ percent infrequent itemsets.
Proof Optimization Ratio & Verification Time

Optimization Ratio & Verification Time ($R$ dataset)

(a) Proof optimization ratio

(b) Client verification time

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Scalability

Scalability (error ratio=1%)

(a) Construction time of one proof (itemset length = 3)

(b) Client verification time

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
References


Thank you!

Questions?
Related Work

Verifiable Computation

- [Bab85, GMR89, PRV12, GGP10] the expensive pre-processing phase is amortized over the future executions.

Integrity Verification of Database-as-a-Service (DaS)

- [PJRT05, Sio05, XWYM07] provide assurance for SQL query results.

Integrity Verification of DMaS

- [WCH+09, DLW13] only provide probabilistic result integrity guarantee.
- [LWM+12, RHPH13] focus on other mining tasks (outlier detection, clustering)

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Related Work

Verifiable Computation

- [Bab85, GMR89, PRV12, GGP10] the expensive pre-processing phase is amortized over the future executions.

Integrity Verification of Database-as-a-Service (DaS)

- [PJRT05, Sio05, XWYM07] provide assurance for SQL query results.

---

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Related Work

Verifiable Computation

• [Bab85, GMR89, PRV12, GGP10] the expensive pre-processing phase is amortized over the future executions.

Integrity Verification of Database-as-a-Service (DaS)

• [PJRT05, Sio05, XWYM07] provide assurance for SQL query results.

Integrity Verification of DMaS

• [WCH+09, DLW13] only provide probabilistic result integrity guarantee.
• [LWM+12, RHPH13] focus on other mining tasks (outlier detection, clustering)

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang
Comparison on $S_1$ dataset

<table>
<thead>
<tr>
<th>$min_{sup}$</th>
<th># of Freq. Itemsets</th>
<th>Client side</th>
<th>Server side</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Verify</td>
<td>Proof prep.</td>
</tr>
<tr>
<td>402</td>
<td>10</td>
<td>0.000164</td>
<td>24.72</td>
</tr>
<tr>
<td>203</td>
<td>50</td>
<td>0.001358</td>
<td>266.985</td>
</tr>
<tr>
<td>157</td>
<td>99</td>
<td>0.00332</td>
<td>572.591</td>
</tr>
</tbody>
</table>

(time measured in seconds)

Integrity Verification of Outsourced Frequent Itemset Mining with Deterministic Guarantee. ICDM 13. Dong, Liu, Wang