**Data-mining-as-a-service (DMaS) Model**

- **Service provider (server)** - provides storage and computational power as the service.
- **Data owner (client)** - outsources dataset and data mining computations to the server.

**Problem:**
How can the client of weak computational power to verify that \( R = F(D) \) efficiently?

**Our Problem Setting**

- **Outsourced data mining computation**: K-means clustering
- **Lloyd’s method**
- **DMaS paradigm**: infrastructure-as-a-service (IaaS)
  - The client outsources both data and the code of Lloyd’s method to the IaaS service provider (server).
  - The code is executed at the server side.
  - The server provides storage and hardware for the computation.
- **Type of dishonest server**: the sloppy server that intends to terminate the clustering early to save computational cost

**Our Solutions in a Nutshell**

- **Deterministic approach**
  - Returns 100% certainty of clustering correctness.
  - Key idea: use the Voronoi diagram to pick the neighboring centroids for verification.
- **Probabilistic approach**
  - Returns a probabilistic correctness guarantee.
  - Key idea: insert synthetic clusters for verification.
- **Comparison of our two solutions**

<table>
<thead>
<tr>
<th>Approach</th>
<th>Complexity of Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>( O(nk_d + k \log k + k^{(d+2)/2}) ) if ( k &gt; (d+1)(d+2)/2 ) and ( n &gt; k^{(d+2)/2} - 1 )</td>
</tr>
<tr>
<td></td>
<td>( O(nk) ) otherwise</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>( O(m) )</td>
</tr>
</tbody>
</table>

- \( n \): |D|
- \( m \): # of artificial tuples;
- \( k \): number of clusters in D;
- \( d \): # of attributes of D;
- \( k_d \): the average number of centroid neighbors in Voronoi diagram

**Deterministic Approach**

- **Goal**: verify each tuple has been assigned to the nearest centroid
- **Brute-force approach**: for each tuple, compute its distance to all cluster centroids.
- **Our approach**: for each tuple, compute its distance to all of its Voronoi centroid neighbors.

**Probabilistic Approach**

- **Goal**: achieve \((\alpha, \beta)\) - correctness - verify the cluster result of accuracy \( \beta \) with belief probability no less than \( \alpha \).
- **Our approach**: insert a set of artificial tuples \( AT \) that is well-separated from \( D \) so that \( AT \) will not influence the original clusters.

**Experiments**

<table>
<thead>
<tr>
<th>Detection Probability</th>
<th>Mining Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Verification Time**

Our verification approaches only takes at most 1.2% time of k-means clustering.

**Next Steps**

- Extend to the server with more cheating power
- Extend to Software-as-a-service (SaaS) paradigm
  - The client only outsources the dataset
  - The server runs its own code
- Extend to other Centroid-based Clustering Algorithms (e.g., k-medoids clustering method)