The Security Assessment of Association Mining Algorithms

Tom Johnsten
T-Johnsten@wiu.edu
Computer Science Department
Western Illinois University
Macomb, IL 61455, USA

Vijay V. Raghavan
raghavan@cacs.louisiana.edu
Center for Advanced Computer Studies
University of Louisiana Lafayette
Lafayette, LA 70504, USA

Kevin R. Hill
KR-Hill@wiu.edu
Computer Science Department
Western Illinois University
Macomb, IL 61455, USA

Abstract

Association mining algorithms can be used to discover sensitive knowledge from non-protected data that are voluntarily released for mining purposes. In this paper, we present a risk assessment procedure designed to control the security and privacy threats presented by association mining algorithms. The assessment procedure is based upon a new data analysis process that we introduce called Knowledge Hiding in Databases (KHD). The specific contributions of this paper include a description of the KHD process, the development and implementation of a risk assessment procedure for association mining algorithms, and the results of an experiment conducted to assess the effectiveness of the proposed work.
1 Introduction

Nowadays companies and organizations frequently use KDD technology to analyze their stored data to discover valuable patterns or rules that can help them maintain their competitive edge. Recently, researchers within the information security community have begun to examine the impact of this technology on database security [1,3,4,5,6,7,8,9,13]. The primary focus of this work has been on the security threat presented by classification mining. In contrast, the work reported on in this paper is concerned with the security and privacy threat presented by association mining [7,13].

Association mining, like classification mining, presents a threat to database security. To illustrate this threat, consider the data shown in Tables -1 and -2. Suppose that each transaction in Table-1 corresponds to items purchased at a local health store. The Cust-Id data is obtained from loyalty cards that the health store offers to its customers. To acquire a loyalty card a customer fills out a form that includes such personal information as name and home address. A loyalty card benefits customers by making them eligible for a discount on purchased items. The security threat presented in this example is the extent to which the collected data facilitates the identification of individuals with specific health conditions. For example, a store employee could construct a concept hierarchy to identify transactions from Table-1 associated with a particular health condition. After which, the employee could apply an association mining algorithm, along with the hierarchy, to identify individual customers from Table-2 associated with the targeted health condition.

In [5], Clifton and Marks provide an alternative example to illustrate the security threat presented by association mining. Their example is based on a supermarket that releases its customer-transaction data to a wholesale distributor. The distributor contends that the data will allow them to track inventory and reduce warehouse costs. However, the distributor, through the application of an association mining algorithm, determines that customers who purchase their competitor’s product X also typically purchase product Y. As a result, the distributor offers customers a discount on product Y when purchased in conjunction with their own equivalent for product X. The security threat is the extent to which the discovered information may result in unfair competition among the distributors.

The two examples presented above motivate the need to develop countermeasures to address the security and privacy threats related to association mining. To that end, the rest of this paper is organized as follows. Section 2 presents a methodology to analyze the impact of KDD technology on database security. The methodology consists of five steps that make up a new data analysis process that we call Knowledge Hiding in Databases (KHD). The goal of KHD, in contrast to KDD, is the non-trivial hiding of potentially sensitive knowledge in data. Section 3 outlines a risk assessment procedure designed to control the security and privacy threats presented by association mining. The proposed procedure is based upon the KHD methodology presented in Section 2. Section 4 describes a security assessment tool developed as part of this work. The tool, referred to as the Association Mining Risk Assessment Tool (AMRAT), allows for the application of the assessment procedure presented in Section 3. Section 5 describes the results of an experiment that was conducted to assess the effectiveness of the assessment procedure as implemented by AMRAT. Finally, Section 6 summarizes the paper and discusses some future related research projects.
<table>
<thead>
<tr>
<th>Cust-Id</th>
<th>St.-Johns-Wort</th>
<th>Ginkgo-Biloba</th>
<th>Cats-Claw</th>
<th>Maitake</th>
<th>LAR_PUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>023</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>004</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>006</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

Table 1: Health Store Transaction Data.

<table>
<thead>
<tr>
<th>Cust-Id</th>
<th>Name</th>
<th>Address</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Chris Bennet</td>
<td>701 S. Randolph</td>
<td>...</td>
</tr>
<tr>
<td>004</td>
<td>Linda Jones</td>
<td>100 W. Jefferson</td>
<td>...</td>
</tr>
<tr>
<td>006</td>
<td>Steve Anderson</td>
<td>808 E. Murray</td>
<td>...</td>
</tr>
<tr>
<td>011</td>
<td>Anne Richardson</td>
<td>107 N. Chandler</td>
<td>...</td>
</tr>
<tr>
<td>023</td>
<td>Jack VanBrooker</td>
<td>218 W. Poneer</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

Table 2: Health Store Customer Data.

2 KHD Methodology

As mentioned in the introduction, the goal of Knowledge Hiding in Databases (KHD) is the non-trivial hiding of potentially sensitive knowledge in data. We define the term ‘non-trivial’ to imply that knowledge is concealed in a manner that maximizes the amount of released data and maintains, to the greatest extent possible, the integrity of the data. In this context, data integrity is defined as the ability to obtain accurate knowledge from parts of the data that are legitimately available. The inputs into the KHD process are a set of security constraints that must be satisfied and a collection of data $D$; and, the output is a collection of data $D'$, derived from $D$, that satisfies the given security constraints. In general, KHD is an iterative process consisting of the following five steps:

- Identify Sensitive Knowledge
- Identify Data Mining Algorithms
- Formulate Security Policies
- Risk Assessment
- Sanitize Data

The first step is to identify the sensitive knowledge that needs to be concealed within the data. For example the statement, "store employees may not have knowledge of individual customers suffering from sensitive health conditions", represents knowledge, or information, that an organization may wish to conceal. The identification of sensitive knowledge is an important task since the resulting analysis will only be as complete as the identified knowledge. In general, such knowledge is derived from an organization’s stated security policies and its collected data. The second step is to match the knowledge identified in the first step with
data mining algorithms that are capable of discovering it. For example, the previous inference related to "sensitive health conditions" could be discovered through the application of an association mining algorithm. Of course, if there does not exist a data mining algorithm that is capable of discovering some piece of knowledge K, then K is concealed within the data.

The third step is to translate the sensitive knowledge identified in the first step into security policies that can be evaluated against the collected data. The structure of the policies will depend, in large part, on the classes of data mining algorithms identified in the second step. The fourth step is to evaluate the formulated policies against the collected data to determine if there are policy violations. If it is determined that there will be violations, then it is necessary to sanitize the data in order to conceal, or hide, the sensitive knowledge. In general, modifying a collection of data involves altering existing data, concealing existing data, or introducing additional (spurious) data. The specific manner in which data are modified is dependent upon several factors including, the data mining algorithms identified in the second step, the need to maximize the amount of released data, and the need to maintain the integrity of the data. A subtle, but significant, issue is that in some instances the act of modifying the data may affect the fundamental behavior of the targeted data mining algorithms [9].

3 KHD Process: Association Mining

In this section, we examine in greater detail the KHD process in the context of association mining.

3.1 Sensitive Knowledge

Association mining can be used to discover sensitive knowledge that is expressed as an association among stored items. For example, the discovery of associations among the items in Table-1 may lead to the disclosure of customers suffering from sensitive health conditions. In order to identify such knowledge, or information, we propose the use of fault-trees along with an organization’s security policies.

The construction of a fault-tree requires the identification of a top-event and the causal events related to the top-event [11]. In the current context, a top-event represents a specific piece of knowledge that an organization wishes to conceal within their collected data; while, a causal event represents either the discovery of a piece of knowledge or the occurrence of transactions that include a subset of items. Causal events that represent the discovery of a piece of knowledge are specified as non-terminal nodes and those that identify specific transactions are represented as terminal nodes. An example of a constructed fault-tree is shown in Figure-1. This fault-tree corresponds to the health store example in Table-1, where the sensitive knowledge includes transactions involving the purchase of items related to depression or AIDS, or transactions that include items frequently occurring in purchases of $25.00 or more. There are three terminal nodes in the fault-tree. Individually, these nodes express an association between a CUST_ID item and depression items, a CUST_ID item and AIDS items, and a subset of stored items and a LAR_PUR item. The terms depression and AIDS represent concepts that are defined in terms of items stored in the database. These
two concepts are defined in Figure-2.

### 3.2 Data Mining Algorithms

As stated above, the second step in the KHD process is to match the knowledge identified in the first step with data mining algorithms that are capable of discovering it. Our focus here is on the sensitive knowledge that can be discovered through association mining. For instance, the knowledge defined in Figure-1 is relevant since it may be obtained through the application of an association mining algorithm.

### 3.3 Security Policies

This step requires transforming a constructed fault-tree into an appropriate set of security policies. To facilitate such a transformation, we propose the use of a predefined collection of templates. A template represents one or more association rules and has three user-defined threshold values that correspond to a measured level of support, confidence and improvement. The measurements of support, confidence and improvement are typically used to evaluate the accuracy and strength of a discovered association rule [2,10].

The support and confidence of an association rule, $X \rightarrow Y$ (where $X$ and $Y$ are sets of items), are defined as the percentage of transactions in the database that contain both $X$ and $Y$, and the percentage of transactions containing $X$ that contain $Y$, respectively. The measurement of improvement is defined as the joint probability of $X$ and $Y$ divided by the product of the probability of $X$ and the probability of $Y$ [2]. If the improvement is greater than one, the rule is better at predicting the result than random chance.
The user-defined threshold values reflect the sensitivity of the knowledge represented by the template. A security policy defined in terms of a template is satisfied if and only if the calculated level of support, confidence and improvement are all below their corresponding threshold value. Our concept of a "template" is analogous to the concept of an "inclusive" template as used by association mining algorithms to direct the knowledge discovery process [10]. However, the proposed templates represent knowledge for which the reported accuracy should be relatively low (in spite of the robustness of the data mining algorithm).

As a part of this work, we have designed twelve distinct templates for specifying association-based security policies. The templates are listed in Figure-3. Its our contention that most, if not all, association-based security policies can be specified in terms of these twelve templates. The templates are defined in terms of a specific item, any item, any subset of items and a specific concept. In particular, a "specific" item is a single named item, an "any" item is a placeholder for an arbitrary single item, an "any subset" item is a placeholder for an arbitrary subset of items, and a "specific" concept is a named subset of items. Several of the templates, including template types -2, -3, -4, -5, -8, -9, -10 and -11, represent multiple association rules. Figure-4 shows the templates corresponding to the fault-tree in Figure-1. The first and second templates identify customers associated with transactions involving the purchase of items related to Depression and AIDS, respectively. The third template identifies subsets of items that occur with transactions involving the LAR_PUR item.

3.4 Risk Assessment

The risk assessment step requires the evaluation of security policies that have been defined in terms of the templates in Figure-3. This type of evaluation is in contrast to the evaluation of individual association rules described in [7] and [13]. An outline of the procedure to evaluate policies defined in terms of templates is shown in Figure-5. First, each template $T$ is expanded into one or more corresponding association rules. Second, each rule is evaluated against the collected data using the measures of support, confidence and improvement. Finally, each rule is recorded along with an indication as to whether there is a violation of security policy. Section 4 describes the implementation of this procedure in the context of AMRAT.
Figure 3: Security Templates for Association Mining.

<table>
<thead>
<tr>
<th>Type</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-1</td>
<td>Specific Item -&gt; Specific Item</td>
</tr>
<tr>
<td>Type-2</td>
<td>Specific Item -&gt; Any Item</td>
</tr>
<tr>
<td>Type-3</td>
<td>Any Item -&gt; Specific Item</td>
</tr>
<tr>
<td>Type-4</td>
<td>Specific Item -&gt; Any Subset of Items</td>
</tr>
<tr>
<td>Type-5</td>
<td>Any Subset of Items -&gt; Specific Item</td>
</tr>
<tr>
<td>Type-6</td>
<td>Specific Item -&gt; Specific Concept</td>
</tr>
<tr>
<td>Type-7</td>
<td>Specific Concept -&gt; Specific Item</td>
</tr>
<tr>
<td>Type-8</td>
<td>Any Item -&gt; Specific Concept</td>
</tr>
<tr>
<td>Type-9</td>
<td>Specific Concept -&gt; Any Item</td>
</tr>
<tr>
<td>Type-10</td>
<td>Any Subset of Items -&gt; Specific Concept</td>
</tr>
<tr>
<td>Type-11</td>
<td>Specific Concept -&gt; Any Subset of Items</td>
</tr>
<tr>
<td>Type-12</td>
<td>Specific Concept -&gt; Specific Concept</td>
</tr>
</tbody>
</table>

Figure 4: Example Association Templates.

3.5 Sanitizing Data

The process of sanitizing data requires reducing an association rule’s reported level of accuracy and strength to ensure that there is no violation of policy [7,13]. In this context, we treat both the left-hand side and the right-hand side of an association rule as a logical expression of stored items. We present two different approaches for sanitizing a collection of data.

One approach is to remove items from the database to ensure that the association rules that violate policy no longer exist. In order to maximize the amount of available data, we introduce the concept of a Minimum Coverage Item Set (MCIS). We define a MCIS as follows. Given a set of association rules \( \mathcal{A} \), a MCIS is a minimum set of items in which at

Figure 5: Risk Assessment Algorithm.
least one of the items in the set is included in each rule \( a \in A \). For example, the set of items, \( \{I1, I6\} \), is a MCIS with respect to the association rules in Figure-6. The concealment of items I1 and I6 from the database guarantees that all rules in Figure-6 have zero support, confidence and improvement. This approach to sanitizing data maintains the integrity of the data since the values of individual items are not altered. The significance of which is that any "non-sensitive" association rule discovered from the sanitized data will have the same measured level of support, confidence and improvement as with respect to the unsanitized data.

An alternative approach is to selectively modify certain item values [7,13]. We have developed three separate procedures designed to reduce an association rule’s level of support, confidence and improvement to ensure that there is no violation of policy. The developed procedures are outlined in Figure-7 and Figure-8. In each procedure, the primary step is the determination of the number of item values, \( N \), that must be modified in order to achieve the desired outcome. To calculate this quantity, the given association rule is partitioned into three parts, its left-hand side (LHS), its right-hand side (RHS) and the logical conjunction of its left-hand side and right-hand side (LHSRHS). In Figure-7 and Figure-8, the symbols \#(LHS), \#(RHS) and \#(LHSRHS) denote the number of transactions that satisfy the rule’s LHS, RHS and LHSRHS, respectively. In the case of improvement the strategy is to reduce the number of transactions that satisfy a rule’s LHSRHS and RHS until the rule’s improvement level is below one. Once the value of \( N \) is determined, the next step is to select an item from the rule’s RHS for which to alter \( N \) of its values. If the domain of the selected item is binary, then \( N \) of its values are replaced with zeros; otherwise, \( N \) of its values are replaced with NULL. For example, if \( N \) is ten and the selected (item, value) pair is (I1, '123'), then ten random occurrences of '123' belonging to item I1 are replaced with NULL.

The transactions that are altered must not only include the selected RHS item but must also satisfy the rule’s logical conjunction LHSRHS.

The advantage of the second approach is that it maximizes the amount of available data, although it does not ensure the integrity of the data. In general, the modification of item values will alter the discovery of "non-sensitive" association rules. However, it is possible to determine the impact that the modification of an item’s values has on a rule’s measured support and confidence. Specifically, for a given rule \( X \rightarrow Y \) discovered from the sanitized data there are four possible cases to consider and they are outlined in Figure-9.

Figure 6: MCIS Example.
Support:

// Number of items to modify
N = #(LHSRHS) − (threshold_support * num_transactions) + 1
Randomly select N transactions from the database that satisfy
the expression (LHSRHS)
For each selected transaction
select a RHS item and alter its value

Confidence

// Number of items to modify
N = (#(LHSRHS) − (threshold_confidence * #(LHS)) + 1
Randomly select N transactions from the database that satisfy
the expression (LHSRHS)
For each selected transaction
select a RHS item and alter its value

Figure 7: Data Sanitizing Procedures: Support and Confidence.

Improvement

// Number of items to modify
LHSRHS_temp = #(LHSRHS)
RHS_temp = #(RHS)
done = false
while (!done & (LHSRHS_temp > 0))
    diff = (LHSRHS_temp * num_transactions) − (#(LHS) * RHS_temp)
    if (diff > 0)
        LHSRHS_temp = LHSRHS_temp − 1
        RHS_temp = RHS_temp − 1
    else
        done = true
N = #LHSRHS − LHSRHS_temp
Randomly select N transactions from the database that satisfy
the expression (LHSRHS)
For each selected transaction
select a RHS item and alter its value

Figure 8: Data Sanitizing Procedure: Improvement.
sanitized data that satisfy the logical expression \((X \& Y)\), and \(P_{\text{MAX}}(X)\) is the maximum number of altered values over the items \(x \in X\). To illustrate, suppose the non-sensitive association rule, \(x_1 \& x_2 \rightarrow y\), is discovered in the sanitized data with a measured level of support and confidence of \((3/100 = 3\%)\) and \((3/50 = 60\%)\), respectively. If \(P_{\text{MAX}}(X)\) is 6, then the actual level of support and confidence is within the interval \([30.0\%, 36.0\%]\) and \([53.5\%, 64.2\%]\), respectively. Case-III addresses the situation in which items whose values have been modified are included as part of the rule’s right-hand side. \(P_{\text{MAX}}(Y)\) is the maximum number of altered values over the items \(y \in Y\). Consider again the non-sensitive association rule, \(x_1 \& x_2 \rightarrow y\), with support equal to 30\% and confidence equal to 60\%. If \(P_{\text{MAX}}(Y)\) is 10, then the actual level of support and confidence is within the interval \([30.0\%, 40.0\%]\) and \([60.0\%, 80.0\%]\), respectively. Finally, Case-IV addresses the situation in which items whose values have been modified are included as part of the rule’s left-hand side and right-hand side. \(P_{\text{MAX}}(X,Y)\) is the maximum number of altered values over the items \(x \in X\) and \(y \in Y\). The significance of establishing intervals of support and confidence is that such information can be used to quantify the integrity of the sanitized data.

### 4 AMRAT: Association Mining Risk Assessment Tool

We have developed a tool, referred to as the Association Mining Risk Assessment Tool (AMRAT), that implements the risk assessment procedure outlined in Section 3. The tool allows a security administrator to systematically alter a collection of data so that there is no violation of a stated security policy. The architecture of AMRAT is shown in Figure-10. The input into the tool is one or more concept hierarchies, a set of association templates and a collection of stored data. The concept hierarchies and association templates are created by the security administrator from the organization’s security policies. The output produced by AMRAT includes a collection of sanitized data and a risk assessment report. Currently,
AMRAT allows a security administrator to sanitize a collection of data by either concealing items from the data or by altering item values. The risk assessment report provides a security administrator with a summary of the sanitation process.

As shown in Figure-10, AMRAT is comprised of three components, concept analyzer, association analyzer and data sanitizer. The role of the concept analyzer is to provide the association analyzer with information regarding the structure of the given concept hierarchies. This information includes the individual items that comprise a named concept. AMRAT allows concepts to represent expressions of items defined in terms of the logical operators AND and OR. The role of the association analyzer is to expand the given association templates into individual association rules and to evaluate each of the rules against the collected data. The evaluation process determines if a rule violates its corresponding security policy. AMRAT supports security policies defined in terms of support, confidence and improvement. The results of the association analyzer are utilized by the data sanitizer. The primary role of the sanitizer is to transform the collected data to ensure that there are no violations of security policies. As mentioned above, AMRAT allows a security administrator to choose a particular method for altering the data. The process of altering item values with respect to a specific rule may cause a violation of security policy with respect to other rules. As a result, the sanitizer iteratively evaluates the association rules until all stated rules are evaluated with no reported policy violations.

5 Experimental Investigation

An experiment was conducted to assess the effectiveness of the proposed assessment procedure. As part of the experiment, a small transaction database was constructed that contained
sixteen items related to a health store. The sixteen items included a CUST-ID item that associated a customer with each transaction and a total of eight items associated with AIDS and depression. The database contained twenty-five distinct customers and a total of one-hundred transactions. The objective of the experiment was to determine if the application of the assessment procedure would conceal knowledge about customers associated with items related to either AIDS or depression.

The execution of the experiment required the construction of a set of association templates along with a corresponding concept hierarchy. The constructed templates and hierarchy are shown in Figure-11 and Figure-12, respectively. The templates in Figure-11 represent security policies defined with a confidence value of 0.5 and a support value of 1.0. In other words, there is a violation of security policy when at least half of a customer’s transactions include items related to AIDS or depression.

The experiment itself consisted of three steps. First, the transaction database was mined using IBM’s Intelligent Miner to ensure the presence of customers associated with AIDS or depression [12]. Second, the constructed database was evaluated using AMRAT along with the specified templates and concept hierarchy. Figure-13 shows the rules, corresponding to the templates in Figure-11, that were determined by AMRAT to be in violation of a stated security policy. The evaluation process was performed twice, once in which the database was sanitized through the removal of items and a second time in which the database was sanitized through the modification of item values. In the first approach, not surprisingly, the CUST_ID item was removed and in the second approach twenty-two item values were altered (changed from 1 to 0). Specifically, in the latter case seven values of the item Kava were altered, two values of the item Ginkgo Biloba were altered, one value of the item Thistle was altered, seven values of the item Babchi were altered, and five values of the item Maitke were altered. Finally, the two sanitized databases were mined using IBM’s Intelligent Miner. In both instances there were no violations of security policies. The results suggest that the application of the assessment procedure has the potential to control the security and privacy threats presented by association mining.

6 Conclusion and Future Work

In this paper, we proposed a risk assessment procedure for controlling the security and privacy threats presented by association mining. The proposed procedure is based upon a new data analysis process called Knowledge Hiding in Databases (KHD). The goal of KHD is the non-trivial hiding of potentially sensitive knowledge in data. The KHD methodology consists of five steps: identify sensitive knowledge, identify data mining algorithms, formulate security policies, risk assessment, and sanitize data. These five steps form the basis of our assessment procedure. The results from our initial experiments suggest that the assessment
Figure 12: Health-Store: Concept Hierarchy.

Figure 13: Non-Secure Association Rules.

procedure is an effective countermeasure against the security and privacy threats presented by association mining algorithms.

We have several additional research projects planned with respect to this work. Our immediate plans include additional experiments to further evaluate the developed assessment procedure, investigate alternative approaches to sanitizing data, and to extend AMRAT to allow greater control over the risk assessment procedure and to provide more detailed information on the integrity of the sanitized data. We also plan to extend the current work to include the security and privacy threats presented by the discovery of sequential patterns. Finally, we plan to investigate the development of a new data analysis process that is similar to KHD called Knowledge Fabrication in Databases (KFD). The goal of KFD, in contrast to KHD, is to alter a collection of data so that it contains knowledge that is intentionally misleading. A potential application of KFD is the release of data by government agencies in the context of homeland security.
References


