圖書流通紀錄之一般化相關規則找尋之研究
The Research on Finding Generalized Association Rules from Library Circulation Records

研究生：洪志淵 撰

中華民國九十年七月
Contents
The Research on Finding Generalized Association Rules from Library Circulation Records ...........................................................................................................................0
Abstract ..................................................................................................................................................................................5
Chapter 1 . Introduction .................................................................................................................................6
1.1. Motivation ..................................................................................................................................................6
1.2. Thesis Organization ..................................................................................................................................7
Chapter 2 . Problem Description .....................................................................................................................8
Chapter 3 . Our Approach ..................................................................................................................................13
3.1. Identifying Relevant Patron Attributes ..................................................................................................13
3.2. Algorithms for Generating Large Itemsets ............................................................................................14
3.2.1. Problem Definition ..........................................................................................................................14
3.2.2. Lattice Structure ..............................................................................................................................15
3.2.3. Algorithm Basic ...............................................................................................................................16
3.2.4. Algorithm K-pass .............................................................................................................................18
3.2.5. Algorithm MergePrune ......................................................................................................................20
3.3. Identifying Interesting Rules ...................................................................................................................25
Chapter 4 . Evaluation Plan ..........................................................................................................................30
4.1. Generation of Synthetic Data ................................................................................................................30
4.2. Relative Performance of Algorithms ....................................................................................................33
Chapter 5 . Empirical Results .......................................................................................................................39
5.1. Identifying relevant patron attributes ...................................................................................................39
5.2. Generating patron-book rules ...............................................................................................................41
5.3. Pruning uninteresting rules ..................................................................................................................42
5.4. Effectiveness of the patron-book rules .................................................................................................43
Chapter 6 . Literature Review .........................................................................................................................49
6.1. SDI (Selective Dissemination of Information) ......................................................................................49
6.2. Recommendation Approaches ..............................................................................................................50
6.3. Data Mining Methodology ...................................................................................................................52
6.4. Clustering ...............................................................................................................................................53
6.5. Classification ..........................................................................................................................................53
6.6. Association Rules ..................................................................................................................................54
Chapter 7 . Conclusion .......................................................................................................................................59
7.1. Summary ...............................................................................................................................................59
7.2. Contributions .........................................................................................................................................59
Bibliography .................................................................................................................................................................60
List of Figures

Figure 2-1: Example of Book Taxonomy.................................................................10
Figure 2-2: Example of Patron Taxonomy.............................................................11
Figure 3-1: Example of two concept hierarchies HA1, HA2.................................16
Figure 3-2: Example of lattices LatA1A2..............................................................16
Figure 3-3: Pseudo-code for Algorithm Basic ......................................................18
Figure 3-4: Pseudo-code for algorithm k-pass ....................................................20
Figure 3-5: Example of prune technique..............................................................23
Figure 3-6: Pseudo-code for algorithm MergePrune ...........................................25
Figure 3-7: Pseudo-code for identifying interesting rules ...................................29
Figure 4-1: Changing merged attribute..............................................................35
Figure 4-2: The impact of merge and prune......................................................36
Figure 4-3: Number of candidates on each pass .............................................36
Figure 4-4: The scalability of “MergePrune” ...................................................37
Figure 4-5: The impact of number of patron attributes ....................................38
Figure 5-1: Candidate attribute values..............................................................40
Figure 5-2: Simplified Decision Tree .................................................................41
Figure 5-3: The relative performance of Basic, K-pass, and MergePrune on NSYSU circulation data .................................................................42
Figure 5-4: The relative rules remain.................................................................43
Figure 5-5: User filled in user’s program, status, and acceptable book numbers .................................................................44
Figure 5-6: User marks interesting books............................................................45
Figure 5-7: The relationships between Precision, Recall, and Classification accuracy .........................................................................................46
List of Tables
Table 2-1: Example of Circulation Database ................................................. 11
Table 4-1: Book Parameters for Synthetic Data Generation ......................... 30
Table 4-2: Patron Parameters for Synthetic Data Generation ....................... 31
Table 4-3: Transaction Parameters for Synthetic Data Generation ............... 32
Table 4-4: Parameter settings ..................................................................... 34
Table 5-1: Results of New Book Recommender System .............................. 47
Abstract
Libraries have long been widely recognized as import information-offering institutes. Thousands of new books are acquired per month by our university—a mid-sized university in Taiwan), and patrons may have difficulties identifying the small set of books that really interest them. This gives rise to the problem of finding an effective way to recommend patrons the newly arrived books in a library. In this work, we address this problem in finding generalized association rules between patrons and books. We first discuss how to identify relevant but independent patron attributes in regard of the books they checked out. Then, we propose a set of algorithms for generating large itemsets and evaluate their performance experimentally. In addition, we define interestingness of rules and propose an algorithm for pruning uninteresting rules. Finally, we apply our approach to the circulation data of National SUN Yat-Sen University library and report our experiences.
Chapter 1. Introduction

1.1. Motivation
Libraries have long been widely recognized as important information-offering institutes that preserve and provide their patrons with periodicals, newspapers, encyclopedia, dictionaries, audio and video types of data, online public access catalog, and more recently the electronic databases. Almost every university student has the experience of using library services provided by his university library. Students can circulate books, read magazines, watch films, and copy papers in the library. With the advent of Internet, modern libraries have provided services that crossed the boundary of their physical buildings. A great amount of information traditionally placed on the shelves of the library has now been available on the Internet. Virtually every traditional library service has its counterpart on the Internet. For example, most university students can check book circulation status, request a checked-out books to be returned, access to a great number of electronic databases, and even watch films, all from the web sites of their university libraries. Speaking of the new book recommendation, besides posting the list of new books on some physical bulletin board inside the library, librarians today have also posted this message on some electronic bulletin board and library homepage. However, this list could be long (e.g., thousands of new books are acquired per month by our university—a mid-sized university in Taiwan), and patrons may have difficulties identifying the small set of books that really interest them. In the publishing industry, an increasing number of customized services have been developed. The most noted example is from amazon.com, which provides customized book information to enhance customers’ motivation of buying books and in turn increases its revenue. However, similar work does not seem to exist in the library community.
Most libraries today use computerized software systems that provide on-line public access catalogs (OPAC) and that facilitate book circulation. As a consequence, massive amounts of circulated book transaction data have been collected and stored in databases of OPAC systems. In this thesis, we develop techniques that identify the associations between patrons and the types of books in which they are interested from the circulation database. These associations are in turn used to selectively choose from a set of newly arrived books and recommend to patrons.

1.2. Thesis Organization
This thesis is organized as follows. A detailed problem description is given in chapter 2. As patrons are associated with quite a number of attributes, those that are related to the types of books they checked out need to be identified first to reduce the computation overhead of subsequent rule mining. Chapter 3 deals with the generation of patron-book association rules in detail. It first is devoted to the discussion of how to identify relevant but independent patron attributes in regard of the books they checked out. Then, several algorithms designed for generating large itemsets are described. As the rules produced by the proposed algorithms may contain some duplication, finally, we define interestingness of rules and propose an algorithm for pruning uninteresting rules. Chapter 4 describes the performance evaluation for algorithms by applying synthetic circulation data as input for our experiments. In chapter 5, we apply our approach to the circulation data of National Sun Yat-Sen University library and report our experiences. In chapter 6, we review the related work in the fields of SDI, recommendation approaches, and several data mining techniques. Chapter 7 draws some conclusions from the work described in this thesis.
Chapter 2. Problem Description

The ultimate objective of the work presented in this thesis is to find an effective way to recommend patrons the newly arrived books in a library. We propose to make use of the transaction data stored in the circulation databases available in many of today’s libraries. Circulation data records interactions among different patrons or between one patron and multiple books.

Our approach to address the problem is to find generalized association rules between patrons and books. An OPAC system is capable of recording the historical and present information about books checked out by patrons. By discovering the generalized association rules between patrons’ attributes and books, we obtain knowledge for recommending new books. The problem is how to efficiently discover these rules.

Let \( P = \{ p_1, p_2, \ldots, p_m \} \) be a set of patron literals, which constitutes all patron attribute values and their generalizations. Besides, assume that there are \( k \) patron attributes and that \( P \) are partitioned into \( k \) subsets \( P_1, \ldots, P_k \), each of which contains patron literals pertaining to a particular attribute. \( B = \{ b_1, b_2, \ldots, b_n \} \) is a set of book literals, which composes all book types. A taxonomy on books, denoted \( T(B) \), is a directed acyclic graph with the set of vertices being \( B \). A taxonomy on the \( i \)’th patron attribute, denoted \( T(P_i) \), where \( P_i \subseteq P \), is a directed acyclic graph with the set of vertices equal to \( P_i \). An edge in \( T \) represents an is-a relationship. Note that it is possible to combine several hierarchical taxonomies into a lattice taxonomy. This explains why we define a taxonomy as a directed acyclic graph rather than a hierarchy. If there is an edge in \( T \) from \( p \) to \( c \), we call \( p \) a parent of \( c \) and \( c \) a child of \( p \) (i.e., \( c \) represents a specialization of \( p \)). Let \( D \) be a set of transactions, where each transaction
\( d_i \) is a tuple \((p_{i1}, p_{i2}, \ldots, p_{ik}, b)\), where \( p_{ij} \in P_j \) and \( b \subseteq B \). We say that a transaction \( d_i=(p_{i1}, p_{i2}, \ldots, p_{ik}, b) \) supports a patron type \( p'=(p'_{i1}, p'_{i2}, \ldots, p'_{ik}) \) if \( p_{ij}=p'_{ij} \) or \( p'_{ij} \) is an ancestor of \( p_{ij} \) in the taxonomy \( T(P_j) \), \( 1 \leq j \leq k \). Similarly, we say that a transaction \( d_i=(p_{i1}, p_{i2}, \ldots, p_{ik}, b) \) supports a book item \( b_j \in B \) if \( b_j \in b \) or \( b_j \) is an ancestor of \( b \) in \( T(B) \). A patron-book association rule is an implication of the form \( X \Rightarrow Y \), where \( X \in P_1 \times P_2 \times \ldots \times P_k \), \( Y \in B \). The rule \( X \Rightarrow Y \) holds in the transaction set \( D \) with confidence \( c\% \) if \( c \% \) of transactions in \( D \) that support \( X \) also support \( Y \). The rule \( X \Rightarrow Y \) has support \( s\% \) in the transaction set \( D \) if \( s\% \) of transactions in \( D \) support both \( X \) and \( Y \). Therefore, given a set of transactions \( D \) and several patron taxonomies \( P_1, P_2, \ldots, P_k \) (each for the generalization of one patron attribute) and one book taxonomy \( B \), the problem of mining patron-book association rules in circulation data is to discover all rules that have support and confidence greater than the user-specified minimum support (called \( MinSup \)) and minimum confidence (called \( MinConf \)) respectively.

For example, let \( B=\{ \text{Natural Sciences, Mathematics, Astronomy, Physics, Chemistry, Geology, Biology, Analytical Chemistry, Qualitative Analysis, Quantitative Analysis, Theoretical Chemistry, Inorganic Chemistry, \ldots} \} \), \( P=\{ \text{Bachelor, Master, Doctor, Science, Liberal arts, Management, \ldots} \} \), and their taxonomies are shown in Figure 2-1 and Figure 2-2. Consider the database shown in Table 2-1. Let \( MinSup \) be 30\% (that is, 2 transactions) and \( MinConf \) 60\%. 

Figure 2-1: Example of Book Taxonomy
The following patron-book association rules can be discovered in this case:

Junior and Science patron  ⇒  Chemistry book

Bachelor and Physics patron  ⇒  Quantitative Analysis book

Table 2-1: Example of Circulation Database

<table>
<thead>
<tr>
<th>Tid</th>
<th>Name</th>
<th>Status</th>
<th>College</th>
<th>Book</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Lee</td>
<td>Freshman</td>
<td>Physics</td>
<td>Quantitative Analysis</td>
</tr>
<tr>
<td>200</td>
<td>Wang</td>
<td>Doctor</td>
<td>Electrical</td>
<td>Electrical Engineering, Keyboard instruments</td>
</tr>
<tr>
<td>300</td>
<td>Cheng</td>
<td>Master</td>
<td>Electrical</td>
<td>English Literature,</td>
</tr>
<tr>
<td>400</td>
<td>Peter</td>
<td>Junior</td>
<td>Chemistry</td>
<td>Theoretical Chemistry, General history of China</td>
</tr>
<tr>
<td>500</td>
<td>Mary</td>
<td>Junior</td>
<td>Physics</td>
<td>Quantitative Analysis, Methodology</td>
</tr>
<tr>
<td>600</td>
<td>Bob</td>
<td>Senior</td>
<td>Chemistry</td>
<td></td>
</tr>
</tbody>
</table>
In fact, the problem for mining useful patterns for library new book recommendation is a specialization of the problem of discovering generalized association rules from market basket database [Agra95, Han95]. The problem of the generalized association rules from market basket database is to find association between the items that may be from any level of the taxonomy, given a set of transaction and a taxonomy on the items. One can certainly employ the existing approaches to solving this problem in the library circulation context. Specifically, a circulation transaction can be seen as a market basket transaction by treating both patron and books checked out as ordinary data items. Any approach proposed, for example in [Agra95], can then be used to discover generalized associations among these data items. However, this straightforward approach is inefficient and generates many redundant rules with antecedent and consequence being of the same type (books or patrons). We therefore propose to develop efficient algorithms specifically for discovering patron-book association rules from library circulation data.

By considering taxonomies of both patrons and books, we may find that many rules are related to each other. For example, if the rule $\text{Computer science Patrons} \Rightarrow \text{Mathematics Books}$ satisfies both minimum support and minimum confidence, so too does the rule $\text{Computer Science Patrons} \Rightarrow \text{Science Books}$ as Science is an ancestor of Mathematics in the book taxonomy. Besides, we can claim that $\text{Engineering} \Rightarrow \text{Mathematics Books}$ must also preserve minimum support. Are all these highly related rules equally interesting? If not, which one should be selected, and what is the criterion for deciding interestingness? We propose a measure that determines the interestingness of generalized association rules and develop an algorithm to prune those uninteresting ones.
Chapter 3. Our Approach

3.1. Identifying Relevant Patron Attributes

Before proceeding to mine the generalized association rules that relate patrons’ attribute values and book classes, we propose to analyze the patron attributes first. The goal is to identify the attributes that are highly correlated to types of books checked out and are independent. The advice from experts in the library may be one important source about these attributes. For example, our library experts at NSYSU library suggested that college, grade, status, and working unit among the patron's attributes are highly related to the types of books in which they are interested. The most straightforward approach for verifying the knowledge of the library experts (i.e., evaluating the correlation between a patron’s attribute and book classes) is to compute correlation coefficient. In statistic domain, coefficient of simple correlation is a parameter $\gamma$ to determine the linear relation between independent variable $X$ and dependent variable $Y$. The range of $\gamma$ is $-1$ to $+1$. $\gamma = 0$ represent no association between $X$ and $Y$. $\gamma = +1(-1)$ represent totally positive (negative) association between $X$ and $Y$. However, we observe that coefficient of simple correlation can be applied only to the variables of numeric types. In order to identify the association between patron attributes and the book attribute that are of categorical types, we propose to use the classification decision tree induction technique. The classification decision tree induction technique has the ability to identify attributes that distinguish classes. In this application, book types are treated as classes, and patron attributes form the candidate attributes for distinguishing book types. To avoid the overwhelming amount of book types, we choose only book types at a particular level in the book taxonomy (say, the second level). The set of patron attributes appeared in the turning point of decision tree satisfies two properties:
1. They are likely to correlate to the classes of books checked out. A patron attribute that has a low correlation to the types of books checked out is unlikely to be chosen at the turning points.

2. They represent a set of independent patron attributes with respect to the types of books checked out. For example, for a university library in Taiwan, the two patron attributes college and sex are likely to be dependent to each other. Usually the college of liberal art is made up of mostly female students, while the college of engineering consists of mostly male students. Once one attribute is selected, other dependent attributes will not be picked.

We will report our experience in applying NSYSU library circulation data to this approach in Chapter 5.

3.2. Algorithms for Generating Large Itemsets

The bottleneck in finding association rules lies on the enumeration of large itemsets [Agra94]. In this chapter we discuss large itemset generation approaches for library circulation data. We first provide a formal definition of the problem of large itemset generation. Then, we propose three algorithms that generate large itemsets, namely algorithm Basic, algorithm K-pass, and algorithm MergePrune.

3.2.1. Problem Definition

A transaction \( t_i \) in the circulation database is denoted as

\[
  t_i = \langle p_{i,1}, p_{i,2}, \ldots, p_{i,k}, b_{i,1}, b_{i,2}, \ldots, b_{i,m} \rangle \quad (1 \leq i \leq n, k \geq 1, m \geq 1),
\]

where \( p_{i,j} \) is the \( j \)'th attribute of the \( i \)'th patron, and \( b_{i,j} \) is the book item for the \( j \)'th book that the \( i \)'th patron has ever checked out.
Note that a transaction represents the set of books a particular user has checked out over a certain period of time. The quantities and times of a book checked out are not recorded since this information is not used by our approach. Besides, taxonomies (is-a hierarchies) over patron attributes and book items are assumed to be available, and each item in a transaction comes from the leaves of the corresponding taxonomy.

A patron-book itemset is of the form \((p_{i,1}, p_{i,2}, \ldots, p_{i,j}, b)\), where \(p_{i,j} \in P\) and \(b \in B\). We say a patron-book itemset is large if it is supported by no less than a user-specified amount of transactions. The problem addressed is this chapter is how to identify all large patron-book itemsets, given a circulation database, several patron hierarchies, and a book hierarchy.

### 3.2.2. Lattice Structure

Each attribute is associated with a concept hierarchy. A concept hierarchy enables us to generalize or specialize attribute values. For example:

- \{chemistry, physics, math, …\} \(\in\) science college
- \{electrical engineering, computer science, mechanical engineering, …\} \(\in\) engineering college
- \{freshman, sophomore, junior, senior\} \(\in\) undergraduate

For a freshman patron, its’ generalization is undergraduate. The specialization of a student in engineering college could be electrical engineering. Note that multiple concept hierarchies can be combined into one specific structure called concept lattice. Suppose there are two concept hierarchies \(HA_1\) and \(HA_2\), each having three levels as shown in Figure 3-1.
HA1 and HA2 can be merged into one lattice LatA1A2 as shown in Figure 3-2:

Let HA1(X) denote the set of values in level X of the concept hierarchy HA1, and HA2(Y) the set of values in level Y of the concept hierarchy HA2. Each node (X, Y) in LatA1A2 is a set of ordered pairs HA1(X)×HA2(Y). Therefore, (1, 0) denotes the Cartesian product of the set of nodes in level one of HA1 and the set of nodes in level zero of HA2.

3.2.3. Algorithm Basic

Suppose there are $m$ patron attributes that we find relevant to the types of books checked out. Each patron attribute is associated with a unique concept hierarchy. If we merge the $m$ concept hierarchies into one big lattice, then the job of finding large
itemset becomes much simpler because large item sets can be obtained by scanning circulation database only twice: the first scanning identifies the large patron types and the large book types, while the second scanning finds the set of large 2-patron-book itemsets, each of which comprises one patron type and one book type. Specifically, we first extend each transaction $t_i = (p_{i,1}, p_{i,2}, ..., p_{i,k}, b_{i,1}, b_{i,2}, ..., b_{i,m})$ (1 ≤ i ≤ n, k ≥ 1, m ≥ 1) in the circulation database by adding all ancestors of $p_{i,j}$ in the patron lattice and all ancestors of $b_{i,j}$, 1 ≤ j ≤ m, in the book hierarchy. Let us call this extended transaction $t_i'$. Now $t_i$ supports a patron value X if and only if $X \in t_i'$, and $t_i$ supports a book type Y if and only if $Y \in t_i'$. The first pass of the algorithm simply counts item occurrences to determine the frequent 1-itemset (which could be patron-itemset or book-itemset). After that, both frequent patron-itemsets and frequent book-itemsets are identified. Then, we take the Cartesian product of these two itemsets to generate the candidate patron-book itemsets $C$. The circulation database is scanned again to identify the subset of $C$ that are frequent. This algorithm called Basic due to its straightforward manner, is listed in Figure 3-3:
Algorithm Basic

/* Assume that \( D \) is the circulation database and \( \text{MinSup} \) is the user-specified support threshold */

begin

for each transaction \( t_i = \langle p_{i,1}, p_{i,2}, ..., p_{i,k}, b_{i,1}, b_{i,2}, ..., b_{i,m} \rangle \in D, 1 \leq i \leq n, k \geq 1, m \geq 1 \), do begin

Extend \( t_i \) by adding all ancestors of \( p_{i,j} \) and \( b_{i,j'} \), \( 1 \leq j \leq k \), \( 1 \leq j' \leq m \), and removing any duplicates;

Increment the count of every item in \( t_i \);

end;

\( L_1(P) = \{ c | c.\text{count} \geq \text{MinSup}, c \text{ is in a node in the patron lattice} \} \);

\( L_1(B) = \{ c | c.\text{count} \geq \text{MinSup}, c \text{ is in a node in the book hierarchy} \} \);

\( C = L_1(P) \times L_1(B) \);

for each extended transactions \( t_j \in D \) do begin

Increment the count of each candidate \( \langle p, b \rangle \) in \( C \) such that both \( p \) and \( b \) appear in \( t_j \);

end;

\( L = \{ c | c \in C \text{ and } c.\text{count} \geq \text{MinSup} \} \);

return \( L \);

end.

Figure 3-3: Pseudo-code for Algorithm Basic

3.2.4. Algorithm K-pass

The algorithm Basic is simple. However, it may suffer from the explosive size of the candidate patron-book itemsets generated from the Cartesian product of patron-itemsets and book-itemsets. We observe that a patron-book itemset \( \langle p_{i,1}, p_{i,2}, ..., p_{i,k}, b \rangle \) is frequent only if the item set resulted from dropping any of the patron attribute values (i.e., \( p_{i,1}, p_{i,2}, ..., p_{i,k} \)) remains frequent. Specifically, let \( L_k \) denote
the large itemsets of the form \( <p_{i,1}, p_{i,2}, \ldots, p_{i,k}, b> \). A candidate itemset \( C_k \) can be generated by joining \( L_{k-1} \). This procedure is like Apriori candidate generation algorithm [Agra95] except that each itemset must contain exactly one book type. We add in each transaction in the circulation database \( D \) all ancestors of each book and patron item by referencing their own concept hierarchies (is-a hierarchies). After first scanning the dataset \( D \), we get frequent one patron itemsets \( L_1(P) \), frequent book one book itemsets \( L_1(B) \) respectively. If an item is not a member of \( L_1(P) \) or \( L_1(B) \), it will not appear in any large patron-book itemset and therefore is useless. We delete all the useless items in every transaction of \( D \) for reduce the size of \( D \). The set \( C_1 \) of candidate 1-itemsets is defined as \( L_1(P) \times L_1(B) \). The database is scanned to find the set \( L_1 \) of large 1-itemsets from \( C_1 \). A following phrase, say pass \( k \), is composed of two steps. First, we use Apriori candidate generation function to generate the set \( C_k \) of candidate itemsets by joining two large \( k-1 \) itemsets in \( L_{k-1} \) on their common \( k-2 \) patron attribute values and the common book attribute value. Next, the dataset \( D \) is scanned and the support of candidates in \( C_k \) is counted. The set \( L_k \) of large \( k \) itemsets are candidates in \( C_k \) with minimum support. The algorithm, called k-pass, is listed in Figure 3-4:
Algorithm k-pass

/* Assume that $D$ is the circulation database and $MinSup$ is the user-specified support threshold */

begin
  for each transaction $t_i =< p_{i,1}, p_{i,2}, ..., p_{i,k}, b_{i,1}, b_{i,2}, ..., b_{i,m} >\in D, 1 \leq i \leq n, k \geq 1, m \geq 1,$ do begin
    Extend $t_i$ by adding all ancestors of $p_{i,j}$ and $b_{i,j'}, 1 \leq j \leq k, 1 \leq j' \leq m,$ and removing any duplicates;
    Increment the count of every item in $t_i$;
  end;

  $L_1(P) = \{ c | c.count \geq MinSup, c \text{ is in a node in a patron hierarchy} \}$;

  $L_1(B) = \{ c | c.count \geq MinSup, c \text{ is in a node in the book hierarchy} \}$;

  $C_1 = L_1(P) \times L_1(B)$;

  for each extended transactions $t \in D$ do begin
    Delete any item in $t$ that does not appear in either $L_1(P)$ or $L_1(B)$;
  end;

  $L_1 = \text{All candidates in } C_1 \text{ with minimum support};$

  $k = 2$;

  while ($L_{k-1} \neq \emptyset$) do begin
    Join $L_{k-1}$ and $L_{k-1}$ based on one book attribute and $k$-2 common patron attributes, and
    Put the result in $C_k$;
    for each transaction $t \in D$ do begin
      Increment the count of all candidates in $C_k$ that are contained in $t$;
    end;

    $L_k = \{ c | c \in C_k \text{ and } c.count \geq MinSup \}$;

    $k = k + 1$;
  end;

end.

Figure 3-4: Pseudo-code for algorithm k-pass

3.2.5. Algorithm MergePrune

In algorithm Basic, the patron concept lattice could be of huge size because it is a product of multiple patron hierarchies. As a result, the entire candidate set is unlikely to reside in memory. On the other hand, algorithm k-pass needs to scan the circulation database for up to $k$ times, where $k$ is the number of patron attributes. As described in
[Agra94], the overall performance of mining association rules is affected by the size of the candidate set at each iteration as well as the number of database scanning. The algorithm MergePrune is designed to solve both problems.

The algorithm MergePrune, as implied by its name, is made up of two techniques, namely merge and prune. The merge technique partitions the set of patron attributes into a set of disjoint attribute groups. The concept hierarchies of attributes in the same group are merged into a lattice structure. Obviously the merge technique is a generalization of both Basic and k-pass. At the one extreme, when all attributes are assigned to be the same single group, this approach is exactly the Basic algorithm. At the other extreme, when the attributes are partitioned into $k$ groups, each of which consists of only one attribute, this approach is reduced to k-pass algorithm. By tuning the size of the partition, the merge technique intends to reduce the number of database scanning while keeping the size of the candidate set at each iteration in control.

The prune technique aims to further reduce the size of the candidate set generated at each iteration. Consider how Basic and k-pass generate the first candidate set (denoted $C$ in Basic and $C_1$ in k-pass): the product of the set of large 1-patron-itemsets ($L_1(P)$) and the set of large 1-book-itemsets ($L_1(B)$). However, we observe that it is not necessary to join every element $p$ in $L_1(P)$ and every element $b$ in $L_1(B)$ if we are aware that the combination of $p$ and $b$ cannot possibly form a large patron-book itemset. Such awareness is possible if some information can be recorded when scanning the database to obtain $L_1(P)$ and $L_1(B)$. Let $C_1(P)$ and $C_1(B)$ denote the set of all nodes in patron lattices and the set of nodes in book hierarchy respectively. The purpose of the initial database scanning is to obtain the subset $L_1(P)$ of $C_1(P)$ and
the subset $L_1(B)$ of $C_1(B)$. Let $S_p$ be the set of nodes in the second level of all patron lattices and $S_b$ be the set of nodes in the second level of the book hierarchy. For each element $b$ in $C_1(B)$, we would like to record the number of patrons of each type $p' \in S_p$ that have checked out the books of type $b$. Similarly, for each element $p$ in $C_1(P)$, we also count the number of books of each type $b' \in S_b$ that have been checked out by the patrons of type $p$. Now consider an element $p$ in $C_1(P)$ and an element $b$ in $C_1(B)$. Let $p' \in S_p$ be equal to or the ancestor of $p$ in some patron lattice and $b' \in S_b$ be equal to or the ancestor of $b$ in the book hierarchy. The combination of $p$ and $b$ is large only if both the count corresponding to $p'$ in $b$ and that corresponding to $b'$ in $p$ are no less than $MinSup$.

The same idea can be applied to the generation of candidate sets in later iterations. At the end of iteration $i$, a set $L_i$ of large $i$-patron-book itemsets is generated. Now consider an itemset $l_{ij} = <p_{j1}, p_{j2}, ..., p_{ji}, b>$ in $L_i$, where $P_{ji}$ is a node in a patron lattice and $b$ a node in the book hierarchy. Let $P_j$ be the set of patron attributes not involved in $l_{ij}$, and $N_j$ be the set of nodes in the second level of concept lattices of attributes in $P_j$. For each patron node $p'$ in $N_j$, we count the number of transactions that support $l_{ij}$ and that belong to patrons of type $p'$. Suppose we want to join two $i$-patron-book itemsets $\varphi_{j1}, p_{j2}, ..., p_{ji}, b>$ and $\varphi_{j'1}, p_{j'2}, ..., p_{j'i}, b>$ that have $i$-1 common patron attribute values and the common book attribute value $b$. Without loss of generality, assume $p_{j1}$ and $p_{j'1}$ are those that not among the common attribute values, and $p_{j1}'(p_{j'1}) \in N_i$ is equal to or an ancestor of $p_{j1}(p_{j'1})$ in the corresponding patron lattice. The combined $(i+1)$-patron-book itemset is large only if the count corresponding to $p_{j1}'$ in $<p_{j1}, p_{j2}, ..., p_{ji}, b>$ and that corresponding to $p_{j'1}$ in $<p_{j1}, p_{j2}, ..., p_{ji}, b>$ are both no less than $MinSup$. For example, consider the following three
patron lattices (hierarchies) and one book hierarchy as showed in Figure 3-5. Let $l_{21} = \{3ABC, 1ABC, BB\}$, $l_{22} = \{3ABC, 2ADE, BB\}$, $l_{23} = \{1ABC, 2ADE, BB\}, \ldots$ in $L_2$. Suppose we want to join two 2-patron-book itemsets $\{3ABC, 1ABC, BB\}$, $\{3ABC, 2ADE, BB\}$ and $\{1ABC, 2ADE, BB\}$. The combined 3-patron-book itemset is large only if the count corresponding to $<2AD>$ (which is the 2’nd level ancestor of $<2ADE>$) in $\{3ABC, 1ABC, BB\}$ and that corresponding to $<1AB>$ (which is the 2’nd level ancestor of $<1ABC>$) in $\{3ABC, 2ADE, BB\}$ and that corresponding to $<3AB>$ (which is the 2’nd level ancestor of $<3ABC>$) in $\{1ABC, 2ADE, BB\}$ are both no less than $MinSup$.

![Figure 3-5: Example of prune technique](image)

The pseudo-code the MergePrune algorithm is listed in Figure 3-6.
Algorithm MergePrune($g$: a partition of patron attribute set)
/* Assume that the patron attribute set is partitioned into [\{p_1, ..., p_{g1-1}\},\{p_{g1},...,p_{g2-1}\},
\ldots,\{p_{g(r-1)},...,p_k\}],
\text{D is the circulation database and MinSup is the user-specified support threshold */

begin
Merge the concept hierarchies of patron attributes in each group
\{p_{g(i-1)}, ..., p_{gi-1}\} into a lattice $T_i$, 1\leq i \leq r;

for each transaction $t_i = \langle p_{i1}, p_{i2}, ..., p_{ij}, b_{i1}, b_{i2}, ..., b_{im} \rangle \in \text{D}, 1 \leq i \leq n, k \geq 1, m \geq 1$, do begin
Extend $t_i$ by adding all ancestors of \{p_{g(i-1)}, ..., p_{gi-1}\} by referencing the $j$th patron
lattice and $b_{ij}$ by referencing the book hierarchy, 1\leq j \leq r, 1\leq i \leq m, and removing any
duplicates;
Increment the count of every 1-patron itemset $p$ and 1-book itemset $b$ in $t_i$;
Let $p'$ and $b'$ be the level 2 ancestors of $p$ and $b$ in the patron lattice and book hierarchy
respectively. Increment $p.b'$ and $b.p'$;
end;

$L_1(P) = \{c \mid c.count \geq \text{MinSup}, c \text{ is in a node in a patron hierarchy}\}$;
$L_1(B) = \{c \mid c.count \geq \text{MinSup}, c \text{ is in a node in the book hierarchy}\}$;

$C_1 = \{\langle p, b \rangle \mid p \in L_1(P), b \in L_1(B), p.b' \geq \text{MinSup}, \text{ and } b.p \geq \text{MinSup}\}$

for each transaction $t \in \text{D}$ do begin
Delete any item in $t$ that does not appear in either $L_1(P)$ or $L_1(B)$;
end;

for each transaction $t \in \text{D}$ do begin
  for every candidate $p_j$ in $C_1$ that are contained in $t = \langle p_1, ..., p_r, ... \rangle$ do begin
    Increment the count of $p_j$;
    Let $p_j'$ be the level 2 ancestor of $p_j$ in the corresponding patron lattice, 1\leq i \leq r, i\neq j;
    Increment $p_j.p_j'$;
  end;
end;

$L_1 = \{c \mid c \in C_1 \text{ and } c.count \geq \text{MinSup}\}$;
$$k=2;$$

**while** ($L_{k-1} \neq \emptyset$) **do begin**

**for** each itemset $l=<p_{j1}, p_{j2}, \ldots, p_{jk-1}, b>$ in $L_{k-1}$ **do begin**

**for** each itemset $l'=<p_{j'1}, p_{j'2}, \ldots, p_{jk'-1}, b'>$ in $L_{k-1}$ **do begin**

if ($b=b'$) and there exist $k-2$ common patron attributes (say, $p_{j2}, \ldots, p_{jk-1}$) **then**

Let $p_{j1}$ and $p_{j'1}$ be the ancestors of $p_{j1}$ and $p_{j'1}$ in level 2 of patron lattice;

if ($l.p_{j1} \geq \text{MinSup}$) and ($l'.p_{j1} \geq \text{MinSup}$) **then**

Add $<p_{j1}, p_{j1}', p_{j2}, \ldots, p_{jk-1}, b>$ to $C_k$;

**end;**

**end;**

**for** each transaction $t \in D$ **do begin**

**for** every candidate $c=<p_1, p_2, \ldots, p_k, b>$ in $C_k$ that are contained in $t=<p_1, \ldots, p_r, \ldots>$ **do begin**

Increment the count of $c$;

Let $p_i'$ be the level 2 ancestor in the corresponding patron lattice, $k < i \leq r$;

Increment $c.p_i'$;

**end;**

**end;**

$L_k = \{c | c \in C_k \text{ and } c.\text{count} \geq \text{MinSup}\}$;

$k = k + 1$;

**end;**

**end.**

Figure 3-6: Pseudo-code for algorithm MergePrune

### 3.3. Identifying Interesting Rules

A patron-book association rule is said to be valid if it satisfies both minimum support and minimum confidence. However, of all the valid patron-book association rules, some could be related to another in either the patron itemset part (the antecedent) or the book itemset part (the conclusion), and therefore the existence of one such rule could make another not interesting any more. In [Agra95], Agrawal et al. defined an interesting measure that is used to prune from the set of valid rules those uninteresting
rules. In their work, a rule \( R \) is interesting if and only if for every close ancestor \( R' \) of \( R \), the support of \( R \) is more than \( \gamma \) times the expected support derived from \( R' \) or the confidence of \( R \) is more than \( \gamma \) times the expected confidence derived from \( R' \), where \( \gamma \) is a user-specified threshold. An itemset \( Z' \) is said to be an ancestor of another \( Z \) if \( Z' \) can be obtained from \( Z \) by replacing one or more items in \( Z \) with their ancestors and \( Z \) and \( Z' \) have the same number of items. An ancestor \( R':X' \Rightarrow Y' \) of \( R: X \Rightarrow Y \), where \( X' \) is an ancestor of or equal to \( X \) and \( Y' \) is an ancestor of or equal to \( Y \) in their respective taxonomies but \( (X', Y') \neq (X, Y) \), is said to be close if there does not exist any another ancestor \( R'' \) of \( R \) such that \( R' \) is an ancestor of \( R'' \) and \( R'' \) is an ancestor of \( R \). Let \( \text{Sup}(R) \) and \( \text{Conf}(R) \) denote the support and confidence of \( R \) respectively. The expected support of \( R: X \Rightarrow Y \) given the ancestor \( R':X' \Rightarrow Y' \), denoted \( E(\text{Sup}(R)) \), is defined as \( \frac{\text{Sup}(X)}{\text{Sup}(X')} \times \frac{\text{Sup}(Y)}{\text{Sup}(Y')} \times \text{Sup}(R') \). The expected confidence of \( R: X \Rightarrow Y \) given the ancestor \( R':X' \Rightarrow Y' \), denoted \( E(\text{Conf}(R)) \), is defined as \( \frac{\text{Sup}(Y)}{\text{Sup}(Y')} \times \text{Conf}(R') \).

The above observation originally made by Agrawal et al. naturally lead to the development of techniques that prune those less significant rules given a set of ancestor rules. These rules are considered insignificant because they do not convey additional information than their ancestors and are less general. However, we have also noticed the existence of some ancestor rules that are valid simply because some of their descendant rules are valid. In this case, these ancestor rules, though valid, should not be considered interesting. Consider the following rule \( R_1 \): “compute science student” \( \Rightarrow \) “computer books”. If \( R_1 \) is valid, then the rule \( R_2 \): “computer science student” \( \Rightarrow \) “mathematics books” must also be valid because every computer book is also classified as a mathematics book by the library classification scheme. However, \( R_2 \) may not be interesting if most transactions that support \( R_2 \) also
support $R_1$. We see $R_2$ as interesting only when many students who major in computer science also ever checked out some mathematics books that are not computer related. Consider another rule $R_3$: “engineering student” $\Rightarrow$ “computer books”. If $R_1$ is valid, then $R_3$ must also satisfy minimum support. Again, $R_3$ is not interesting if most transactions that support $R_3$ come from those support $R_1$. In contrast, we will view $R_3$ as interesting if many engineering students whose major are not computer science also ever checked out computer books. In other words, an ancestor rule is considered interesting only if it is capable for convey more general information.

Based on the two observations described above, we develop our “interestingness” measure as follows:

**Definition.** (pseudo-interestingness)

Consider a valid patron-book association rule $R: P \Rightarrow B$. Suppose there also exists a set of valid rules $R_1: P \Rightarrow B_1$, $R_2: P \Rightarrow B_2$, …, $R_k: P \Rightarrow B_k$, and another set of valid rules $R_1': P_1 \Rightarrow B$, $R_2': P_2 \Rightarrow B$, …, $R_k': P_k \Rightarrow B$, where $B$ is an ancestor of $B_i$, $1 \leq i \leq k$, and $P$ is an ancestor of $P_i$, $1 \leq i \leq k'$. $R$ is pseudo-interesting if

1. $P \Rightarrow B-(B_1 \cup B_2 \cup \ldots \cup B_k)$ is valid, and
2. $P-(P_1 \cup P_2 \cup \ldots \cup P_k') \Rightarrow B$ is valid.

**Definition.** (interestingness)

A patron-book association rule $R: P \Rightarrow B$ is interesting if it is pseudo-interesting and either of the following two conditions hold:

1. None of $R$’s ancestors are interesting.
2. For each valid close ancestor $R'$ of $R$, the support of $R$ is more than $\gamma$ times the expected support derived from $R'$ or the confidence of $R$ is more than $\gamma$ times the expected confidence derived from $R'$.
The set $\mathcal{R}$ of all valid rules can be seen as a partial order set (POSET) $(\mathcal{R}, \prec)$, where $r_1 \prec r_2$, $r_1, r_2 \in \mathcal{R}$ if $r_1$ is an ancestor of $r_2$. This POSET can be realized as a directed acyclic graph $G=(\mathcal{R}, E)$, where $(r_1, r_2) \in E$ if $r_1$ is a close ancestor of $r_2$. We first traverse $G$ in a bottom-up manner to find the set of pseudo-interesting rules. Then a top-down traversal on these pseudo-interesting rules are followed to identify the set of interesting rules. The pseudo-code of the algorithm is listed in Figure 3-7.
/* Assume that $Sup(u)$ is the support of the valid rule $u$, $Sup(u_p)$ is the support of patron part of $u$

$U$ is an ancestor of $u$ */

begin

for each vertex $u \in V[G]$ do begin

    $Sup(R')=0$, $Sup(R'')=0$, $Sup(P'')=0$;

    for each $v \in Adj[u]$ do begin

        if $B$ is an ancestor of $B$, then $Sup(R')=Sup(R')+Sup(v)$;
        if $P$ is an ancestor of $P$, then $Sup(R'')=Sup(R'')+Sup(v)$, $Sup(P'')=Sup(P'')+Sup(v_p)$;

    end;

    if $(Sup(u)-Sup(R') \times Sup(u_p)) > MinSup$ and $\gamma \times \frac{Sup(u)-Sup(R')}{Sup(u_p)} > MinConf$ and
               $Sup(u)-Sup(R'') > MinSup$ and $\gamma \times \frac{Sup(u)-Sup(R'')}{Sup(u_p)} > MinConf$ then

        pseudo-interesting = pseudo-interesting $\cup$ $u$;

    end;

Let each vertex $u$ in $V[G] \in$ pseudo-interesting;

for each vertex $u \in V[G]$ do begin

    if $u$ is no ancestor then

        interesting = interesting $\cup$ $u$; next;

    if $(\gamma \times Sup(u) > Sup(U)$ and $\gamma \times Conf(u) > Sup(U)$) then

        interesting = interesting $\cup$ $u$;

    end;

return interesting;

end.

Figure 3-7: Pseudo-code for identifying interesting rules
Chapter 4 . Evaluation Plan

In this chapter, we first present how to generate synthetic data for evaluate the performance of the three patron-book itemset generation algorithms proposed in chapter 3. Then, we show the results of experiments we have conducted to evaluate the performance of the algorithms.

4.1. Generation of Synthetic Data

Each transaction is composed of a set of patron attributes and a set of books checked out by the patron. We first create a book hierarchy and \( k \) patron attribute hierarchies. Each item in the hierarchy is associated with a weight that indicates the likelihood it will be picked up. We then make up one potentially frequent book types pool and one patron-book types pool. Each book set in the book types pool represents the set of books often checked out together. Each patron-book type in the patron-book types pool represent that some kinds of patron attributes often checked out some kind of book. Finally, we pick book types to each transaction and assign patron attribute types.

To create one book hierarchy and the potentially frequent book types pool, we take the parameters shown in Table 4-1.

Table 4-1: Book Parameters for Synthetic Data Generation

<table>
<thead>
<tr>
<th>( BP )</th>
<th>Average size of the maximal potentially frequent book type</th>
</tr>
</thead>
<tbody>
<tr>
<td>( BI )</td>
<td>Number of maximal potentially frequent book types</td>
</tr>
<tr>
<td>( BN )</td>
<td>Number of book items</td>
</tr>
<tr>
<td>( BF )</td>
<td>Book fanout</td>
</tr>
</tbody>
</table>
For each internal node, determine the number of children by following Poisson distribution with mean $\mu$ equal to fanout $BF$. We assign children to the root, then to the node at the next level until we run out all book items $BN$. Each book type in the hierarchy has a weight associated with it. We randomly assign a weight to each leaf node so that the sum of the weight for all leaf nodes is 1. The weight of each interior node is the sum of the weights of its children. The weights of all nodes in book hierarchy are finally normalized so that the weight sum for all nodes is 1. Until now, we have generated one book hierarchy so as to create potentially frequent book types pool.

For each potentially frequent book set, we first determine its size by following Poisson distribution with mean $\mu$ equal to $BP$. Then we pick each book type in the set by tossing a $BN$-sided weighted coin, where the weight of each side is the probability of choosing the associated book type. Each book set in potentially frequent book type pool has a weight associated with it. The weight is a multiplication of the weights of all constituent book types of all the. Again, the weights in the book types pool are normalized so that, the sum of the weights is 1.

To create $PK$ patron hierarchies and the maximal potentially frequent patron-book types, we take below parameters as in Table 4-2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PI$</td>
<td>Number of maximal potentially frequent patron-book types</td>
</tr>
<tr>
<td>$PN$</td>
<td>Number of total patron attribute types</td>
</tr>
<tr>
<td>$PK$</td>
<td>Number of patron attributes</td>
</tr>
<tr>
<td>$PF$</td>
<td>Fanout</td>
</tr>
</tbody>
</table>

Table 4-2: Patron Parameters for Synthetic Data Generation

In the following we present how to generate $PK$ patron attribute hierarchies and a
pool of potentially frequent patron-book types. It first builds $PK$ patron attribute hierarchies with total patron attribute values being $PN$. We model the hierarchy as a forest. For each internal node, determine the number of children by following Poisson distribution with mean $\mu$ equal to fanout $PF$. We assign children to the root, then to the node at the next level until we run out all patron types $PN/PK$. There are $PK$ patron hierarchies being build. Each patron attribute node in the patron hierarchy has a weight associated with it. We pick a random number for each leaf node so that the sum of the weights for all leaf nodes is 1. The weight of each interior node is the sum of the weights of its children. The weights of all nodes in each patron attribute hierarchy are normalized so that the weight sum for all nodes is 1.

Next, we generate the maximal potentially frequent patron-book types. Each potentially frequent patron-book type in the patron-book types pool has a weight associated with it. Each patron-book type of the form $<p_1, p_2, \ldots, p_k, b_i>$ is generated, where $p_k$ is one patron attribute picked out from one patron attribute of the $k$th patron attribute hierarchy and one book type picked out from one of the book types set in the second level of the book hierarchy. The weight of a patron-book type is the multiplication of the weights of constituent patron attributes and book type. We normalized the weights with the same book type in the maximal potentially frequent patron-book types pool; the sum of the weights with the same book type is 1.

We create transactions based on the following parameters as listed in Table 4-3.

<table>
<thead>
<tr>
<th>$D$</th>
<th>Number of transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Average size of the transactions</td>
</tr>
</tbody>
</table>

Table 4-3: Transaction Parameters for Synthetic Data Generation
After preparing the above two frequent pools, we follow Poisson distribution with mean $\mu$ equal to $T$ to determine the size of the next transaction. The book of the transaction is determined by first picking a book set in the frequent book types pool based on the respective weight. Each non-leaf book type $b$ in a transaction is specialized to leaf book type tossing an $m$-sided weighted coin, where $m$ is the number of children of the non-leaf book type, and the weight is associated with the children, to decide which branch to follow, till we reach a leaf book type.

We determine which patron attribute type in the patron-book types pool to put into a transaction by the following. We search for the biggest number of book type in the second level of the book hierarchy for each transaction and toss $n$-sided weighted coin, where $n$ is the number of patron type with the same book type, and the weight correspond to the weight of patron-book type. Each non-leaf patron attribute type $p$ in a transaction is specialized to leaf patron attribute type. We toss an $x$-sided weighted coin, where $x$ is the number of children, and the weight is associated with the children, to decide which branch to follow, till we reach a leaf patron attribute value.

4.2. Relative Performance of Algorithms
The parameters and their settings are shown in Table 4-4. We performed several experiments on a Celeron 400 personal computer with 56 MB of main memory, and running Microsoft Windows 98.
In Figure 4-1, we first examined the impact of attribute merge on performance with the “MergePrune” algorithm. In this experiment, we applied the prune technique to reduce the size of the candidate set. Six patron attributes were merged in different ways, namely (1,1,1,1,1,1), (2,2,2), (3,3) in turn as the horizontal axis. The (1,1,1,1,1,1) merged attribute represents “k-pass”. The (2,2,2) merged attribute represents “MergePrune” merged into three groups. The (3,3) merged attribute represents “MergePrune” merged into two groups. We observed the (3,3) merged attribute performed best, because it only scanned dataset for three times. The

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$MinSup$</td>
<td>$5%$</td>
</tr>
<tr>
<td>$BP$</td>
<td>Average size of the maximal potentially frequent book item set</td>
</tr>
<tr>
<td>$BI$</td>
<td>Number of maximal potentially frequent book types</td>
</tr>
<tr>
<td>$BN$</td>
<td>Number of book items</td>
</tr>
<tr>
<td>$BF$</td>
<td>Book fanout</td>
</tr>
<tr>
<td>$PI$</td>
<td>Number of maximal potentially frequent patron book types</td>
</tr>
<tr>
<td>$PN$</td>
<td>Number of total patron attribute values</td>
</tr>
<tr>
<td>$PK$</td>
<td>Patron attributes having $k$ concept hierarchies</td>
</tr>
<tr>
<td>$PF$</td>
<td>Patron Fanout</td>
</tr>
<tr>
<td>$D$</td>
<td>Number of transactions</td>
</tr>
<tr>
<td>$T$</td>
<td>Average size of the transactions</td>
</tr>
</tbody>
</table>
performance of basic algorithm is not reported here because it introduced a huge candidate set and was unable to run on our experimental platform.

In the following experiment, we intend to compare the effectiveness of “prune” technique. We varied the number of transactions from 2,000 to 10 thousands. Figure 4-2 and 4-3 show the execution time and the size of candidate set with and without “prune” technique respectively. We observed that there were huge useless candidates generated without the prune technique.
In Figure 4-4, we investigated how the “MergePrune” algorithm scaled up with the number of transactions. We tested the “MergePrune” algorithm in the different
scalability by varying the number of transactions from 10,000 to 70,000 in turn as the horizontal axis. We observed the execution time increased linearly when the number of transactions increased. The smaller minimum support brought more candidates to be tested.

In Figure 4-4, we investigated how the number of patron attributes impacts. We tested the “MergePrune” algorithm by varying the number of patron attributes from 2 to 6 in turn as the horizontal axis. We observed the execution time increased and was in a geometric progression when the number of patron attributes increased. The fewer patron attributes introduce fewer times of database scanning and creating much smaller size of candidates.
Figure 4-5: The impact of number of patron attributes
Chapter 5 . Empirical Results

We have applied our proposed approach to the NSYSU library circulation data. This chapter reports our experiences in such an application.

NSYSU is a mid-sized university in Taiwan with around 6000 students enrolled. To date, its library houses over 582,000 volumes of books and bound periodicals; this number is still increasing at the rate of 10% per year. The OPAC system used at NSYSU library is from INNOVATIVE. Unfortunately, this OPAC system does not record historical data. To make our study sensible, we have accumulated the circulation data for about half a year to make our circulation database involve both present and historical circulation data. At the time of the study, our circulation databases consists of 4615 transactions, each of which record the set of books checked out by a separate student patron in the last half year[1].

5.1. Identifying relevant patron attributes

We first apply the approach presented in chapter 3 to identify attributes relevant to the types of books checked out. The OPAC system at NSYSU library records several attributes about patrons, including sex, address, degree, program, work-unit, and status. To focus our attention on student patrons, we chose four attributes, namely sex, address, degree, and program. Except for sex, which has only two values: ‘M’ and ‘F’, every other attribute has a concept hierarchy. To apply the classification techniques, the domain of each attribute is fixed at the second level of the associated concept hierarchy, which is shown in Figure 5-1. With respect to the book hierarchy, we selectively chose book types from the Chinese classification scheme [賴永祥 89] and

[1] As students compose most patrons at NSYSU library, we focus our attention on student patrons only. Other types of patrons, such as faculty, staff, and off-campus patrons, are not considered.
form a book hierarchy of 3 levels. The resultant book hierarchy is shown in Appendix 2. Again, values in the second level of the book hierarchy are chosen for classification purpose. The classification algorithm we adopted was C4.5.

sex:M, F
address:county names
degree:PHD,Master,Bachelor.
program:Science,Engineering,Social,Liberal,Ocean,Management..

Figure 5-1: Candidate attribute values

Figure 5-2 shows part of the decision tree we obtained by running C4.5 algorithm. In the decision tree, both sex and address do not appear in any turning point, and we conclude that they both have a low correlation to the types of books checked out. The other two attributes, namely college and status, serve as patron attributes in identifying generalized patron-book rules.
5.2. Generating patron-book rules

We have applied our circulation database to the three algorithms proposed in Chapter 3. The $MinSup$ was set to 1%; the $MinConf$ was set to 30%. Their relative performance is shown in Figure 5-3. The algorithm Basic turns out to be the best due to the relative small amount of transactions in the experimental circulation database and the fewer number of patron attributes (only 2). However, MergePrune still performed close to the Basic in this case.
5.3. Pruning uninteresting rules

The large patron-book patterns are then used for generating patron-book association rules. The $\lambda$ was set to 1. There are totally more than 300 rules. After applying our approach for pruning uninteresting rules, 117 rules were left out. These interesting rules are listed in Appendix 1. We also apply Agrawal interesting measure for pruning uninteresting rules, only 203 rules were left out. Our empirical experience showed that more than one third of the rules are pruned by our proposed interestingness criterion, which greatly reduce the effort incurred by the librarians to examine rules. Their relative rules remain is shown in Figure 5-4.
5.4. Effectiveness of the patron-book rules

To evaluate the effectiveness of the patron-book rules, we have implemented a web-based system that recommends new books to patrons based on personal background, namely the degree sought and the program studied. A patron interacts with the system by providing his degree and program information. The system then identifies from the rules bases those whose antecedent (i.e., patron types) matches the provided background information. The conclusion parts (i.e., book types) are then used to identify books of interest from the new book database. To evaluate the effectiveness of these rules, 70% of the recommended books are from the set of interesting books identified by our approach, while the other 30% are from books in which the system thinks the patron is not interested for comparison purpose. The minimum support and minimum confidence thresholds are set at 1% and 30% respectively. Patrons are asked to mark those books they feel interested. Screenshots
of the system are shown in Figure 5-5 and Figure 5-6.

![User filled in user's program, status, and acceptable book numbers](image-url)

Figure 5-5: User filled in user's program, status, and acceptable book numbers
For a given user, let the set of books recommended by the system be $S_{Recom}$. The set $S_{Recom}$ is made up of two exclusive sets $S_{Like}$ and $S_{Dislike}$, representing the 70% and 30% of books described above respectively. Besides, within the set $S_{Recom}$, only a subset $U_{Like}$ is chosen by the user. Their relationships are shown in Figure 5-7 for easy visualization:
We define the following three measures to evaluate the effectiveness of a recommendation given to a user, namely precision, recall and accuracy.

- **Precision (Pr):** \( \frac{|S_{Like} \cap U_{Like}|}{|U_{Like}|} \)
- **Recall (Rec):** \( \frac{|S_{Like} \cap U_{Like}|}{|S_{Like}|} \)
- **Classification accuracy (Acc):** \( \frac{|(S_{Like} \cap U_{Like}) \cup (S_{Dislike} - U_{Like})|}{|S_{Recom}|} \)

Due to time limitation, we have collected only 47 patrons’ response about new book recommendation. Table 5-1 shows the preliminary result. The average precision is 0.7591, the average recall is 0.1022, and the accuracy is 0.3464. As can be seen from the table, most users marked only small amount of books recommended, making both the precision and accuracy low. The reason for such phenomenon is that users tend to choose those books they like most. Books that do not top the list of users’ likes
were not marked. Therefore, the measures precision and accuracy may not retain the meaning they are supposed to. However, from the average recall, we can conclude that the circulation data actually include some hidden information that can be dug out by our proposed approach.

<table>
<thead>
<tr>
<th>$S_{_Like}$</th>
<th>$S_{_Dislike}$</th>
<th>$S_{_Like \cap U_{_Like}}$</th>
<th>$S_{_Dislike - U_{_Like}}$</th>
<th>Pr</th>
<th>Rec</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>18</td>
<td>17</td>
<td>0</td>
<td>0.7500</td>
<td>0.1429</td>
<td>0.3667</td>
</tr>
<tr>
<td>56</td>
<td>24</td>
<td>21</td>
<td>0</td>
<td>0.7273</td>
<td>0.1429</td>
<td>0.3625</td>
</tr>
<tr>
<td>21</td>
<td>9</td>
<td>8</td>
<td>0</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2667</td>
</tr>
<tr>
<td>140</td>
<td>60</td>
<td>54</td>
<td>0</td>
<td>0.5000</td>
<td>0.0429</td>
<td>0.3000</td>
</tr>
<tr>
<td>70</td>
<td>30</td>
<td>28</td>
<td>0</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2800</td>
</tr>
<tr>
<td>140</td>
<td>60</td>
<td>60</td>
<td>0</td>
<td>1.0000</td>
<td>0.0071</td>
<td>0.3050</td>
</tr>
<tr>
<td>140</td>
<td>60</td>
<td>58</td>
<td>0</td>
<td>1.0000</td>
<td>0.0214</td>
<td>0.3050</td>
</tr>
<tr>
<td>140</td>
<td>60</td>
<td>60</td>
<td>0</td>
<td>1.0000</td>
<td>0.0429</td>
<td>0.3300</td>
</tr>
<tr>
<td>28</td>
<td>12</td>
<td>12</td>
<td>0</td>
<td>1.0000</td>
<td>0.1786</td>
<td>0.4250</td>
</tr>
<tr>
<td>21</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>1.0000</td>
<td>0.1429</td>
<td>0.4000</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>0.5000</td>
<td>0.1429</td>
<td>0.3000</td>
</tr>
<tr>
<td>140</td>
<td>60</td>
<td>60</td>
<td>0</td>
<td>1.0000</td>
<td>0.0071</td>
<td>0.3050</td>
</tr>
<tr>
<td>140</td>
<td>60</td>
<td>60</td>
<td>0</td>
<td>1.0000</td>
<td>0.0214</td>
<td>0.3150</td>
</tr>
<tr>
<td>140</td>
<td>60</td>
<td>58</td>
<td>0</td>
<td>1.0000</td>
<td>0.0214</td>
<td>0.3050</td>
</tr>
<tr>
<td>140</td>
<td>60</td>
<td>60</td>
<td>0</td>
<td>1.0000</td>
<td>0.0571</td>
<td>0.3400</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0.5000</td>
<td>0.1429</td>
<td>0.3000</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>13</td>
<td>2</td>
<td>0.7647</td>
<td>0.9286</td>
<td>0.7500</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>0.3333</td>
<td>0.1714</td>
<td>0.2500</td>
</tr>
<tr>
<td>140</td>
<td>60</td>
<td>55</td>
<td>0</td>
<td>0.4444</td>
<td>0.0286</td>
<td>0.2950</td>
</tr>
</tbody>
</table>
Chapter 6. Literature Review

Researchers in library science community has long proposed the idea of personalizing the information to be received by patrons, called selective dissemination of information (SDI). With the advent of World Wide Web, many approaches for implementing recommendation systems have been proposed. In this chapter, we will first survey SDI and then review approaches for implementing recommendation systems. Finally, we will elaborate on data mining techniques.

6.1. SDI (Selective Dissemination of Information)

Many information suppliers like libraries, publishers, and databanks provide the user with an increasing volume of digital information. With the increasing document creating rate, information overloading is becoming an important issue. In a dynamic information society, it is hard for users, equipped with only conventional search capability, to keep up with the fast pace of information generation. Rather than passively requiring users going through the large amount of incoming information, so called pull-based information delivery, many researchers and practitioners advocate the idea of push-based information delivery, by which information is selectively delivered to the interested users. In tradition, libraries and databanks provide such kind of information filtering service, called Selective Dissemination of Information (SDI) [SALT68]. A user submits his interest profile that describes the document class of interest. The user will then passively receive the new documents matching this profile.

Yan et al. made the following observations for an SDI service [Yan94]:

- SDI service should allow a various class of queries as profiles.
- SDI service should be able to evaluate profiles continuously and the user should be notified as soon as a relevant document arrives, not periodically.
• SDI service should be able to handle with a very large number of profiles and a large volume of new documents.
• SDI service should be able to efficiently and reliably distribute the documents to the users.

A number of approaches for realizing the above objectives have been described in [Yan94]. However, SDI service is a passive service because it needs a patron to provide his personal profiles to the library. For those who are unwilling or unable to provide their interest profiles, SDI service would be of little use to them.

6.2. Recommendation Approaches

A large number of approaches have been proposed to identify users’ personal interest patterns. These approaches are intended to be used by recommendation systems to recommend information of various types (e.g. books, music albums, news articles, web sites, and emails) that meet individual interests. These approaches can be classified into two non-exclusive categories, namely content-based recommendation and collaborative recommendation [Smyt00].

Content-based recommendation

A content-based filtering approach characterizes recommendable items by a set of content features and represents users’ interest profiles by a similar feature set. Then, the relevance of a given content item and the user’s profile is measured as the similarity of this recommendable item to the user’s profile. Finally, content-based approaches select recommendable items that have a high degree of similarity to the user’s interest profile.

The book recommending prototype system LIBRA [Raym98] is one kind of
content-based recommendation systems. It downloads a list of pages of book-description URLs of broadly relevant titles as book content and parses the given book by applying a simple pattern-based information-extraction system to extract data about each title. Next, the system learns a user profile by asking the user to select from a set of training books and provide a 1-10 rating for each of the selected book. A bag-of-words simple Bayesian text classifier is employed by LIBRA to figure out the posterior category probabilities for each book. Based on posterior probability of a positive categorization, the top-scoring recommendations are presented to the user as the preferred ranking.

However, as pointed out in [Shar95], content-based approaches exhibit the following limitations:

- In content-based approach, the items must be of some machine parsable form (e.g. text).
- There are no surprising recommendations found by Content-based filtering techniques.
- Those items based on quality, style or point-of-view can not be filtered by content-based filtering methods.

**Collaborative recommendation**

Collaborative filtering techniques rely on the experiences of users with similar interests to recommend items. Specifically, the collaborative filtering techniques look for relevance among users by observing their ratings assigned to items in a particular training set. The nearest–neighbor users are those that exhibit the strongest relevance to the target user. These users then act as “recommendation partners” for the target user, and collaborative approach recommend the target user items that appear in the
profiles of these recommendation partners (but not in the target user’s profile).

Content-based approaches exploit one kind of information about the contents of each item the target user accesses, and collaborative approaches exploit a second kind of information of what others in the same community thought of each item. It is explicit that effective recommendation services require aspects of both approaches.

However, we consider that content-based and collaborative are not suitable for new book recommendation in library due to the following reasons.

- Both recommendation approaches require each user have enough transactions for data analysis. However, each patron in the library usually borrowed only a few books. Therefore, it is difficult to determine a patron’s interest with a small number of transactions.

- The collaborative approach is not suitable for recommending new books because these techniques can only recommend books already read by some other patrons, whereas new books are those that have not been checked out by any patrons. Besides, there is insufficient overlap among books checked out by patrons. Our observations from the library circulation data at NSYSU showed that even the most popular book is not checked out by more than fifteen patrons per year. Therefore, we conclude that it is not appropriate to recommend new books based on each individual’s interests. Instead, we will try to identify interests of a user group.

### 6.3. Data Mining Methodology

Data mining is the process of extracting previously unknown, valid, and actionable patterns, knowledge, or high-level information from large databases and then using
the information to make crucial business decision. It has been of great interest to academia and industry. In the following, we review three important data mining techniques that are related to our work, namely clustering, classification, and association rules.

6.4. Clustering
Over the past years, clustering analysis techniques have been widely applied to many application domains such as customer retention, target marketing, document clustering, and so on. Clustering analysis is a process that a set of objects is segmented into several clusters in which each object in the same cluster is in some way similar and is different from the objects of other clusters. The main goal is to find out clusters present in the data. Noted clustering methods include K-means [Mac67], PAM (Partitioning Around Medoids) [Kauf90], CLARA [Kauf90], and CLARANs [Ng94].

One can apply some clustering method to partition patrons into a set of groups according to the types of books checked out. Each group comprises patrons that checked out similar books. However, as new books are those that are never checked out, we still do not have clues on how to dispatch new books user groups. We made the same argument for the inapplicability of collaborative approach.

6.5. Classification
Classification analysis techniques have many direct application domains such as medical diagnosis, credit approval, cross selling, fraud detection, and text
categorization. Its main goal is to build a model of the classification attribute based upon the other attributes from a set of instances (called training set). The training set contains a plurality of records. Each instance in the training set is tagged with a class label. Once the model is build, the model is used in order to predict the value of the class label for the next new instances. ID3 [Quin86] and its descendants (e.g., C4.5) [Quin93], CN2 [Clar89], and backpropagation neural network [Rich91] are among the popular methods for conducting classification.

ID3 decision tree induction technique is one commonly used approach. The idea behind the ID3 algorithm is to use an iterative method to build up decision trees until each partition contains most examples from a particular class. ID3 looks for the attribute that best separates the given training instances. If the attribute perfectly partitions the training instances then ID3 stops. Otherwise, it iteratively operates on the \( n \) (assuming that the selected attribute values are partitioned into \( n \) subsets) partitioned subsets to get their best attribute. In each iterative pass, it picks the best attribute and never looks back to reconsider earlier choices. In order to measure how well an attribute separates training instances into targeted classes, they borrow an idea from information theory called entropy. Entropy measures the amount of information in an attribute.

Classification techniques are not suitable to our problem because our goal is not to predict the patron (book) group to which a particular patron (book) belongs.

### 6.6. Association Rules

Agrawal, Imielinski, and Swami introduced the problem of mining association rules in a transaction database [Agra93], where each transaction is represented as a set of
items. An association rule is an expression of the form $X \Rightarrow Y$, where $X$ and $Y$ are sets of items. An example association rule is: “30% of the transactions that contain beer also contain diapers; 2% of all transactions contain both beers and diapers”. The former percentage (30%) is called the confidence of the rule, while the later percentage (2%) is named as the support of the rule. The problem is to find all association rules that satisfy user-specified minimum support and minimum confidence constraints.

Many algorithms for generating association rules have been introduced in the literature (e.g., see [Agra93], [Agra94], [Park95], and [Brin97]). These methods, though different in their subtle details, are based on a common approach as described below. This approach is composed of two steps. Firstly, find all sets of items (called itemsets) that have transaction support above the user-specified minimum support. Itemsets with minimum support are called large itemsets. Secondly, use the large itemsets to generate the desired rules. It was found that the first step dominates the entire process in terms of execution time. An algorithm called Apriori has been developed to find all large itemsets. The Apriori algorithm first counts item occurrences to determine the large 1-itemsets, i.e., the itemsets that contain only single elements. A subsequent pass, say pass $k$, consists of two steps. Firstly, the large itemsets $L_{k-1}$ found in the $(k-1)$th pass are used to generate the candidate itemsets $C_k$. Secondly, the database is scanned and the support of candidates in $C_k$ is counted to generate large itemsets $L_k$ This procedure is repeated until $L_k=\emptyset$.

Concept hierarchies on attributes in the context of data mining were first introduced by Han et al. [Han92]. Concept hierarchies represent necessary background knowledge that controls the generalization process.
For example, the following shows two example concept hierarchies on student’s major and student’s origin respectively:

\{biology, chemistry, computing, ..., physics\} \subset science

\{literature, music, ..., painting\} \subset art

\{science, art\} \subset any

\{Burnaby, ..., Vancouver, Victoria\} \subset British Columbia

\{Calgary, ..., Edmonton, Lethbridge\} \subset Alberta

\{Hamilton, Toronto, ..., Waterloo\} \subset Ontario

\{British Columbia, ..., Alberta, Ontario\} \subset domestic

\{Bombay, ..., New Delhi\} \subset India

\{Beijing, Nanjing, ..., Shanghai\} \subset China

\{China, India, Germany, ..., Switzerland\} \subset foreign

\{domestic, foreign\} \subset any

Using a set of concept hierarchies, Han et al. developed several tree-climbing algorithms to characterize a particular set of records [Han92]. An example application is to find the characteristics of graduate students with good grades.

Han et al. also introduced the problem of mining multiple level association rules with each attribute being associated with a concept hierarchy [Han95]. To reduce the explosive effort in mining rules at any levels in hierarchies, they require a threshold to be specified on the number of nodes in a level of interest. They use a hierarchy-information encoded transaction table, instead of the original transaction table, in iterative data mining.

The problem of finding associations between items at any level of the taxonomies, or called generalized association rules, was also first proposed by
Agrawal and Srikant [Agra95]. For example, given a taxonomy that says that Foremost is-a 2% is-a milk, we may infer the following rule: “75% of people who buy bread also buy 2% milk.” And the basic solution to this problem is to add all ancestors of each item in a transaction to the transaction, and then run any of the algorithms for mining association rules on these extended transactions.

Agrawal et al. introduced the problem of mining association rules in large relational tables containing both quantitative and categorical attributes [Agra96]. An example of such an association might be “10% of married people between age 50 and 60 have at least 2 cars”. Relational tables in most business and scientific domains have richer attribute types. Attributes can be quantitative (e.g. age, income) or categorical (e.g. zip code, make of car).

Aggarwal et al. introduced the problem of finding online algorithms for a special case of the quantitative association rule problem [Agga98]. They refer to this problem as the profile association rule problem. A profiles association rules is one in which the left hand side of the rule consists of customer profile information, such as age, salary, years of education, and martial status. The right hand side of the rules consists of customer behavior information, such as buying milk, diaper, beer, etc. The rules need to satisfy a given support and confidence requirement as in conventional association rules. The objective of mining profiles association rules is to identify customer profiles for target marketing, where the support indicates the size of the potential target market and the confidence gives the effectiveness of the rule. Thus, profile association rules are related to quantitative association rules, in that they would like to relate the quantitative profile attributes in the antecedent to the sparse behavioral attributes in the consequent. They built a multidimensional index called S-tree on the
profile attributes of the data, so as to be able to answer queries within a specified query box easily. Each node in S-tree carries information about the behavioral attributes for the data points enclosed.

In our problem, we need to find out which patrons like to read which type of books. Note that book types are nodes in book taxonomy. So we need to consider hierarchy information in both sides of the rules. We have chosen to base our algorithm on generalized association rules techniques in the domain of new book recommendation because:

- Association is searching for association relationship between two sets of items. The strongness of an association rule is determined by support threshold and confidence threshold. The two itemsets respectively are patrons’ attribute domain and book classes.

- The values in patrons’ attributes and book classes can both be related to each other in a hierarchical manner. The antecedent and consequence of an association rule can appear in the high level or low level of the concept hierarchies by utilizing generalized association rules analysis techniques.
Chapter 7. Conclusion

7.1. Summary
We examined in this thesis the issue of mining generalized association rules for recommending new books in libraries. Novel algorithms for finding generalized association rules between patron attributes and book classes were proposed and their relative performance were compared. We also defined a measure to determine interestingness of generalized association rules and developed algorithms to prune those uninteresting ones. Finally, we implemented a book recommendation system prototype that incorporates the proposed algorithms to provide suggestions on who a given set of new books should be promoted to.

7.2. Contributions
Several contributions of the research have been made by this thesis:

- Our object is to find association rules between patrons and books. Number of patron roots experiment demonstrates that increasing the number of patron roots has a large load on generating large itemsets. We use decision tree induction technique to identify relevant patron attributes. The heuristic succeeds in checking out patron attributes with high correlation and finding a set of independent patron attributes.

- We showed how the best features of the two proposed heuristics can be joined into a hybrid algorithm, called “MergePrune”. Scale-up experiment demonstrated that “MergePrune” scales linearly with the number of transactions.

- User is persecuted by many uninteresting or redundant rules generated along with the interesting rules. We draw a new interest measure to prune uninteresting rules by utilizing the taxonomy information.
Bibliography


Appendix 1

Interesting Rules

Social ==> Sociology  sup= 0.021  conf= 0.438
Social ==> Economics  sup= 0.021  conf= 0.438
Social ==> Political Science  sup= 0.025  conf= 0.521
Freshman-Liberal ==> Special literature  sup= 0.012  conf= 0.545
Freshman-Science ==> Natural Sciences  sup= 0.010  conf= 0.323
Freshman-Science ==> Novel  sup= 0.013  conf= 0.419
Freshman-Engineering ==> Arithmetic  sup= 0.017  conf= 0.425
Freshman-Engineering ==> Applied Sciences  sup= 0.014  conf= 0.350
Freshman-Engineering ==> Novel  sup= 0.012  conf= 0.300
Freshman-Mechanical Engineering ==> Language and Literature  sup= 0.011  conf= 0.611
Freshman-Management ==> Mathematics  sup= 0.017  conf= 0.327
Freshman-Management ==> Business Management  sup= 0.022  conf= 0.423
Freshman-Management ==> Novel  sup= 0.016  conf= 0.308
Freshman-Business ==> Language and Literature  sup= 0.010  conf= 0.588
Liberal ==> Special literature  sup= 0.062  conf= 0.425
Sophomore-Liberal ==> Special literature  sup= 0.010  conf= 0.435
Sophomore-Science ==> Natural Sciences  sup= 0.011  conf= 0.440
Sophomore-Engineering ==> Arithmetic  sup= 0.010  conf= 0.345
Sophomore-Engineering ==> Engineering  sup= 0.010  conf= 0.345
Sophomore-Management ==> Business  sup= 0.014  conf= 0.438
Sophomore-Management ==> Special literature  sup= 0.011  conf= 0.344
Science ==> Language and Literature  sup= 0.063  conf= 0.401
Junior-Liberal ==> Chinese Literature  sup= 0.010  conf= 0.435
Junior-Engineering ==> Arithmetic  sup= 0.012  conf= 0.387
Junior-Engineering ==> Electrical Engineering  sup= 0.010  conf= 0.323
Junior-Management ==> Business Management  sup= 0.024  conf= 0.533
Junior-Management ==> Finance  sup= 0.014  conf= 0.311
Junior-Management ==> Language and Literature  sup= 0.018  conf= 0.400
Junior-Business ==> Business Management  sup= 0.012  conf= 0.750
Engineering ==> Engineering  sup= 0.106  conf= 0.417
Senior-Liberal ==> Statistics  sup= 0.011  conf= 0.344
Senior-Liberal ==> Chinese Literature  sup= 0.012  conf= 0.375
Senior-Liberal ==> Special literature  sup= 0.012  conf= 0.375
Senior-Science ==> Natural Sciences  sup= 0.012  conf= 0.444
Senior-Engineering ==> Arithmetic  sup= 0.013  conf= 0.371
Senior-Engineering ==> Engineering  sup= 0.017  conf= 0.486
Senior-Management ==> Arithmetic  sup= 0.018  conf= 0.346
Senior-Management ==> Business Management  sup= 0.023  conf= 0.442
Senior-Management ==> Language and Literature  sup= 0.019  conf= 0.365
Senior-Business ==> Business Management  sup= 0.010  conf= 0.588
Management ==> Business Management  sup= 0.143  conf= 0.453
Management ==> Social Sciences  sup= 0.117  conf= 0.370
Management ==> Language and Literature  sup= 0.109  conf= 0.345
Bachelor ==> Applied Sciences  sup= 0.191  conf= 0.333
Bachelor ==> Special literature  sup= 0.181  conf= 0.316
Bachelor-Liberal ==> Chinese Literature  sup= 0.040  conf= 0.404
Bachelor-Liberal ==> Novel  sup= 0.030  conf= 0.303
Bachelor-Chinese Literature ==> Chinese Philosophy  sup= 0.017  conf= 0.405
Bachelor-Chinese Literature ==> Poem theory  sup= 0.016  conf= 0.381
Bachelor-Chinese Literature ==> Classical poetry theory  sup= 0.018  conf= 0.429
Bachelor-Chinese Literature ==> Novel  sup= 0.019  conf= 0.452
Bachelor-Foreign Language ==> Special literature  sup= 0.014  conf= 0.438
Bachelor-Foreign Language ==> Western literatures  sup= 0.014  conf= 0.438
Bachelor-Science ==> Natural Sciences  sup= 0.044  conf= 0.407
Bachelor-Science ==> Special literature  sup= 0.034  conf= 0.315
Bachelor-Biologic Science and Life Science ==> Biology  sup= 0.013  conf= 0.433
Bachelor-Biologic Science and Life Science ==> Special literature  sup= 0.010  conf= 0.333
Bachelor-Chemistry ==> Language and Literature  sup= 0.012  conf= 0.545
Bachelor-Physics ==> Mathematics  sup= 0.010  conf= 0.385
Bachelor-Physics ==> Physics  sup= 0.012  conf= 0.462
Bachelor-Physics ==> Language and Literature  sup= 0.012  conf= 0.462
Bachelor-Applied Mathematics ==> Mathematics  sup= 0.010  conf= 0.345
Bachelor-Applied Mathematics ==> Language and Literature  sup= 0.010  conf= 0.345
Bachelor-Engineering ==> Arithmetic  sup= 0.052  conf= 0.388
Bachelor-Engineering ==> Engineering  sup= 0.049  conf= 0.366
Bachelor-Engineering ==> Language and Literature  sup= 0.045  conf= 0.336
Bachelor-Electrical Engineering ==> Arithmetic  sup= 0.021  conf= 0.362
Bachelor-Electrical Engineering ==> Electrical Engineering \( sup = 0.018 \) \( conf = 0.310 \)
Bachelor-Mechanical Engineering ==> Arithmetic \( sup = 0.023 \) \( conf = 0.371 \)
Bachelor-Mechanical Engineering ==> Engineering \( sup = 0.027 \) \( conf = 0.435 \)
Bachelor-Mechanical Engineering ==> Language and Literature \( sup = 0.021 \) \( conf = 0.339 \)
Bachelor-Management ==> Arithmetic \( sup = 0.055 \) \( conf = 0.304 \)
Bachelor-Management ==> Business Management \( sup = 0.078 \) \( conf = 0.431 \)
Bachelor-Management ==> Special literature \( sup = 0.060 \) \( conf = 0.331 \)
Bachelor-Business ==> Business Management \( sup = 0.035 \) \( conf = 0.522 \)
Bachelor-Business ==> Special literature \( sup = 0.027 \) \( conf = 0.403 \)
Bachelor-Informance ==> Arithmetic \( sup = 0.017 \) \( conf = 0.567 \)
Bachelor-Informance ==> Business Management \( sup = 0.011 \) \( conf = 0.367 \)
Bachelor-Informance ==> Special literature \( sup = 0.011 \) \( conf = 0.367 \)
Bachelor-Finance ==> Business Management \( sup = 0.011 \) \( conf = 0.314 \)
Bachelor-Finance ==> Finance theory \( sup = 0.011 \) \( conf = 0.314 \)
Bachelor-Finance ==> Novel \( sup = 0.012 \) \( conf = 0.343 \)
Bachelor-Marine ==> Biology \( sup = 0.016 \) \( conf = 0.314 \)
Bachelor-Marine ==> Language and Literature \( sup = 0.016 \) \( conf = 0.314 \)
Bachelor-Marine Resources ==> Biology \( sup = 0.014 \) \( conf = 0.538 \)
Master-Social ==> Statistics \( sup = 0.012 \) \( conf = 0.333 \)
Master-Social ==> Sociology \( sup = 0.015 \) \( conf = 0.417 \)
Master-Social ==> Economics \( sup = 0.016 \) \( conf = 0.444 \)
Master-Social ==> Political Science \( sup = 0.020 \) \( conf = 0.556 \)
Master-Liberal ==> Social Sciences \( sup = 0.012 \) \( conf = 0.353 \)
Master-Liberal ==> Linguistics \( sup = 0.011 \) \( conf = 0.324 \)
Master-Liberal ==> Literature \( sup = 0.012 \) \( conf = 0.353 \)
Master-Liberal ==> Poem theory \( sup = 0.011 \) \( conf = 0.324 \)
Master-Liberal ==> Special literature \( sup = 0.013 \) \( conf = 0.382 \)
Master-Political Sciences ==> Political Science \( sup = 0.010 \) \( conf = 0.833 \)
Master-Chinese Literature ==> Chinese Literature \( sup = 0.015 \) \( conf = 0.882 \)
Master-Chinese Literature ==> Special literature \( sup = 0.010 \) \( conf = 0.588 \)
Master-Human Resources ==> Business Management \( sup = 0.010 \) \( conf = 0.625 \)
Master-Science ==> Mathematics \( sup = 0.011 \) \( conf = 0.306 \)
Master-Science ==> Applied Sciences \( sup = 0.011 \) \( conf = 0.306 \)
Master-Science ==> Language and Literature \( sup = 0.011 \) \( conf = 0.306 \)
Master-Engineering ==> Arithmetic  sup= 0.037  conf= 0.407
Master-Engineering ==> Electrical Engineering  sup= 0.031  conf= 0.341
Master-Electrical Engineering ==> Arithmetic  sup= 0.014  conf= 0.424
Master-Electrical Engineering ==> Electrical Engineering  sup= 0.016  conf= 0.485
Master-Mechanical Engineering ==> Arithmetic  sup= 0.014  conf= 0.467
Master-Mechanical Engineering ==> Engineering  sup= 0.019  conf= 0.633
Master-Management ==> Mathematics  sup= 0.035  conf= 0.321
Master-Management ==> Business Management  sup= 0.056  conf= 0.514
Master-Management ==> Economics  sup= 0.039  conf= 0.358
Master-Business ==> Business Management  sup= 0.019  conf= 0.704
Master-Business ==> Economics  sup= 0.011  conf= 0.407
Master-Business ==> Language and Literature  sup= 0.010  conf= 0.370
Master-Finance ==> Finance theory  sup= 0.010  conf= 0.556
Master-Marine ==> Mathematics  sup= 0.012  conf= 0.429
Ph.D.-Engineering ==> Applied Sciences  sup= 0.013  conf= 0.591
### Appendix 2

#### Chinese Classification Scheme

<table>
<thead>
<tr>
<th>Generalities</th>
<th>Philosophy</th>
</tr>
</thead>
<tbody>
<tr>
<td>000 特藏 Special Collections</td>
<td>100 哲學總論 Philosophy</td>
</tr>
<tr>
<td>010 目錄學總論 Bibliographies</td>
<td>110 思想學問概說 Thought</td>
</tr>
<tr>
<td>020 圖書館學總論 Library Science</td>
<td>120 中國哲學總論 Chinese Philosophy</td>
</tr>
<tr>
<td>030 國學總論 Sinology</td>
<td>130 東方哲學總論 Oriental Philosophy</td>
</tr>
<tr>
<td>040 類書:百科全書 Encyclopedias</td>
<td>140 西洋哲學總論 Western Philosophy</td>
</tr>
<tr>
<td>050 普通雜誌 General Periodicals</td>
<td>150 理論學 logic</td>
</tr>
<tr>
<td>060 普通會社 General Organizations</td>
<td>160 形而上學:玄學 Metaphysics</td>
</tr>
<tr>
<td>070 普通論叢 General Essays</td>
<td>170 心理學 Psychology</td>
</tr>
<tr>
<td>080 普通叢書 General Collections</td>
<td>180 美學 Esthetics</td>
</tr>
<tr>
<td>090 群經 Collected Classics</td>
<td>190 倫理學 Ethics(Moral Philosophy)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Religions</th>
<th>Natural Sciences</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 總論 Religions</td>
<td>300 總論 Pure Sciences</td>
</tr>
<tr>
<td>210 比較宗教學 Comparative Religions</td>
<td>310 數學 Mathematics</td>
</tr>
<tr>
<td>220 佛教 Buddhism</td>
<td>320 天文 Astronomy</td>
</tr>
<tr>
<td>230 道教 Taoism</td>
<td>330 物理 Physics</td>
</tr>
<tr>
<td>240 基督教 Christianity</td>
<td>340 化學 Chemistry</td>
</tr>
<tr>
<td>250 回教 Islam; Mohammedanism</td>
<td>350 地質 Geology</td>
</tr>
<tr>
<td>260 猶太教 Judaism</td>
<td>360 生物;博物 Biology; Natural History</td>
</tr>
<tr>
<td>270 其他各教 Other Religions</td>
<td>370 植物 Botany</td>
</tr>
<tr>
<td>Code</td>
<td>Category</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
</tr>
<tr>
<td>280</td>
<td>神話 Mythology</td>
</tr>
<tr>
<td>290</td>
<td>術數;迷信;奇跡 Astrology; Superstition</td>
</tr>
<tr>
<td></td>
<td>應用科學類 Applies Sciences</td>
</tr>
<tr>
<td>400</td>
<td>應用科學總論 Applied Sciences</td>
</tr>
<tr>
<td>410</td>
<td>醫學總論 Medical Sciences</td>
</tr>
<tr>
<td>420</td>
<td>家事 Domestic Sciences</td>
</tr>
<tr>
<td>430</td>
<td>農業 Agriculture</td>
</tr>
<tr>
<td>440</td>
<td>工程 Engineering</td>
</tr>
<tr>
<td>450</td>
<td>礦治 Mining and Metallurgy</td>
</tr>
<tr>
<td>460</td>
<td>應用化學;化學工藝 Chemical Technology</td>
</tr>
<tr>
<td>470</td>
<td>製造 Manufacture</td>
</tr>
<tr>
<td>480</td>
<td>商業;各種營業 Trades</td>
</tr>
<tr>
<td>490</td>
<td>商學;經營學 Business</td>
</tr>
<tr>
<td></td>
<td>中國史地類 History and Geography</td>
</tr>
<tr>
<td>600</td>
<td>史地總論 History and Geography</td>
</tr>
<tr>
<td>610</td>
<td>中國通史 General History of China</td>
</tr>
<tr>
<td>620</td>
<td>中國斷代史 Chinese History by Period</td>
</tr>
<tr>
<td>630</td>
<td>中國文化史 History of Chinese Civilization</td>
</tr>
<tr>
<td></td>
<td>中國外交史 History of Foreign Intercourse</td>
</tr>
<tr>
<td>640</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>650</td>
<td>史料 Historical materials</td>
</tr>
<tr>
<td>-----</td>
<td>--------------------------</td>
</tr>
<tr>
<td>660</td>
<td>地理總志 Geography of China</td>
</tr>
<tr>
<td>670</td>
<td>方志 Local Gazetteers of China</td>
</tr>
<tr>
<td>680</td>
<td>類志 Topical Topography</td>
</tr>
<tr>
<td>690</td>
<td>中國遊記 Travels</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>800</th>
<th>語言文字學 Linguistics</th>
<th>900</th>
<th>總論 Arts</th>
</tr>
</thead>
<tbody>
<tr>
<td>810</td>
<td>文學 Literature</td>
<td>910</td>
<td>音樂 Music</td>
</tr>
<tr>
<td>820</td>
<td>中國文學 Chinese Literature</td>
<td>920</td>
<td>建築 Architecture</td>
</tr>
<tr>
<td>830</td>
<td>總集 General Literature</td>
<td>930</td>
<td>雕塑 Sculpture</td>
</tr>
<tr>
<td>840</td>
<td>別集 individual Works</td>
<td>940</td>
<td>書畫 Painting and Calligraphy</td>
</tr>
<tr>
<td>850</td>
<td>特種文藝 Special Literature</td>
<td>950</td>
<td>攝影 Photography</td>
</tr>
<tr>
<td>860</td>
<td>東洋文學 Oriental Literature</td>
<td>960</td>
<td>圖案;裝飾 Decorative Arts</td>
</tr>
<tr>
<td>870</td>
<td>西洋文學 Western Literature</td>
<td>970</td>
<td>技藝 Accomplishments</td>
</tr>
<tr>
<td>880</td>
<td>西方諸小國文學</td>
<td>980</td>
<td>戲劇 Theater</td>
</tr>
<tr>
<td>890</td>
<td>新聞學 Journalism</td>
<td>990</td>
<td>遊戲;娛樂;休閒 Recreation</td>
</tr>
<tr>
<td></td>
<td>類志 Topical Topography</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>方志 Local Gazetteers of China</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>中國遊記 Travels</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>900</th>
<th>總論 Arts</th>
</tr>
</thead>
<tbody>
<tr>
<td>910</td>
<td>音樂 Music</td>
</tr>
<tr>
<td>920</td>
<td>建築 Architecture</td>
</tr>
<tr>
<td>930</td>
<td>雕塑 Sculpture</td>
</tr>
<tr>
<td>940</td>
<td>書畫 Painting and Calligraphy</td>
</tr>
<tr>
<td>950</td>
<td>攝影 Photography</td>
</tr>
<tr>
<td>960</td>
<td>圖案;裝飾 Decorative Arts</td>
</tr>
<tr>
<td>970</td>
<td>技藝 Accomplishments</td>
</tr>
<tr>
<td>980</td>
<td>戲劇 Theater</td>
</tr>
<tr>
<td>990</td>
<td>遊戲;娛樂;休閒 Recreation</td>
</tr>
</tbody>
</table>