

Chapter 5

Evaluation Method

It is important to understand the approximating behavior of `ApproxMAP`. The accuracy of the approximation can be evaluated in terms of how well it finds the real underlying patterns and whether or not it generates any spurious patterns. However, it is difficult to calculate analytically what patterns will be generated because of the complexity of the algorithm.

As an alternative, we have developed a general evaluation method that can objectively evaluate the quality of the results produced by any sequential pattern mining method. Using this method, we are able to understand the behavior of an algorithm empirically by running extensive systematic experiments on synthetic data.

The evaluation method is a matrix of four experiments - (1) random data, (2) patterned data, and patterned data with (3) varying degree of noise, and (4) varying number of outliers - assessed on five criteria: (1) recoverability, (2) precision, (3) the total number of result patterns returned, (4) the number of spurious patterns, and (5) the number of redundant patterns. Recoverability, defined in section 5.2, provides a good estimation of how well the underlying trends in the data are detected. Precision, adopted from ROC analysis [46], is a good measure of how many incorrect items are mixed in with the correct items in the result patterns. Both recoverability and precision are measured at the item level. On the other hand, the numbers of spurious and redundant patterns along with the total number of patterns returned give an overview of the result at the sequence level. In summary, a good model would produce (1) high recoverability and precision, with (2) small number of spurious and redundant patterns, and (3) a manageable number of result patterns.

This evaluation method will enable researchers not only to use synthetic data to benchmark performance in terms of speed, but also to quantify the quality of the results. Such benchmarking will become increasingly important as more data mining methods focus on approximate solutions.

Table 5.1: Parameters for the random data generator

Notation	Meaning
$\ \mathcal{I}\ $	# of items
N_{seq}	# of data sequences
L_{seq}	Average # of itemsets per data sequence
I_{seq}	Average # of items per itemset in the database

Table 5.2: Parameters for the IBM patterned data generator

Notation	Meaning
$\ \mathcal{I}\ $	# of items
$\ \Lambda\ $	# of potentially frequent itemsets
N_{seq}	# of data sequences
N_{pat}	# of base patterns (potentially frequent sequential patterns)
L_{seq}	Average # of itemsets per data sequence
L_{pat}	Average # of itemsets per base pattern
I_{seq}	Average # of items per itemset in the database
I_{pat}	Average # of items per itemset in base patterns

5.1 Synthetic Data

In this section, we describe the four class of synthetic datasets used for each of the four experiments : (1) random data, (2) patterned data, and patterned data with (3) varying degree of noise, and (4) varying number of outliers.

5.1.1 Random Data

Random data is generated by assuming independence between items both within and across itemsets. The probability of an item occurring is uniformly distributed. The number of distinct items and the number of sequences generated are determined by user set parameters $\|\mathcal{I}\|$ and N_{seq} respective. The number of itemsets in a sequence and the number of items in an itemset follow a Poisson distribution with mean L_{seq} and I_{seq} respectively. The full user parameters are listed in Table 5.1

5.1.2 Patterned Data

For patterned data, we use the well known IBM synthetic data generator first introduced in [2]. Given several parameters (Table 5.2), the IBM data generator produces a patterned database and reports the base patterns used to generate it. Since it was first published in 1995, the IBM data generator has been used extensively as a performance benchmark in association rule mining and sequential pattern mining. However, to the best of our knowledge, no previous study has measured how well the various methods recover the known base patterns. In this dissertation, we develop some evaluation criteria to use in conjunction with the IBM data generator to measure the quality of the results.

The data is generated in two phases. First, it generates N_{pat} potentially frequent sequential patterns, called *base patterns*, according to user parameters L_{pat} and I_{pat} . Secondly, each sequence in the database is built by combining these base patterns until the size specified by user parameters L_{seq} and I_{seq} are met. Along with each base pattern, the data generator reports the expected frequency, $E(F_B)$, and the expected length (total number of items), $E(L_B)$, in the database for each base pattern. The $E(F_B)$ is given as a percentage of the size of the database and the $E(L_B)$ is given as a percentage of the number of items in the base pattern.

There are two steps involved in building the base patterns. First, the set of potentially frequent itemsets, Λ , are built by randomly selecting items from the distinct set of items in \mathcal{I} . The probability of an item occurring is exponentially distributed. The size of each itemset is randomly determined using a Poisson distribution with mean I_{pat} . The number of distinct items and the number of potentially frequent itemsets are determined by user set parameters $\|\mathcal{I}\|$ and $\|\Lambda\|$ respective.

Second, the base patterns are then built by selecting, corrupting, and concatenating itemsets selected from the set of potentially frequent itemsets. The selection and corruption is based on the $P(select)$ and $P(corrupt)$ randomly assign to each potentially frequent itemset. The selection probability is exponentially distributed then normalized to sum to 1. The corruption probability is normally distributed. Corrupting means randomly deleting items from the selected potentially frequent itemset. N_{pat} determines how many base patterns to construct, and L_{pat} determines the average number of itemsets in the base patterns. More precisely, the number of itemsets in a base pattern is randomly assigned from a Poisson distribution with mean L_{pat} . Base patterns built in this manner then become the potentially frequent sequential patterns.

The database is then built in a similar manner by selecting, corrupting, and combining the base patterns. As with potentially frequent itemsets, each base pattern is also assigned a separate $P(select)$ and $P(corrupt)$. Just as with potentially frequent itemsets, the $P(select)$ is exponentially distributed then normalized to sum to 1 and $P(corrupt)$ is normally distributed. The $P(select)$ is the likelihood a base pattern will appear in the database. Thus, it is equal to the expected frequency of a base pattern, $E(F_B)$, in the database. The $P(corrupt)$ is the likelihood of a selected base pattern to be corrupted before it is used to construct a database sequence. Corrupting base patterns is defined as randomly deleting items from the selected base pattern. Hence, $1 - P(corrupt)$ is roughly the expected length (total number of items), $E(L_B)$, of the base pattern in a sequence in the database.

$$\begin{aligned} E(F_B) &= P(select) \\ E(L_B) &\simeq 1 - P(corrupt) \end{aligned} \tag{5.1}$$

Each sequence is built by combining enough base patterns until the size required, deter-

Table 5.3: A DB sequence built from 3 base patterns

Base Patterns	(D)	(A, J)	(B)	(A)	(E, H)	(C)	(L)	(F)	(M)
	(G)	(I)		(K)	(F, I)			(D)	
DB Sequence	(D, G)	(A, J)	(B)	(K)	(A, F, I)	(E, H)	(C)	(D, F, L)	

mined by L_{seq} and I_{seq} for the database, is met. Hence, many sequences are generated using more than one base pattern. Base patterns are combined by interleaving them so that the order of the itemsets are maintained.

Table 5.3 demonstrates how 3 base patterns are combined to build a database sequence. The parameters for the database were: $L_{seq}=10$, $I_{seq}=2.5$ and the parameters of the base patterns were: $L_{pat} = 7$, $I_{pat}=2$. The itemsets that are crossed out were deleted in the corruption process.

In essence, sequential pattern mining is difficult because the data has confounding noise rather than random noise. The noise is introduced into each data sequence in the form of tiny bits of another base pattern. In section 5.1.3 we discuss how to add controlled level of random noise in addition to the confounding noise in the patterned data in order to test for the effects of noise.

Similarly, outliers may exist in the data in the form of very weak base patterns. The expected frequency of the base patterns have exponential distribution. Thus, the weakest base patterns can have expected frequency be so small that the base pattern occurs in only a handful of the database sequences. These are in practice outliers that occur rarely in the database. For example, when there are 100 base patterns, 11 base patterns have expected frequency less than 0.1% of which 1 base pattern has expected frequency less than 0.01%. Thus, even when $N_{seq}=10000$, the weakest base pattern would occur in less than 1 sequence ($10,000*0.01\%=1$ seq). In section 5.1.4 we discuss how to add controlled level of strictly random sequences in addition to outliers in the form of very weak base patterns in the patterned data to test for effects of outliers.

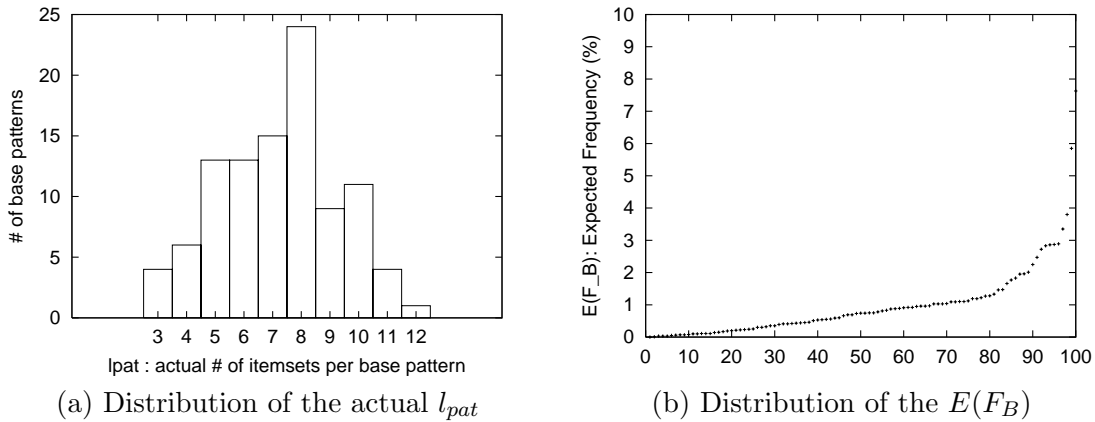
Example

To better understand the properties of the synthetic data generated we look closely at a particular synthetic dataset. A common configuration of the synthetic dataset used in the experiments is given in Table 5.4. The configuration can be understood as follows.

1. There are $\|I\|=1,000$ unique items in the synthetic database.
2. Using these $\|I\|=1,000$ unique items $\|\Lambda\|=5,000$ itemsets were generated at random. These are the potentially frequent itemsets used to construct the base patterns.

Table 5.4: A common configuration of the synthetic dataset used in the experiments

Notation	Meaning	Default value
$\ I\ $	# of items	1,000
$\ \Lambda\ $	# of potentially frequent itemsets	5,000
N_{seq}	# of data sequences	10,000
N_{pat}	# of base pattern sequences	100
L_{seq}	Avg. # of itemsets per data sequence	10
L_{pat}	Avg. # of itemsets per base pattern	7
I_{seq}	Avg. # of items per itemset in the database	2.5
I_{pat}	Avg. # of items per itemset in base patterns	2

Figure 5.1: Distributions from the synthetic database specified in Table 5.4

3. On average there are $I_{pat}=2$ items per itemset in these 5,000 potentially frequent itemsets.
4. $N_{pat}=100$ base patterns were randomly constructed using the 5,000 potentially frequent itemsets.
5. On average there are $L_{pat}=7$ itemsets per base pattern. The actual distribution of the number of itemsets for each of the 100 base patterns, l_{pat} , is given in Figure 5.1(a). Remember that l_{pat} has a poisson distribution with mean at L_{pat} . When $L_{pat}=7$, 10% of the patterns have between 3 or 4 itemsets in the base patterns. On the other hand 5% of the patterns have more than 10 itemsets per base pattern. The remaining 85% of base patterns have between 5 to 10 itemsets per pattern. Values of $L_{pat} < 7$ starts to introduce base patterns of less than 3 itemsets per pattern. Thus, $L_{pat}=7$ is the practical minimum value that will embed sequential patterns of more than 2 itemsets into the synthetic data.
6. $N_{seq}=10,000$ data sequences were constructed using the 100 base patterns.
7. The distribution of the expected frequencies, $E(F_B)$, of the 100 base patterns is given in

Figure 5.1(b). Of 100 base patterns, 11 have $E(F_B) < 0.1\%$ ($0.1\% * 10,000=10$ seq). Of them, 1 base pattern has expected frequency less than 0.01% ($0.01\% * 10,000=1$ seq). As discussed above these are the practical outliers that occur rarely in the database. On the other hand, there are 12 base pattern with $E(F_B) > 2\%$ ($2\% * 10,000=200$ seqs). Of these the four largest $E(F_B)$ are 7.63%, 5.85%, 3.80%, and 3.35% respectively. The other 8 are all between 2% and 3% ($2\% < E(F_B) \leq 3\%$). The majority, 77 base patterns, have $E(F_B)$ between 0.1% and 2% ($10 \text{ seq} = 0.1\% < E(F_B) \leq 2\% = 200 \text{ seqs}$).

8. The base patterns were combined so that on average there are $L_{seq}=10$ itemsets per data sequence and $I_{seq}=2.5$ items per itemset in a data sequence. Note that since $L_{pat}=7$ is the practical minimum for embedding sequential patterns into the synthetic data, L_{seq} should be greater than 7. Thus, in many of our experiments which require us to test on a range of L_{seq} , we vary L_{seq} from 10 to 50.

5.1.3 Patterned data with varying degree of noise

Noise occurs at the item level in sequential data. Therefore, to introduce varying degree of controlled noise into the IBM patterned data, we use a corruption probability α . Items in the patterned database are randomly changed into another item or deleted with probability α . This implies that $1 - \alpha$ is the probability of any item remaining the same. Hence, when $\alpha = 0$ no items are changed, and higher values of α imply a higher level of noise [64].

5.1.4 Patterned data with varying degree of outliers

Outliers are sequences that are unlike most other sequences in the data. That is there are very few sequences similar to the outlier sequence in the data. A randomly generated sequence, such as the sequences generated for the random data, can be such an outlier sequence. Thus, we introduce controlled level of outliers into the data by adding varying number of random sequences to the IBM patterned data. The random sequences are generated using the same parameters L_{seq} , I_{seq} , and $\|\mathcal{I}\|$ as those used to generate the patterned data. In the rest of the dissertation, random sequences added to patterned data is referred to as outliers.

5.2 Evaluation Criteria

The effectiveness of a sequential pattern mining method can be evaluated in terms of how well it finds the real underlying patterns in the data (the base patterns), and whether or not it generates any confounding information. However, the number of base patterns found or missed is not alone an accurate measure of how well the base patterns were detected because it can not take into account how much of a base pattern was detected (which items in the base pattern were detected) or how strong (frequent) the pattern is in the data. Instead, we

Table 5.5: Confusion Matrix

		predicted (Result Patterns Generated)	
		negative	postive
actual (Base Patterns Embedded)	negative	a (NA)	b (Extraneous Items)
	positive	c (Missed Items)	d (Pattern Items)

report a comprehensive view by measuring this information at two different levels; (1) at the item level and (2) at the sequence level.

5.2.1 Evaluation at the item level

At the item level, we adapt the ROC analysis to measure recoverability and precision. ROC analysis is commonly used to evaluate classification systems with known actual values [46]. The confusion matrix contains information about the actual and predicted classifications [46]. The confusion matrix for our evaluation is given in Table 5.5. The actual patterns are the base patterns that were embedded into the database. The predicted patterns are the result patterns generated from any sequential pattern mining algorithm. Then the true positive items, called *pattern items*, are those items in the result patterns that can be directly mapped back to a base pattern. The remaining items in the result patterns, the false positive items, are defined as *extraneous items*. These are items that do not come from the embedded patterns, but rather the algorithm falsely assumes to be part of the base patterns. There are a couple of reasons why this occurs. We discuss this in more detail in section 5.4. The items from the base pattern that were missed in the result patterns, the false negative items, are the *missed items*. In this context, there are no true negative items (cell a). Thus, we use only the cells b, c, and d in our evaluation.

Using the confusion matrix we measure two criteria at the item level. *Recoverability* measures how much of the base patterns have been found. *Precision* measures how precise are the predictions made about the base patterns. That is, precision measures how much confounding information (extraneous items) are included with the true pattern items.

Normally recall, $(\frac{d}{c+d})$, the true positive rate, is used to measure how much of the actual pattern has been found. However, recall is not accurate in this application because base patterns are only *potentially* frequent sequential patterns in the data. The actual occurrence of a base pattern in the data, which is controlled by $E(F_{B_i})$ and $E(L_{B_i})$, varies widely.

$E(F_{B_i})$ is exponentially distributed then normalized to sum to 1. Thus, some base patterns have tiny $E(F_{B_i})$. These base patterns do not exist in the data or occur very rarely. Recovering these patterns are not as crucial as recovering the more frequent base patterns.

$E(L_{B_i})$ controls how many items on average in the base patterns are injected into one occurrence of the base pattern in a sequence. This means that one sequence in the database is not expected to have all the items in the embedded base pattern. Remember that before

Table 5.6: Item counts in the result patterns

Notation	Meaning	Equation
N_{item}	total number of items	$\sum_{\text{rslt pat } \{P_j\}} \ P_j\ $
N_{patI}	total number of pattern items	$\sum_{\text{rslt pat } \{P_j\}} (\max_{\text{base pat } \{B_i\}} \ B_i \otimes P_j\)$
N_{extraI}	total number of extraneous items	$\sum_{\text{rslt pat } \{P_j\}} (\ P_j\ - \max_{\text{base pat } \{B_i\}} \ B_i \otimes P_j\)$

a base pattern is embedded into a data sequence we corrupt the base pattern by randomly deleting items from it. $E(L_{B_i})$ controls how many items on average are deleted in this process. Thus, we can not expect to find all the items in a base pattern in one database sequence.

Therefore, taking $E(F_{B_i})$ and $E(L_{B_i})$ into account, we designed a weighted measure, *recoverability*, which can more accurately evaluate how much of the base patterns have been recovered. Specifically, given (1) a set of base patterns, $\{B_i\}$, along with $E(F_{B_i})$ and $E(L_{B_i})$ for each base pattern, and (2) a set of result patterns, $\{P_j\}$, let each result pattern map back to the most similar base pattern. That is, the result pattern, P_j , is matched with the base pattern, B_i , if the longest common subsequence between P_j and B_i , denoted as $B_i \otimes P_j$, is the maximum over all base patterns. We indicate this matching by referring to the matched result patterns with two indices. $P_j(i)$ denotes that pattern P_j has been mapped to base pattern B_i .

Now let $P_{max}(i)$ be the max pattern for base pattern B_i . A *max pattern*, $P_{max}(i)$, is the result pattern that shares the most items with a base pattern, B_i , over all result patterns mapped to the same base pattern. Furthermore, at least half of the items in P_j has to come from the base pattern B_i . Thus, $\max_{\text{rslt pat } \{P_j(i)\}} \|B_i \otimes P_j\|$ ¹ is the most number of items recovered for a base pattern B_i . In essence, max patterns recovered the most information about a particular base pattern. Note that, there is either one or no max pattern for each base pattern. There could be no max pattern for a base pattern if none of the result patterns recovered enough of the items from the base pattern.

Since $E(L_{B_i}) \cdot \|B_i\|$ is the expected number of items (from the base pattern B_i) in one occurrence of B_i in a data sequence, $\frac{\max_{\text{rslt pat } \{P_j(i)\}} \|B_i \otimes P_j\|}{E(L_{B_i}) \cdot \|B_i\|}$ would be the fraction of the expected number of items found. $E(L_{B_i})$ is an expected value, thus sometimes the actual observed value, $\max_{\text{rslt pat } \{P_j(i)\}} \|B_i \otimes P_j\|$ is greater than $E(L_{B_i}) \cdot \|B_i\|$. In such cases, we truncate the value of $\frac{\max_{\text{rslt pat } \{P_j(i)\}} \|B_i \otimes P_j\|}{E(L_{B_i}) \cdot \|B_i\|}$ to one so that recoverability stays between 0 and 1. Intuitively, if the recoverability of the mining is high, major portions of the base patterns have been found. Then, we define the recoverability as follows,

$$\text{Recoverability } \mathcal{R} = \sum_{\text{base pat } \{B_i\}} E(F_{B_i}) \cdot \min \left\{ \begin{array}{l} 1 \\ \left(\frac{\max_{\text{rslt pat } \{P_j(i)\}} \|B_i \otimes P_j\|}{E(L_{B_i}) \cdot \|B_i\|} \right) \end{array} \right. \quad (5.2)$$

¹ $\|seq_i\|$ =length of seq_i denotes the total number of items in seq_i

In ROC analysis, *precision*, $\frac{d}{b+d}$, is the proportion of the predicted positive items that were correct. It is a good measure of how much of the prediction is correct [46]. In sequential pattern mining, precision measures how much confounding information (extraneous items) are mixed in with the pattern items in the result pattern. Remember that when the result pattern P_j is mapped to base pattern B_i , the items in both the result pattern and the base pattern, $B_i \otimes P_j$, are defined as pattern items. Note that the result pattern P_j is mapped to base pattern B_i , when $\|B_i \otimes P_j\|$ is maximum over all base patterns. Thus, the number of pattern items for a result pattern, P_j , is $\max_{\{B_i\}} \|B_i \otimes P_j\|$. The remaining items in the result pattern, P_j , are the extraneous items. The different item counts in the result patterns are give in Table 5.6.

Denoted as \mathcal{P} , precision can be calculated using Table 5.6 as either

$$\text{Precision } \mathcal{P} = \frac{\sum_{\text{rslt pat } \{P_j\}} (\max_{\text{base pat } \{B_i\}} \|B_i \otimes P_j\|)}{\sum_{\text{rslt pat } \{P_j\}} \|P_j\|} \times 100\% \quad (5.3)$$

or

$$\text{Precision } \mathcal{P} = 1 - \frac{\sum_{\text{rslt pat } \{P_j\}} (\|P_j\| - \max_{\text{base pat } \{B_i\}} \|B_i \otimes P_j\|)}{\sum_{\text{rslt pat } \{P_j\}} \|P_j\|} \times 100\% \quad (5.4)$$

We tend to report using Equation 5.4 to indicate the exact number of extraneous items in the results.

5.2.2 Evaluation at the sequence level

At the sequence level, the three criteria that we measure are (1) the total number of result patterns, (2) the number of spurious patterns, and (3) the number of redundant patterns. To do so, we categorize the result patterns into spurious, redundant, or max patterns depending on the composition of pattern items and extraneous items. We do not report on the number of max patterns because it can be easily calculated by

$$N_{max} = N_{total} - N_{spur} - N_{redun}$$

Spurious patterns are those that were not embedded into the database, but what the algorithms incorrectly assumed to be sequential patterns in the data. In this evaluation, spurious patterns are defined as the result patterns that have more extraneous items than pattern items. As discussed in the previous section, max patterns are those that recover the most pattern items from each of the base patterns. The remaining sequential patterns are redundant patterns. These are result patterns, P_a , which match with a base pattern, B_i , but there exists another result pattern, P_{max} , that match with the same base pattern but better in the sense that $\|B_i \otimes P_{max}\| \geq \|B_i \otimes P_a\|$. Therefore these patterns are redundant data that clutter the results.

Table 5.7: Evaluation Criteria

criteria	Meaning	Level	Unit
\mathcal{R}	Recoverability: the degree of the base patterns detected (Eq. 5.2)	item	%
\mathcal{P}	Precision: 1-degree of extraneous items in the result patterns (Eq. 5.4)	item	%
N_{spur}	# of spurious patterns ($N_{extraI} > N_{patI}$)	seq	# of patterns
N_{redun}	# of redundant patterns	seq	# of patterns
N_{total}	total # of result patterns returned	seq	# of patterns

5.2.3 Units for the evaluation criteria

Recoverability and precision is reported as a percentage of the total number of items in the result ranging from 0% to 100%. In comparison, the spurious patterns and redundant patterns are reported as number of patterns. These measures can easily be changed to percentage of the total number of result patterns as needed.

We report the actual number of patterns because the number of spurious patterns can be tiny as a percentage of total number of result patterns. In fact, by definition we expect that there will be only a few spurious patterns if the algorithm is reasonably good. In such situations, we would want to be see exactly how few spurious patterns are in the result rather than its proportion in the result. For example, one of our experiments on the support model ² had over 250,000 result patterns of which 6,648 (2.66%) were spurious patterns.

Unlike spurious patterns, redundant patterns are not incorrect patterns. Sometimes, they can even have additional information, such as suggesting slight variations of a strong pattern in the data. The most negative effect of redundant patterns is the confounding effect it can have on understanding the results when there are too many of them. Hence, the exact number of redundant patterns is directly related to the interference factor. For example, it is easy to glean some information and/or ignore 10 redundant patterns of 20 result patterns but not so easy to work through 50% of 250,000 patterns.

The five evaluation criteria is summarized in Table 5.7.

5.3 Example

Let Table 5.8 be the base patterns used to construct a sequence database. The expected frequency $E(F_{B_i})$, the expected length after corruption $E(L_{B_i})$, and the actual length $\|B_i\|$ of the base patterns are also given. In addition, let Table 5.9 be the result patterns returned by a sequential pattern mining algorithm. The actual length of the result patterns is given in column $\|P_j\|$. Then, the evaluation is done in the following steps:

1. *Identify total number of result patterns, $N_{total} = \|\{P_j\}\|$. $N_{total}=5$ in this example.*

²We did a comparison study between our method and the conventional support model reported in chapter 7.

Table 5.8: Base Patterns $\{B_i\}$: $N_{pat} = 3, L_{pat} = 7, I_{pat} = 2$

ID	Base Patterns B_i								$E(F_{B_i})$	$E(L_{B_i})$	$\ B_i\ $
B_1	(PR)	(Q)	(IST)	(IJ)	(U)	(D)	(NT)	(I)	0.566	0.784	13
B_2	(FR)	(M)	(GK)	(C)	(B)	(Y)	(CL)		0.331	0.805	10
B_3	(D)	(AV)	(CZ)	(HR)	(B)				0.103	0.659	8

Table 5.9: Result Patterns $\{P_j\}$

ID	Result Pattern P_j								$\ P_j\ $
P_1	(PR)	(Q)	(I)	(IJ)	(IJU)	(U)	(D)	(T)	12
P_2	(GKQ)	(IT)	(IJ)	(D)	(NT)				10
P_3	(P)	(IS)	(U)	(DV)	(NT)				8
P_4	(FR)	(M)	(C)	(BU)	(Y)	(CL)			9
P_5	(F)	(AV)	(CL)	(BSU)	(I)				9

Table 5.10: Worksheet : $\mathcal{R} = 84\%, \mathcal{P} = 1 - \frac{12}{48} = 75\%, N_{total} = 5, N_{spur} = 1, N_{Redun} = 2$

ID	Base Pattern B_i								$\ B_i\ $	$E(F_{B_i})$	$E(L_{B_i})$	$E(L_{B_i}) \cdot \ B_i\ $
ID	Result Pattern P_j								$\ P_j\ $	N_{patI}	N_{extraI}	$\mathcal{R}(B_i)$
B_1	(PR)	(Q)	(IST)	(IJ)	(U)	(D)	(NT)	(I)	13	0.566	0.784	10
P_1	(PR)	(Q)	(I)	(IJ)	(IJU)	(U)	(D)	(T)	12	9	3	9/10=0.9
P_2		(GKQ)	(IT)	(IJ)		(D)	(NT)		10	8	2	Redundant
P_3	(P)		(IS)		(U)	(DV)	(NT)		8	7	1	Redundant
B_2	(FR)	(M)	(GK)	(C)	(B)	(Y)	(CL)		10	0.331	0.805	8
P_4	(FR)	(M)		(C)	(BU)	(Y)	(CL)		9	8	1	8/8=1
B_3	(D)	(AV)	(CZ)	(HR)	(B)				8	0.103	0.659	5
P_5	(F)	(AV)	(CL)		(BSU)	(I)			9	4	5	Spurious

Table 5.11: Evaluation Results

criteria	Meaning	Results
\mathcal{R}	Recoverability: the degree of the base patterns detected	84%
\mathcal{P}	Precision: 1-degree of extraneous items in the result patterns	75%
N_{spur}	# of spurious patterns	1
N_{redun}	# of redundant patterns	2
N_{total}	total # of patterns returned	5

2. *Map result patterns to base patterns.* Each result pattern, P_j , is mapped to the best matching base pattern B_i such that $\|B_i \otimes P_j\|$ is maximized over all base patterns B_i in Table 5.10. For result pattern, P_5 , $\|B_1 \otimes P_5\| = \|\{(U)(I)\}\| = 2$, $\|B_2 \otimes P_5\| = \|\{(F)(C)(B)\}\| = 3$, and $\|B_3 \otimes P_5\| = \|\{(AV)(C)(B)\}\| = 4$. Thus, result pattern P_5 is mapped to base pattern B_3 .

3. *Count the number of pattern items and extraneous items for each result pattern.* For each result pattern in Table 5.10, the number of pattern items and extraneous items are given in the fourth and fifth column labeled N_{patI} and N_{extraI} . Result pattern P_1 has 9 pattern items it shares with B_1 ((PR)(Q)(I)(IJ)(U)(D)(T)) and 3(=12-9) extraneous items ((IJU)).

4. *Calculate precision, \mathcal{P} .* The total number of pattern items is $9+8+7+8+4=36$. The total number of items in the result pattern is $12+10+8+9+9=48$. Thus, the total number of extraneous items is $48-36=12$.

$$\mathcal{P} = \left(1 - \frac{12}{48}\right) \times 100\% = 75\%$$

5. *Identify spurious patterns, N_{spur} .* If a result pattern, P_j , has more extraneous items than pattern items, it is classified as a spurious pattern. P_5 is a spurious pattern because $4 < 5$. Therefore, $N_{spur} = 1$.
6. *Identify max patterns, $P_{max}(i)$.* Of the remaining result patterns, for each base pattern, B_i , identify the max result pattern such that $\|B_i \otimes P_j\|$ is maximized over all result patterns $P_j(i)$ mapped to B_i . In Table 5.10, result patterns are sorted by $\|B_i \otimes P_j\|$ for each base pattern. P_1 and P_4 are max patterns for B_1 and B_2 respectively. B_3 does not have a max pattern.
7. *Identify redundant patterns, N_{redun} .* Any remaining result patterns are redundant patterns. P_2 and P_3 are redundant patterns for B_1 that confound the results. Hence, $N_{redun} = 2$.
8. *Calculate recoverability, \mathcal{R} .* For each max pattern, calculate recoverability with respect to B_i , $\mathcal{R}(B_i) = \frac{\|B_i \otimes P_{max}(i)\|}{E(L_{B_i}) \cdot \|B_i\|}$. Truncate $\mathcal{R}(B_i)$ to 1 if necessary. Weight and sum over all base patterns.

$$\begin{aligned} \mathcal{R} &= E(F_{B_1}) \cdot \mathcal{R}(B_1) + E(F_{B_2}) \cdot \mathcal{R}(B_2) + E(F_{B_3}) \cdot \mathcal{R}(B_3) \\ &= 0.566 \cdot \frac{9}{10} + 0.331 \cdot \frac{8}{8} + 0.103 \cdot 0 \\ &= 0.84 = 84\% \end{aligned}$$

5.4 A Closer Look at Extraneous Items

Extraneous items are those items that are part of the result pattern the algorithms found, but were not embedded into the database intentionally (false positive items). That is, they are not part of the mapped base pattern. There are a couple of reasons why an algorithm would falsely assume extraneous items to be part of the base patterns.

The most obvious reason would be that the data mining algorithm is incorrect. Algorithms can inadvertently inject items into the real embedded patterns. These artificially created items are incorrect. Our evaluation method would correctly report them as extraneous items.

A more likely related reason would be that the data mining algorithm is inaccurate. Different from being incorrect, in these cases the extraneous items do occur *regularly* in the database. However, this is a random occurrence because these items do not come from the base patterns we embedded into the database. Furthermore, it is rooted on the definition of

Table 5.12: Repeated items in a result pattern

ID	len	N_{pat}	N_{extral}	Patterns
P_1	16	13	3	(G) (E) (A E F) (A E) (H) (A D) (B H) (C I) (B H)
B_1	13			(G) (E F) (A) (H) (A D) (B H) (C I) (B H)

regular that is either explicitly or implicitly specified within the model used to define patterns in the algorithm. That means that the definition of patterns used in the algorithm is not accurate enough to differentiate between random occurrences and real patterns in the data. In any case, these inaccurate items are also reported as extraneous items.

In reality, the most common extraneous items are repeated items. We use the most conservative definition of extraneous items. Thus, the current definition will classify repeated items as extraneous items. That is when an item in the base pattern is reported in the result pattern more than once, we consider only one of them as being a pattern item. All other items are classified as extraneous items. Table 5.12 is an illustration from one of our experiments. The E in the second and fourth, and A in the third itemset are all repeated items that were classified as extraneous items. All are lightly colored to indicate its relatively weak presence in the data. Clearly, the E comes from the second itemset in the base pattern, and the A comes from the third itemset in the base pattern. However, there is a much stronger presence of E in the third itemset and A in the fourth itemset of the result pattern. Therefore, the three repeated items were classified as extraneous items.

Another possibility is due to the limitations of the evaluation method. The current evaluation method maps each result pattern to only one base pattern. Thus, any items which do not come from the designated primary base pattern is reported as extraneous items even though they might come from another base pattern. However, statistically with enough data and not enough base patterns, a new underlying pattern can emerge. Recall that we build sequences in the database by combining different base patterns. Consequently, when enough sequences are generated by combining multiple base patterns in a similar way, it will produce a new underlying pattern that is a mix of the basic base patterns. This new mixed pattern will occur regularly in the database and a good algorithm should detect it.

This phenomena can be best depicted through an example from our experiment. A result pattern P_1 with 11 extraneous items of 22 items in total is given in Table 5.13. In this table, color does not represent item weights. The dark items in P_1 are pattern items, and the light items in P_1 are extraneous items. P_1 is mapped to B_1 because they share the most items $\|B_1 \otimes P_1\| = 11$. The 11 light colored items in the result pattern P_1 , which do not come from B_1 , are reported as the extraneous items. Clearly 10 of the 11 extraneous items are from B_2 . There is only one real extraneous item in the third itemset, F (underlined). F is a repeated item from B_1 . This result pattern suggests that there is a group of sequences that combine

Table 5.13: A new underlying trend emerging from 2 base patterns

ID	len	Patterns
P_1	22	(F, J) (A,F,H,N) (E,K,Q) (A,E,G,L,P) (A,O) (G,M,R,S) (A,B)
B_1	14	(F) (F) (K) (A,E,G,L) (G,M,R) (B) (C,D,O)
B_2	14	(J) (A,H,N) (Q) (P) (A,O) (S) (A) (G,L) (I) (R)

the two base patterns, B_1 and B_2 , in a similar manner to create a new underlying pattern P_1 . Essentially, a new underlying pattern can emerge when two base patterns combine in a similar way frequently enough. Such phenomena arise in the IBM data when N_{seq}/N_{pat} is large.

When algorithms detect such mixed patterns, the current evaluation method will incorrectly report all items not belonging to the primary base pattern as extraneous items. Fortunately, such situation could be easily detected. If we were to pick out all the extraneous items from a particular result pattern, then build a separate sequence with them, it should become a subsequence of considerable length of another base pattern. That is the result pattern minus all items in the primary base pattern should produce a sequence very similar to another base pattern.

Nonetheless, incorporating this information into the evaluation upfront is quite difficult. Mapping each result pattern to more than one base pattern for evaluation introduces many new complications. Not the least of which is the proper criteria for mapping result patterns to base patterns. That is how much of a base pattern is required to be present in the result pattern for a mapping to occur. Obviously, one shared item between the result pattern and the base pattern is not enough. But how many is enough? This and other issues will be studied in future work for a more comprehensive evaluation method. For now, whenever there is a large number of extraneous items, we investigate how much of the extraneous items could be mapped back to a non-primary base pattern and include this information in the results.