Use of data mining techniques in the discovery of spatial and temporal earthquake relationship

Gordon Standart, Manuel Penaloza and Ziliang Zong

Department of Mathematics and Computer Science

South Dakota School of Mines and Technology

Rapid City, SD 57701

Gordon.Standart@Mines.sdsmt.edu, Manuel.Penaloza@sdsmt.edu,

Ziliang.Zong@sdsmt.edu

Abstract

Market Basket Analysis (MBA) and Sequential Pattern Analysis (SPA) are important data mining techniques that analyze subsequence patterns in transaction data. This paper presents an analysis of five-year transaction earthquake data that was downloaded from the USGS website. The earth is represented as a grid of regions, and a transaction corresponds to earthquakes that occurred within a 6-hour time window. Even though earthquakes may not have a direct connection, we wanted to identify some MBA and SPA association rules satisfying some minimum user confidence threshold and showing the coincidence of earthquakes at different regions. In the case of SPA, we identified rules with some window gaps between transactions but with low confidence.

The popular Apriori data mining algorithm used for MBA analysis was modified to find frequent subsequences and MBA/SPA rules. Results of these rules with their rule confidence values are provided and analyzed.

1 Introduction

The discovery of hidden patterns in data sets is an important data mining task that has been used successfully by companies. For instance, Amazon analyses customer purchases to make suggestions to other customers of which items they should purchase next after they have purchased a particular item. These recommendations are the result of patterns found in the sequence of purchases by all customers. Intra-transaction analysis using market basket analysis (MBA) are performed to transaction datasets to find high confidence association rules that predict which items on a transaction occur or are purchased together. MBA identifies co-occurrence relationships of items in a transaction. For instance, analyzing supermarket purchases, you may find the rule:

$$\{banana, milk\} \rightarrow \{bread\} \quad 0.9$$

This rule indicates that most of the customers who buy at least these three items on a particular day, if they buy *bananas* and *milk*, usually buy *bread*, with a 90% confidence on that rule. It is also possible to identify inter-transaction relationships, also known as "happens-after" relationships [1], where a set of items occurs before another set of items by using sequential pattern analysis (SPA) technique. That is, SPA finds patterns in datasets using the temporal information of the transactions, which it not used by MBA. Amazon recommendations are a typical application of the SPA technique. Under SPA, The association rule:

$$\{banana, milk\} \Rightarrow \{bread\} \quad 0.9$$

would mean that customers who bought *banana* and *milk* on one day or in some predetermined time window, usually will come back to buy *bread* on a future day or on a later time window. We use a thicker arrow for SPA to differentiate from an MBA rule. It is important for SPA to define the time window of interest between items or events of the left set of the rule with items or events on the right set.

Several algorithms have been written by data mining researchers for MBA and SPA techniques. Transactions under the MBA technique were first analyzed with the AIS algorithm [2]. Later, two faster algorithms were introduced, Apriori and AprioriTiD [3]. These two algorithms have been improved by data mining researchers, such as the Partition algorithm [4] and have generated SPA algorithms, such as AprioriAll [5], Spade [1], GSP [6], and PrefixSpan [7]. The different versions of the Apriori algorithm "differ in how they check candidate itemsets against the database" [8]. A candidate itemset is a set of items that appear together in a transaction. For instance, the Partition algorithm splits the transaction dataset into several partitions, and each one is stored in main memory. The algorithm finds local frequent itemsets for a database partition, and then merges them together to find global frequent itemsets. In the Apriori algorithm, the transaction dataset is accessed k times the dataset, where k is the size of the largest itemset. Instead, the Partition algorithm scans the dataset only twice. The first time to read and find the local frequent itemsets, and the second time to compute the frequency of all the merged frequent itemsets.

2 Problem Description

A set of transactions from user interactions with a system, such as customer purchases, could be analyzed using data mining algorithms to find hidden patterns in the data. Each transaction consists of a least three fields, some customer or transaction ID, the transaction timestamp, and the set of items associated with a transaction (e.g., items purchased by a customer). Table 1 shows a set of customer purchases, where for simplicity, items, customers and transaction timestamps are represented by integers.

Table 1: Sequence of customer purchases.

Customer ID	2	3	1	2	5	6	3	1	2	4	5	2	1
Timestamp	1	2	4	6	7	9	12	13	15	20	21	23	26
Item IDs	2 5 21	2 4 5 20	5 8	1 31 37	2 3 4	2 7 10 20	3 7 21 22	3 4 6 9 10	7 32 35	20 21 22	7 10 11 12	10 22 24	30 31 32
Set	1	2	3	4	5	6	7	8	9	10	11	12	13

If we ignore the customer ID, the set of transactions in this table can be represented as a sequence of ordered purchases, **S**: < {2, 5, 21}, {2, 4, 5, 20}, {5, 8}, {1, 31, 37}, … {30, 31, 32} >. An itemset is a non-empty set of items. A sequence is a list of itemsets, ordered by transaction timestamp. If we denote a sequence, **S,** by $<s_1, s_2, ..., s_n>$, where s_i is an itemset, a sequence **A** denoted by $<a_1, a_2, \ldots, a_m>$ is a subsequence of **S** if the **m** ordered sets in **A** are subsets in the **n** ordered sets of **S**, where $\mathbf{m} \leq \mathbf{n}$. For instance, the subsequence < {2}, {7} >, has the first set {2} as a subset of the first set {2, 5, 21} in S, and the set {7} is a subset of the set {2, 7, 10, 20}, {3, 7, 21, 22}, or {7, 10, 11, 12} in S. If we do not ignore customer ID, then, the following sequences, one for each customer could be defined from Table 1:

SC1: < {5, 8}, {3, 4, 6, 9, 10}, {30, 31, 32} >
SC2: < {2, 5, 21}, {1, 31, 37}, {7, 32, 35}, {10, 22, 24} >
SC3: < {2, 4, 5, 20}, {3, 7, 21, 22} >
SC4: < {20, 21, 22} >
SC5: < {2, 3, 4}, {7, 10, 11, 12} >
SC6: < {2, 7, 10, 20} >

Given this set of sequences, the subsequence < {2}, {7} > is supported by the sequences for customers 2, 3 and 5. The subsequence < {2}, {7} > is a difference subsequence than < {2, 7} >. The latter is supported by the single transaction associated with customer 6.

We are interested in identifying the most frequent subsequences. A subsequence is frequent if its support is greater than or equal to some user-defined minimum support. Our definition for support follows the definitions stated in [9]. The *absolute* support of a subsequence **A** in

a transaction dataset is the number of subsequences **A** in the dataset. On the other hand, the *relative* support of **A** is the fraction of the absolute support divided by the number of sequences in the dataset. We will use both definitions in this paper. That is, the *relative* support of < {2}, {7} > is 3/6. As a support count, the *absolute* support is simply the number of customers supporting the subsequence. For the subsequence < {2}, {7} >, its *absolute* support is 3. If we assume the transaction dataset as a single sequence, instead of one for each customer, then the *absolute* support is the number of subsequences in the entire sequence. That is, the *absolute* support of the subsequence < {2}, {7} > is 12 (it includes overlapping subsequences).

The number of items and number of sets for a subsequence could be more than two. For instance, the subsequence < {2, 5}, {31}, {32} > is only present in the sequence for customer 2, and its relative support is 1. The subsequence < { 4, 6, 9, 10} > is present in the sequence for customer 1. In any case, an n-sequence is any sequence with n items in the entire sequence, regardless of the number of sets in the sequence. That is < { 4, 6, 9, 10} > is a 4-sequence.

3 Data Description

The earthquake data for the entire 2005-09 was collected from the USGS website http://neic.usgs.gov/neis/epic/epic_global.html. This file has 67,828 records. Only quakes with magnitude 4.0 or higher were downloaded from the USGS website. In this data file, the following fields are recorded for each earthquake:

Year, month, day, hhmmss.ms, latitude, longitude, magnitude, depth

The first few records of this file are:

```
2005  01  01  004734.62    3.98    94.48  4.8   30
2005  01  01  005257.32    8.36    92.46  4.8   30
2005  01  01  005646.64    8.27    92.95  4.7   30
2005  01  01  012005.36   13.78   -88.78  4.7  193
2005  01  01  014224.85    7.29    93.92  5.0   30
```

Two transformation processes were performed in the input file:

1. Transform the latitude and longitude to an earth region of 2^o x 2^o. Each region has a RegionID of the form RXXXYYY where XXX is computed as (latitude + 90) / 2, and YYY is computed as (longitude + 180) / 2.
2. Transform the date and time into a time field representing the number of seconds with respect to 12am of 01/01/2005.

Transactions are generated from this transformed file by placing earthquakes within a 6-hour time window into a single transaction. No duplicate regions are stored in each transaction.

The number of transactions is based on the time window. For a 6-hour time window, for the period 01/01/2005 to 12/31/2009, the maximum number of transactions is:

(Num of days from 1/1/2005 to 12/31/2009) * (24 hrs/day) / (6 hrs) = 1826 * 24 / 6 = 7304

The transaction dataset was analyzed before it was processed by the MBA and SPA algorithms. Figure 1 shows the number of transactions with 1, 2, …, n regions, for a 6-hour time window. The actual number of transactions with at least an earthquake was 7286 instead of the maximum of 7304. There are 102 transactions with a single region. The largest number of transactions (912) has 8 regions. There is a single transaction with 20 regions. The average number of regions per transaction is 7.62.

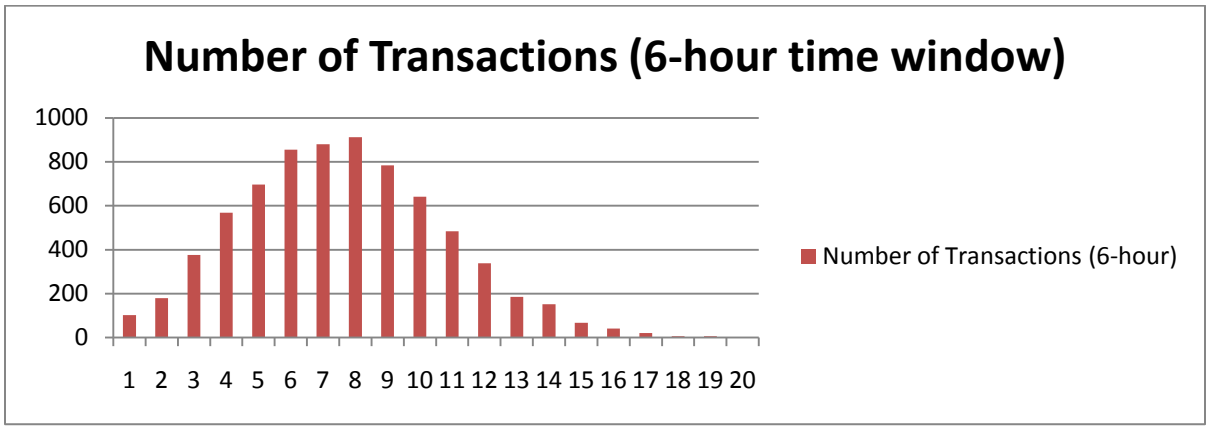

Figure 1. Number of transactions with 1, 2,… regions on a 6-hr time window

Figure 2 shows the number of transactions with 1, 2, …, n regions, for a 3-hour time window. There are 1909 transaction with a single region. At the other side of the histogram, there are only 3 transactions with 14 regions. The average number of regions per transaction is 3.97. The histogram is a little skewed to the right. For smaller time windows, the skew is more significant.

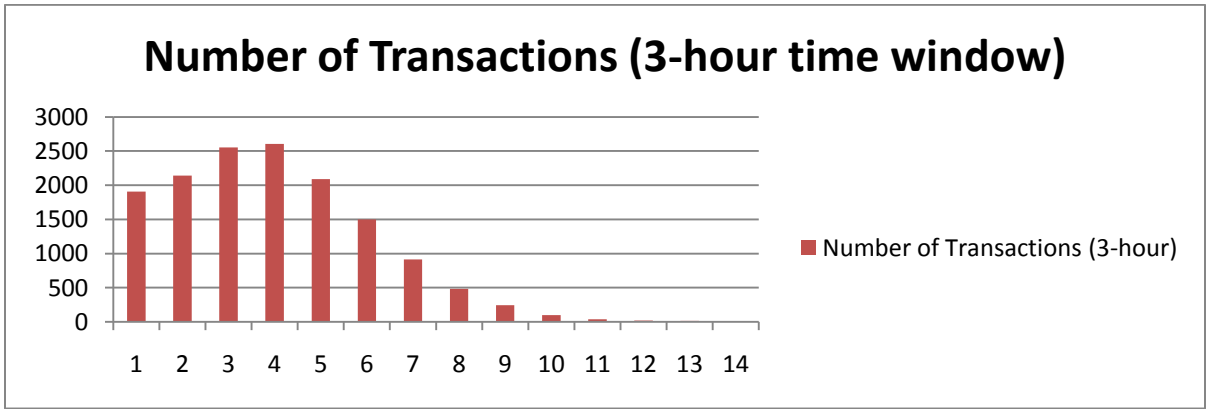

Figure 2. Number of transactions with 1, 2,… regions on a 3-hr time window

Figure 3 shows the number of transactions with 1, 2, …, n regions, for a 12-hour time window. There is not a single transaction with one region. There are 5 transactions with two regions. At the other side of the histogram, there are only 2 transactions with 31 regions. The average number of regions per transaction is 14.57.

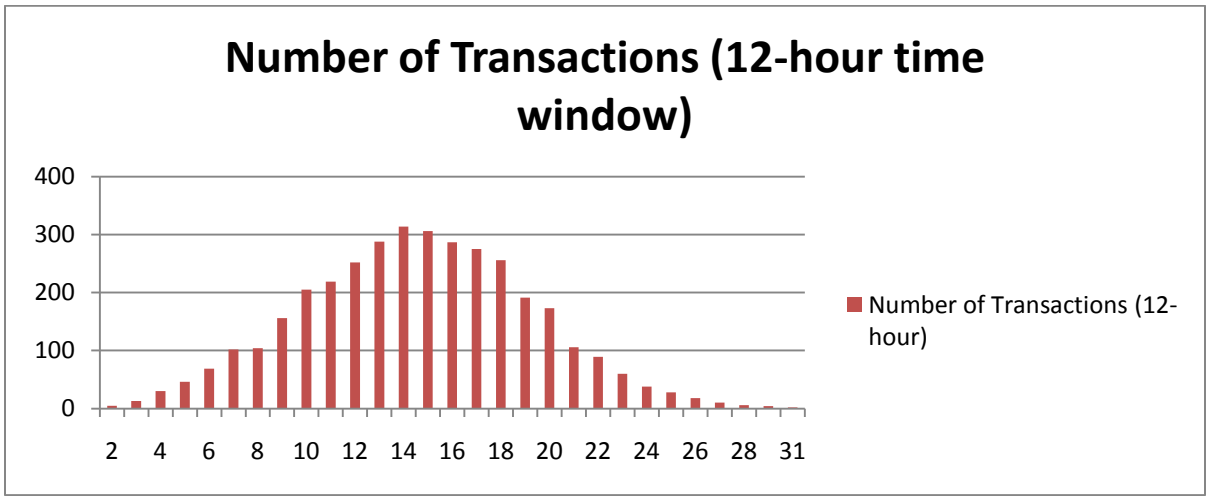

Figure 3. Number of transactions with 1, 2,… regions on a 12-hr time window

From this point we will be using the 6-hour time window which provides a lower granularity than the 12-hour time window, and it does not show a significant skew on the graph. A second analysis consists in finding out the number of earthquakes per region for the 6-hour time window. Figure 4 shows the number of transactions containing each region. It only shows regions present in at least 100 transactions.

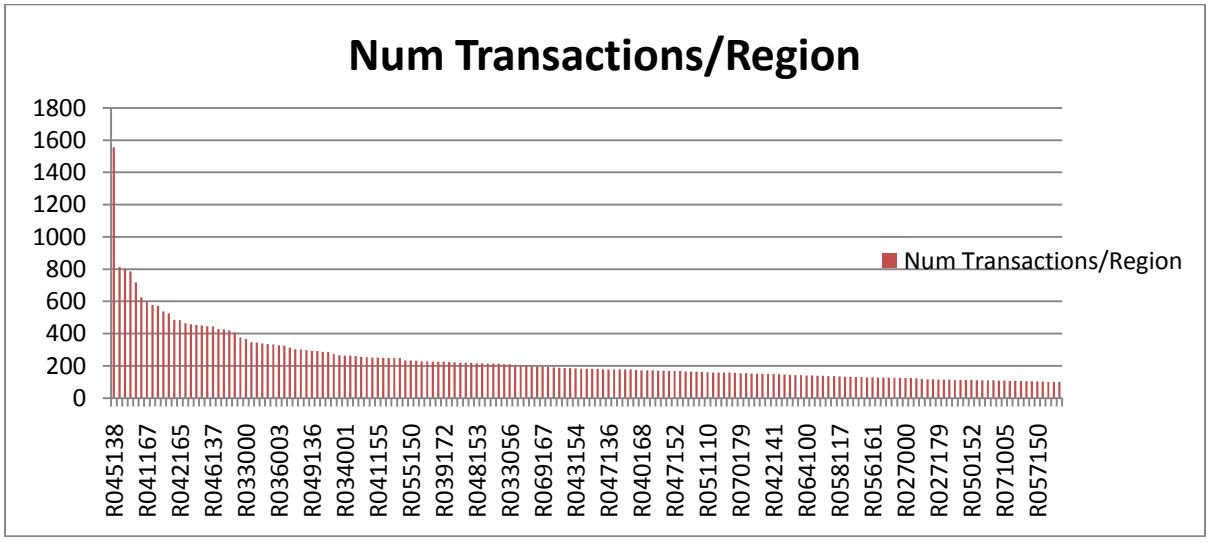

Figure 4. Number of transactions per region.

In Figure 4, there are 2553 regions present in at least one earthquake transaction, out of 16200 distinct regions. The most active region is R045138 which maps to the region with latitude 0^o-2^o and longitude 96^o-98^o and is a member of 1556 transactions. This is the region where the earthquake with magnitude 9.5 produced a devastating tsunami that killed more than 200,000 people. The second most active region is R068167 (label not shown), which is a member of 813 transactions.

4 Implementation Details

Our goal is to identify the one-set and two-set frequent sequences in the transaction dataset that have a ***relative* support, s,** higher than some user-specified ***minsupport***. The notation used during the implementation for one-set and two-set sequences is (R_i is an earth region):

- $<\{R_1, R_2, \ldots, R_n\}>(m)$ represents a one-set sequence where **m** is the support of the frequency defined as the number of transactions that contains the sequence.
- $<\{ R_1, R_2, \ldots, R_n \}>[n]<\{R_s, R_t, \ldots, R_w\}>(m)$ represents a two-set sequence where earthquakes in the regions in the second set ($\{R_s, R_t, \ldots, R_w\}$) happened after earthquakes in the regions of the first set ($\{R_1, R_2, \ldots, R_n\}$). The regions in a set are members of a transaction. The value of **n** indicates the number of gaps (in windows) between these two sets. The value of **m** was defined before.

The algorithm for finding one-set and two-set frequent subsequences is based on the Apriori algorithm, which has two phases. In the first phase, it finds frequent one- and two-set subsequences in a set of transactions. In the second phase, it finds the MBA/SPA rules using the set of frequent subsequences found in phase one. The MBA rules (denoted as X → Y) are generated with the frequent one-set subsequences that have at least two items, whereas the SPA rules (denoted as X ➔ Y) are generated with the frequent two-set subsequences. The algorithm for the first phase is as follows:

Step 1. Generate **n**, 1-set candidate subsequences of a single transaction {e1}, {e2}, …, {en}.
Step 2. Compute the support for each candidate subsequence of step 1.
Step 3. Select subsequences of step 2 that have a support greater than ***minsup***.
Step 4. Generate possible candidate subsequences of 2 transactions applying a merging procedure to the selected subsequences of step 3. Given two selected 1-transaction subsequences {ei} and {ej}, for i<j, the possible 2-transaction subsequences are: {ei, ej}, [{ei}, {ej}], [{ej}, {ei}], [{ei}, {ei}], and [{ej}, {ej}]. The 1-set subsequence {ej, ei} is redundant .
Step 5. Valid candidate subsequences are selected using a pruning procedure described in [1].
Step 6. Select frequent 2-transaction subsequences with support greater than minsup.
Step 7. Repeat steps 4-6 until no more subsequences with higher than k scenes are found.

The algorithm for the second phase is as follows:

```
Input: Set of frequent subsequences, F, min_confidence value
For each frequent one-set subsequence A in F
    For each subset of regions S in A
        Compute confidence (S, A) = support(A) / support(S)
        If (confidence(S, A) > min_confidence) then
            Output the MBA rule S → A
For each frequent two-set subsequence A, B in F
    For each subset of regions S in A
        For each subset of regions T in B
            Compute confidence (S, S U T) = support(S U T) / support(S)
            If (confidence(S, S U T) > min_confidence) then
                Output the SPA rule S ➔[n]  T //n is the gap between A and B
```

The second algorithm was not optimized because the subsequences generated by the first algorithm only included two regions.

5 Results and Analysis

The results of running the program implementing the first algorithm generated the following one-set and two-set subsequences (we show only the top 20 for each set). All of the subsequences are 2-sequences. That is, they only have two regions on each subsequence. We could not find any 3-sequences with significant absolute support.

One-set frequent subsequences

{R045138,R046138}	131
{R045138,R047137}	85
{R045138,R063125}	84
{R046137,R047137}	73
{R042166,R045138}	61
{R045138,R046137}	61
{R041154,R045138}	59
{R063125,R068166}	58
{R034000,R045138}	55
{R036000,R045138}	54
{R045138,R046153}	53
{R041154,R063125}	51
{R045153,R068166}	49
{R045153,R063125}	48
{R026178,R045138}	47
{R036000,R063125}	46
{R052043,R063125}	45
{R046153,R063125}	45

```
{R033000,R045138}        44
{R033000,R063125}        43
```

Two-set frequent subsequences

```
{R046138}[1]{R045138}    128
{R046138}[4]{R045138}    128
{R046138}[16]{R045138}   127
{R046138}[7]{R045138}    125
{R046138}[24]{R045138}   123
{R046138}[6]{R045138}    122
{R046138}[12]{R045138}   122
{R046138}[9]{R045138}    121
{R046138}[11]{R045138}   121
{R046138}[21]{R045138}   121
{R046138}[22]{R045138}   121
{R046138}[0]{R045138}    120
{R046138}[8]{R045138}    120
{R046138}[19]{R045138}   120
{R046138}[2]{R045138}    119
{R046138}[10]{R045138}   119
{R046138}[15]{R045138}   118
{R046138}[18]{R045138}   118
{R046138}[5]{R045138}    116
{R046138}[23]{R045138}   116
```

The program implementing the algorithm for phase two generated the following MBA and SPA rules. We only show the top 10 MBA rules and the top 16 SPA rules with their confidence values. The maximum gap used for the SPA rules was 119 time windows which corresponds to a region(s) on the left side of the rule with at least one earthquake that happened one month apart of the region(s) on the right of the rule.

MBA rules:

```
{R046138} → {R045138,R046138}   confidence: 0.350267
{R068167} → {R068166,R068167}   confidence: 0.342520
{R044139} → {R044139,R045138}   confidence: 0.27193
{R048136} → {R047137,R048136}   confidence: 0.270270
{R051136} → {R047137,R051136}   confidence: 0.198198
{R048136} → {R045138,R048136}   confidence: 0.198198
{R046137} → {R046137,R047137}   confidence: 0.192105
{R047137} → {R045138,R047137}   confidence: 0.174897
{R068166} → {R068166,R068167}   confidence: 0.170254
{R048136} → {R048136,R051136}   confidence: 0.166667
```

SPA rules:

```
{R046138} →[1]{R045138}         confidence: 0.342246
```

```
{R046138} ➔[4]{R045138}          confidence: 0.342246
{R046138} ➔[16]{R045138}         confidence: 0.339572
{R046138} ➔[7]{R045138}          confidence: 0.334225
{R046138} ➔[24]{R045138}         confidence: 0.328877
{R046138} ➔[6]{R045138}          confidence: 0.326203
{R046138} ➔[12]{R045138}         confidence: 0.326203
{R046138} ➔[9]{R045138}          confidence: 0.323529
{R046138} ➔[11]{R045138}         confidence: 0.323529
{R046138} ➔[21]{R045138}         confidence: 0.323529
{R046138} ➔[22]{R045138}         confidence: 0.323529
{R046138} ➔[32]{R045138}         confidence: 0.323529
{R046138} ➔[83]{R045138}         confidence: 0.320856
{R046138} ➔[0]{R045138}          confidence: 0.320856
{R046138} ➔[8]{R045138}          confidence: 0.320856
{R046138} ➔[19]{R045138}         confidence: 0.320856
```

Their confidence is well below the usual range of 0.80 – 1.0 for confidence values used by data miners. SPA rules show higher confidence values which show in most cases the aftershocks that happened in nearby regions. All of the SPA rules show the associations of the same two neighbor regions with different gap values.

6 Conclusions

The low confidence of most of the MBA rules shows that there is no correlation among the earthquakes that occur in the same time window. Higher correlations are shown for SPA rules, which indicates that nearby regions may have had earthquakes that happened closer in time with a major earthquake. It is just a coincidence that several earthquakes occurred on different regions of the globe a few weeks or even a few days apart. The local newspaper in Rapid City (Rapid City Journal) published an article in the March 9th edition where it wrote that the seismologist Bernard Doft of the Royal Netherlands Metereological Institute said that "there is no direct connection between the Haiti and Chile earthquake. These countries are on separate earth's fault lines."

Most of the research on data mining use synthetic data. We used the earthquake dataset because it is a real dataset with a well-defined distribution (Figure 2). However, the results show that having a good data distribution does not guarantee good results. Our future work includes the analysis of satellite image requests to the EROS Data Center at Sioux Falls, South Dakota. They would like to apply data mining for improving the prefetching of scenes from their storage system. We are planning to analyze spatial and temporal relationship in the user requests.

7 Acknowledgements

We gratefully acknowledge the support from the Earth Resources Observation and Science (EROS) Center of U.S. Geological Survey, the U.S. National Science Foundation under Grants No. CNS- 0915762 and the Nelson Research Grant under Grants No. 03988.

REFERENCES

[1] Zaki, Mohammed J. 2001. SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning Journal*, 42, 31-60 (2001).

[2] Agrawal, R.; Imielinski, T. and Swami, A. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207-216, Washington, DC, May 26-28 1993.

[3] Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago, Chile, August 29-September 1 1994.

[4] Savasere,A .; Omiecinski, E.; and Navathe, S. 1995. An efficient algorithm for mining association rules in large databases. In *Proceedings of the 21st VLDB Conference*, Zurich, Swizerland.

[5] Agrawal, R. and Srikant, R. 1995.Mining sequential patterns. In *Proceedings of the 1995 Intl. Conference on Data Engineering (ICDE'95)*, pages 3-14, Taipei, Taiwan.

[6] Srikant, R. and Agrawal, R. 1996. Mining Sequential Patterns: Generalizations and performance improvements. In Proceedings of the 5th Intl. Conference Extending Database Technology.

[7] Pei, J.; Han, J.; Martazavi-Asi, B. and Pinto, H. 2001. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In *Proceedings of the Intel. Conference on Data Engineering (ICDE 2001)*, pages 215-226, Heidelberg, Germany, April 2001.

[8] Bayardo Jr., Roberto J. 1998. Efficently Mining Long Patterns from Databases. In Proceedings of the ACM SIGMOD Intl. Conference on Management of Data, pages 85-93, Seattle, WA, June 1998.

[9] Ayre, J.; Gehrke, J.; Yiu, T.; and Flannick, J. 2002. Sequential Pattern Mining using A Bitmap Representation. In Proceedings of the ACM SIGKDD '02 Conference, pages 429-435, Edmonton, Alberta, Canada.