UNIVERSITÉ ANTONINE
Faculté d’ingénieurs en Informatique,
Multimédia, Réseaux & Télécommunications

Data Mining – Sequence Clustering

Matière : System décisionnel

Effectué par :	NOM Prénom	INF#	Option
	MATTA Elie et al.	Privacy applied	OGL

TABLE OF CONTENTS

Preface

You are a marketing manager of a popular online retailer site. You sell various categories of products, including books, magazines, electronics, cookware, office products, and so on.

Every day, thousands of Web customers come to your site, navigating among different domains of your portal. In a physical shop, you can visually identify those departments and products that attract most customers and the customer interactions on various products.

In a virtual store, you don't see your customers. However, you still want to learn more about your customers to provide them with better services. You want to find out how your customers are using your site and the list of products in which they have shown interest. You also want to know the natural groups among these customers, based on their navigation patterns.

For example, one group of customers shops all sorts of products from your Web site, while others visit only certain categories of books and magazines. This information not only gives you a clear picture of your customer's behaviors in your virtual shop but also allows you to provide personalized shopping guidance to each customer, based on his or her profile.

It's about to analyze navigation sequences and organize sequences into natural groups based on their similarities, using the Sequence Clustering algorithm.

Introduction

I. General view of Data Mining

1) What is data mining?

Data mining is the process of exploring large quantities of data in order to discover meaningful information about the data, in the form of models and rules.

Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified.
Technically, data mining is the process of finding correlations or models among dozens of fields in large relational databases.

Data mining, *the extraction of hidden analytical information from large databases*, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools **predict** future directions and behaviors, allowing businesses to make proactive, knowledge-driven decisions.

Data mining is about analyzing data and finding hidden patterns. During the past decade, large volumes of data have been accumulated and stored in databases. This data comes from business software, such as financial applications, Enterprise Resource Management, Customer Relationship Management, and Web logs. The result of this data collection is that organizations have become “data-rich and knowledge-poor”. The collections of data have become so vast and are increasing so rapidly in size that the practical use of these stores of data has become limited. The main purpose of data mining is to extract patterns from the data, increase its essential value and transfer the data to knowledge.

Data mining can:

- Predict future results based on input.
- Find relationships between different data elements.
- Cluster values to groupings that can be fed back into the data warehouse.

2) What is SQL Server Data Mining?

SQL Server Data Mining is a collection of machine learning algorithms that explore your data for patterns. Once discovered, these patterns can be browsed for greater insight into your data, or they can be applied to new data to create "predictions" - which allow you to determine unknown facts about data based on data the algorithms have seen before.

Typical data warehousing implementations in organizations will allow users to ask and answer questions such as “How many sales were made, by region, by sales person between the months of May and June in 1999?”

Data mining will allow business decision makers to ask and answer questions, such as “Who is my core customer that purchases a particular product we sell?” or “Geographically, how well would a line of products sell in a particular region and who would purchase them, given the sale of similar products in that region?”.

Through the usage of tools like SQL Server 2000 Analysis Services and methodologies such as data warehousing implementations within the Microsoft Data Warehousing Framework, data mining can successfully be implemented as a next step towards uncovering and discovering essential business decision data.

SQL Server Data Mining comes with nine algorithms, plus all of the tools necessary to create, explore and deploy mining models in any enterprise or business applications.

3) What Data Mining Algorithms are available in SQL Server?

The following table represents a definition of the different algorithms available in SQL Server.

Decision Trees	Decision trees algorithm calculates the probability of a result based on values in a training set. For example, a person in the age group 20-30 that makes over $60,000/year and owns a home is more probable to need a lawn service than someone in the age group of 15-19 who doesn't own a home. Based on age, income, and home ownership, Decision trees algorithm can calculate the probability of that person needing a lawn service based on historical values.
Association Rules	Association rules algorithm helps identify relationships between various elements. For example, it's used in cross-selling solutions because it notes relationships between items. It can be used to predict what else someone buying a product will also be interested in purchasing. It can handle incredibly large catalogs, having been tested on catalogs of over half a million items.
Clustering	Classifies cases into distinct groups based on an attribute cells: it is a tool for data analysis, which solves classification problems. Its object is to distribute cases (people, objects, events etc.) into groups, so that the degree of association to be strong between members of the same cluster and weak between members of different clusters. This way each cluster describes, in terms of data collected, the class to which its members belong. Clustering is discovery tool. It may reveal associations and structure in data which, though not previously evident, nevertheless are sensible and useful once found.
Naïve Bayes	Naive bayes algorithm is used to clearly show the differences in a particular variable for various data elements. For example, the Household Income variable differs for each customer in the database, and can be used as a predictor of future purchasing. This model is good at showing the differences between certain groups such as customers who churn (mix) and those who don't.
Sequence clustering	Sequence clustering algorithm is a sequence analysis algorithm provided by Microsoft SQL Server Analysis Services. We can use this algorithm to explore data that contains events that can be linked by following paths, or sequences. The algorithm finds the most common sequences by grouping, or clustering, sequences that are identical. The following are some examples of sequences: • Data that describes the click paths that are created when users navigate or browse a Web site.

	• Data that describes the order in which a customer adds items to a shopping cart at an online retailer.
Time Series	Time series algorithm is used to analyze and forecast time-based data. Sales are the most generally analyzed and predicted data using Time series algorithm. This algorithm looks for patterns across multiple data series so that businesses can determine how different elements affect the analyzed series.
Neural Networks	Neural networks are the core of artificial intelligence. They seek to uncover relationships in data that other algorithms miss. While the Neural Nets algorithm tends to be slower than the other algorithms, it finds relationships that may be non-intuitive.
Linear Regression	Determines the relationship between columns in order to predict an outcome. Linear regression is applicable to numerous data mining situations. Examples are: predicting customer activity on credit cards from demographics and historical activity patterns, predicting the time to failure of equipment based on utilization and environment conditions, predicting expenditures on vacation travel based on historical frequent flier data, predicting staffing requirements at help desks based on historical data and product and sales information, predicting sales from cross selling of products from historical information and predicting the impact of discounts on sales in retail outlets.
Logistic Regression	Logistic regression is one of the most commonly-used statistical techniques. It is used with data in which there is a binary (success-failure) outcome variable, or where the outcome takes the form of a binomial proportion. It determines the relationship between columns in order to evaluate the probability that a column will contain a specific state. In logistic regression, however, one estimates the probability that the outcome variable assumes a certain value, rather than estimating the value itself. The concepts of "odds" and "odds ratio" are examined, as well as "risk ratio" and the difference between the two statistics.

Table 1 the nine algorithms for data mining.

II. General view of Sequence Clustering Algorithm

As the name suggests, Sequence Clustering algorithm is a mix of sequence and clustering techniques. It's designed to analyze a population of cases that contains sequence data and to group those cases into more or less homogeneous segments based on the similarity of those sequences. So Sequence clustering is a data mining technique that takes a number of sequences and groups them in clusters so that each cluster contains similar sequences.

A *sequence* is a series of distinct events (states). Usually the number of discrete states in a sequence is finite.
Sequence data is everywhere in the real world. Lots of information is encoded in sequence form.

For example, a DNA sequence is a series of four discrete states: A (adenosine), G (guanine), C (cytosine), and T (thymidine). The list of courses a student takes at a university forms a sequence. The series of URL clicks of a Web user is a sequence. In a shopping basket example, if we don't care about the order of the product purchases, the business problem of market basket analysis is an association task. If we do care about the order of the product purchases, the purchase data forms a sequence, and this problem is a sequence task.

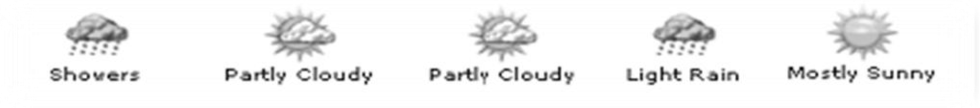


Figure 1 displays a weather forecast sequence.

➢ So what is the concept of the sequence clustering algorithm?
Well it is Markov chain. And now let's take a look at it.

Sequence Clustering Algorithm Principles

As we said before sequence clustering works by merging two technologies, clustering and sequence analysis. The sequence analysis is a Markov chain model.

In Sequence clustering algorithm each case is assigned to each cluster with some probability. Each cluster has an associated Markov chain. So…

I. What is a Markov chain?

A Markov chain named after the mathematics André Markov is a stochastic process with Markov property. Markov chains represent a sequence of multiple variables which the future variable is determined by the present variable. We describe a Markov chain as follows: We have a set of *states*, $S = \{S_1, S_2..., S_n\}$.

Most states emit events; other states like Begin and End are silent.

The process starts in one of these states and moves successively from one state to another. Each move is called a *step*. If the chain is currently in state *Si*, then it moves to state S*j* at the next step with a probability denoted by P*ij*, and this probability does not depend upon which states the chain was in before the current state. The probabilities P*ij* are called *transition probabilities*. Figure 2 shows a Markov chain.

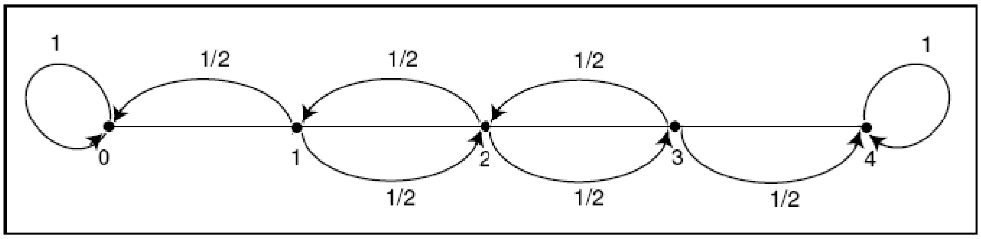


Figure 2 Markov chain.

A Markov chain can also be represented by a state transition matrix as the example shown below:

According to Kemeny, Snell, and Thompson, the Land of Oz is blessed by many things, but not by good weather. They never have two nice days in a row. If they have a nice day, they are just as likely to have snow as rain the next day. If they have snow or rain, they have an even chance of having the same the next day. If there is change from snow or rain, only half of the time is this a change to a nice day. With this information we form a Markov chain as follows. We take as states the kinds of weather R, N, and S. From the above information we determine the transition probabilities. These are most conveniently represented in a square array as:

$$\mathbf{P} = \begin{array}{c} \\ \mathrm{R} \\ \mathrm{N} \\ \mathrm{S} \end{array} \begin{array}{c} \begin{array}{ccc} \mathrm{R} & \mathrm{N} & \mathrm{S} \end{array} \\ \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array} .$$

Table 2 State transition matrix for states {rainy, nice, snowy}

II. Transition Matrix

Looking at the matrix **P** in the example, the entries in the first row represent the probabilities for the various kinds of weather following a rainy day. Similarly, the entries in the second and third rows represent the probabilities for the various kinds of weather following nice and snowy days, respectively. Such a square array is called the *matrix of transition probabilities*, or the *transition matrix.*

We consider the question of determining the probability that, given the chain is in state *i* today; it will be in state *j* two days from now. We denote this probability by *pij*.

In the example, we see that if it is rainy today then the event that it is snowy two days from now is the disjoint union of the following three events:

1) It is rainy tomorrow and snowy two days from now.

2) It is nice tomorrow and snowy two days from now.

3) It is snowy tomorrow and snowy two days from now.

The probability of the first of these events is the product of the conditional probability that it is rainy tomorrow, given that it is rainy today, and the conditional probability that it is snowy two days from now, given that it is rainy tomorrow. Using the transition matrix **P**, we can write this product as $p11p13$. The other events also have probabilities that can be written as products of entries of **P**.

Thus, we have $p13 = p11p13 + p12p23 + p13p33$:

This equation should remind the reader of a dot product of two vectors; we are Dotting the first row of **P** with the third column of **P**. This is just what is done in obtaining the 1*;* 3-entry of the product of **P** with itself. In general, if a Markov chain has *r* states, then

$$P_{ij} = \sum_{k=1}^{r} P_{ik} P_{kj}$$

The following general theorem is easy to prove by using the above observation and induction.

The Theorem let **P** be the transition matrix of a Markov chain. The **ijth** entry $\boldsymbol{P_{ij}^{(n)}}$ of the matrix $\mathbf{P}^{n}$ gives the probability that the Markov chain, starting in state Si, will be in state Sj after n steps.

III. The Hidden Markov Model

The difference between a Hidden Markov Model (HMM) and a normal Markov model is that the state sequence of the model is hidden.

We only know the observed sequence of outputs. There are five attributes of a **HMM:**

- The set of states.
- The output alphabet $\{O_1, O_2, \ldots, O_n\}$
- The probabilities of initial states at t_0
- The state transition probabilities
- The output probabilities of each given state

HMM is used in many applications from voice recognition to DNA analysis.

Sequence Cluster algorithm is based on an observable Markov chain, not on HMM.

Example:

We have *n* biased (unfair) coins (the coins are the states of the HMM), the output alphabet is {*H*, *T*}. We know the transition probabilities among these coins and the output probabilities of *H* and *T* for each coin. We also know the initial probabilities of the coins to flip. But we don't know exactly which coin is used to produce the output at each step, because the state sequence is hidden from us. Based on the sequence of observed outputs, we can figure out the following questions:

- What is the probability of observed outputs $\{O_1, O_2, O_T\}$ given the model, example $P(O_1, O_2, O_T| \quad model)$?
- At each step, what state is most likely given the model and outputs?
- Given an HMM structure and observed data, find the model parameters that maximize $P(O_1, O_2, O_T|model)$.

IV. Order of a Markov chain

The order is one of the important properties of a Markov chain. The order n specifies the probability of a state depending on the previous n states.
The most common chain is the 1st order, which means the probability of each state $\mathbf{x_i}$ depends only on the state of x_{i-1}.

To remember the previous **n** states we can build high order Markov chains using more memory.

Example:

Figure 3 illustrates an example of a Markov chain of the DNA sequence.
The transitions emanating (coming) from a given state define a distribution over the possible next states.

$$P(xi = G|xi\text{–}1 = A) = 0.15$$

Means given the current state *A*, the probability of next state being *G* is 0.15.

Transition probabilities:
P (xi = G|xi-1=A) = 0.15
P (xi = C|xi-1=A) = 0.15
P (xi = T|xi-1=A) = 0.25
P (xi = A|xi-1=A) = 0.35
P (xi = End|xi-1=A) = 0.10

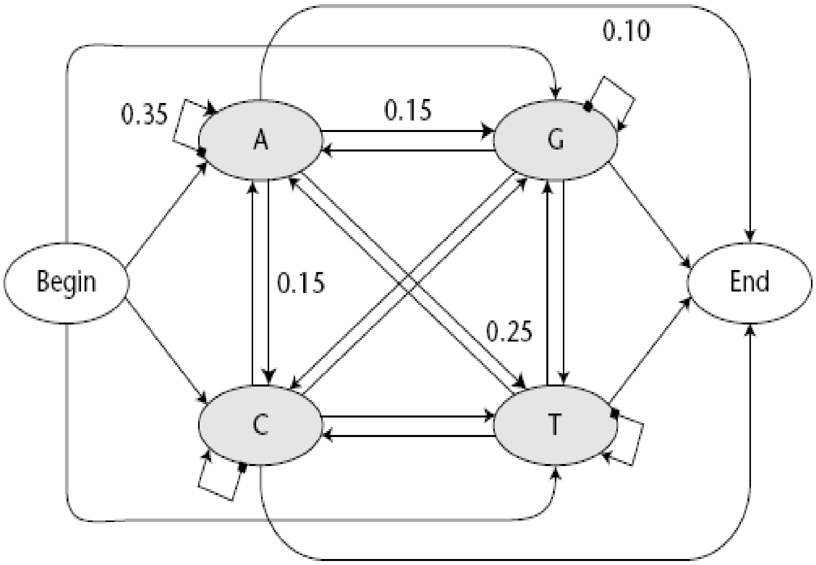


Figure 3 Markov chain model.

An n^{th}-order Markov chain over k states is equivalent to a first order Markov chain over k^n states.

For example, a 2nd- order Markov model for DNA can be treated as a 1st-order Markov model over the following states: AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, and TT. The total number of states is 4^2. The higher the order of a Markov chain, the more memory and time required for the processing.

Based on the Markov chain, for any given length L sequence x $\{x_1, x_2, x_3, x_L\}$, we can calculate the probability of a sequence as follows:

$$
\begin{aligned}
P(x) &= P(x_L \,.\, x_{L-1},\ldots,x_1) \\
&= P(x_L|\, x_{L-1},\ldots,x_1)P\,(x_{L-1}|x_{L-2},\ldots,x_1)\ldots P(x_1)
\end{aligned}
$$

In the case of a 1st-order Markov chain, the probability of each x_i depends only on x_{i-1}, the preceding formula is equivalent to the following:

$$
\begin{aligned}
P(x) &= P(x_L \,.\, x_{L-1},\ldots,x_1) \\
&= P(x_L|x_{L-1})P(x_{L-1}|x_{L-2})\ldots P(x_2|x_1)P(x_1)
\end{aligned}
$$

Clustering with Markov Chain

Sequence clustering is a data mining technique that puts in one cluster all similar sequences. To group sequences in clusters, we use many algorithms which can be helpful in choosing characteristics for each cluster.

One of those algorithms is The Sequence clustering algorithm, where each case is assigned to a cluster with some probability. And a Markov chain is associated to each cluster. Usually clustering algorithm uses n-order Markov chains, and not hidden Markov chain.

Sequence clustering algorithm involves the following steps:

- Initialize the model parameters at random. In fact, a state transition probability of Markov chain must be initialized from each cluster randomly.
- Assign each case to each cluster with some probability.
- Recalculate the whole parameters of each cluster by calculating the probability of each state transition of Markov chain in that cluster considering the probability of each case.
- After reconsidering the different probabilities, the model must be converged; else we must restart work from step 2.

Each model must contain at least one table with the non-sequence attributes, but not more than one nested table of the first, with sequence key.

We will include an example which shows exactly those tables as shown in figure 4:

A company who has a portal Web Site which contains the following domains: news, weather, money, mail… we can create 2 tables:

1- The non-sequence attributes table which contains the user-id and his information.

2- The nested table of the first which contains the domains visited by each user (including the user-id column) with sequence key for each domain entered.

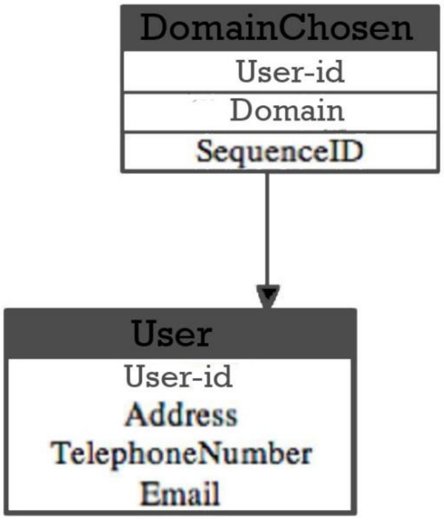


Figure 4 DomainChosen and User tables.

I. Cluster Decomposition

The difference between a sequence clustering and a normal clustering model is the number of natural groups.

1) Normal clustering model

We build a clustering model in less than 10 groups (k<10) because when the number of clusters is too large it will be difficult to deduce the final result.

If we have a large number of distinct groups exist, we build clustering models in multiple steps, and for each step they break the population into a handful groups.

2) Sequence clustering model

When we have a large number of states in sequence it will be many distinct clusters.
For example, a Web navigation scenario has 60 URL categories in a portal site.
The first group of Web customers navigates in news, the second in music and movies and the third group is interested in front pages and weather.

While clustering the customers, we will have a large number of clusters, compared to a non-sequence cluster model. It is easy to deduce these models based on their sequences of states.

If a small number of clusters are specified by a user and there are different types of sequences in a cluster, the algorithm will decompose the cluster into multiple clusters.

For example, if a cluster contains two sets of sequences:

Movie ➪ Music ➪ Download

New ➪ News ➪Weather

The algorithm breaks it into two clusters at the final stage of the model processing.

Algorithm Parameters

Sequence clustering algorithm doesn't have so much parameters, they are few. These parameters are used to control the cluster count, sequence states, etc...

I. Clustering_Method including k-means and EM algorithms

The Clustering_Method indicates which algorithm is used to determine cluster membership. The vanilla versions of each algorithm eschew the scalable framework described previously and operate only on one sample of the data. The possible values for this parameter are:

1 — Scalable EM (default) [1]

2 — Vanilla (non-scalable) EM [2]

3— Scalable K-means [3]

4 — Vanilla (non-scalable) K-means [4]

1) EM Clustering

In EM clustering, the algorithm iteratively refines an initial cluster model to fit the data and determines the probability that a data point exists in a cluster. The algorithm ends the process when the probabilistic model fits the data. The function used to determine the fit is the log-likelihood of the data given the model.

If empty clusters are generated during the process, or if the membership of one or more of the clusters falls below a given threshold, the clusters with low populations are reseeded at new points and the EM algorithm is rerun.

The results of the EM clustering method are probabilistic. This means that every data point belongs to all clusters, but each assignment of a data point to a cluster has a different probability. Because the method allows for clusters to overlap, the sum of items in all the clusters may exceed the total items in the training set. In the mining model results, scores that indicate support are adjusted to account for this.

This algorithm is used as the default because it offers multiple advantages in comparison to k-means clustering:

- Requires one database scan, at most.
- Will work despite limited memory (RAM).
- Has the ability to use a forward-only cursor.
- Outperforms sampling approaches.

The Microsoft implementation provides two options: **scalable** and **non-scalable EM**. By default, in scalable EM [1], the first 50,000 records are used to seed the initial scan. If this is successful, the model uses this data only. If the model cannot be fit using 50,000 records, an additional 50,000 records are read. In non-scalable EM [2], the entire dataset is read regardless of its size. This method might create more accurate clusters, but the memory requirements can be significant. Because scalable EM operates on a local buffer, iterating through the data is much faster, and the algorithm makes much better use of the CPU memory cache than non-scalable EM. Moreover, scalable EM is three times faster than non-scalable EM, even if all the data can fit in main memory. In the majority of cases, the performance improvement does not lead to lower quality of the complete model.

2) K-Means Clustering

K-means clustering [3] is a well-known method of assigning cluster membership by minimizing the differences among items in a cluster while maximizing the distance between clusters. The "means" in k-means refers to the *centroid* of the cluster, which is a data point that is chosen arbitrarily and then refined iteratively until it represents the true mean of all data points in the cluster. The "k" refers to an arbitrary number of points that are used to seed the clustering process. The k-means algorithm calculates the squared Euclidean distances between data records in a cluster and the vector that represents the cluster mean, and converges on a final set of k clusters when that sum reaches its minimum value.

The k-means algorithm assigns each data point to exactly one cluster, and does not allow for uncertainty in membership. Membership in a cluster is expressed as a distance from the centroid.

Typically, the k-means algorithm is used for creating clusters of continuous attributes, where calculating distance to a mean is straightforward. However, the Microsoft implementation adapts the k-means method to cluster discrete attributes, by using probabilities. For discrete attributes, the distance of a data point from a particular cluster is calculated as follows:

1 - P(data point, cluster)

The k-means algorithm provides two methods of sampling the data set: non-scalable K-means [4], which loads the entire data set and makes one clustering pass, or scalable k-means, where the algorithm uses the first 50,000 cases and reads more cases only if it needs more data to achieve a good fit of model to data.

II. List of the algorithm parameters

1) Cluster_Count:

The definition of Cluster_Count in Sequence Clustering algorithm is the same as in Clustering algorithm:
* It defines the number of clusters (groups) that a model contains.
* In practice, the more attributes we have, the more clusters we need to describe our data correctly.
* If we have too many attributes, we may want to organize our data so that the number of clusters is reduced. Ex: The movie vendor: instead of clustering by the individual movies that the customers watched, we can cluster by the *genres* of those movies. This technique reduces the attribute cardinality and creates much more meaningful models.
* If we set this value to 0 the algorithm will automatically choose the best number of clusters for predictive purpose.
* The default value is 10.

2) Minimum_Support:

The definition of Minimum_Support in Sequence Clustering algorithm is the same as in Clustering algorithm.
* It is an integer.
* It specifies the minimum number of cases in each cluster to avoid having clusters with few cases. For example, for privacy reasons we don’t want to create clusters smaller than 10 people. Due to the flexibility of clustering we may have clusters reporting membership lower than this quantity after training. Setting this number too high can create bad results.
* The default value is 10.

3) Maximum_States:

The definition of Maximum_States is the same as in the Clustering algorithm.
* It is an integer.
* It defines the maximum number of different states an attribute can have.
* The default value is 100; attributes with more than 100 states invoke feature selection.

N.B: If the number of states for a *non-sequence* attribute is greater than the maximum number of states, the algorithm uses the attribute's most popular states and treats the remaining states as missing.

4) Maximum_Sequence_States:

* It is an integer.
* It specifies the maximum number of different states that a sequence attribute can have.
* Users can overwrite this value.
* If the sequence data has more states than Maximum_Sequence_States, feature selection is invoked, and the selection is based on the popularity of the states in the marginal model.
* The default value is 64.

TIP: Suppose that there are M distinct sequence states. Each cluster content contains an *M*M* matrix. The processing time is proportional to M^2. If *M* is large, it may take long time to process the model. Our recommendation is to make *M* no more than 100. If there are too many states, for example, hundreds of pages on your Web site, you can reduce *M* by grouping Web pages into categories.

III.Overall Optimized Performance

Here are some general ways to optimize processing using the above parameters mentioned above:

- Controlling the number of clusters generated, by setting a value for the CLUSTER_COUNT parameter.

- Reducing the number of sequences included as attributes, by increasing the value of the MINIMUM_SUPPORT parameter. As a result, rare sequences are eliminated.
- Reducing complexity before processing the model, by grouping related attributes.

In general, you can optimize the performance of an *n*-order Markov chain mode in several ways:

- Controlling the length of the possible sequences.
- Programmatically reducing the value of *n*.

IV. Modeling Flags

The following modeling flags are supported for use with the Microsoft Sequence Clustering algorithm.

- **NOT NULL**
  - Indicates that the column cannot contain a null. An error will result if Analysis Services encounters a null during model training.
  - Applies to the mining structure column.

- **MODEL_EXISTENCE_ONLY**
  - Means that the column will be treated as having two possible states: **Missing** and **Existing**. A null is treated as a **Missing** value.

Applies to the mining model column.

Using the sequence clustering algorithm

The Sequence Clustering algorithm can be useful in many ways such as **click stream analysis**, **customer purchase analysis**, **bioinformatics**, etc…
First of all let's see how to create a DMX query and then how to interpret the model using the Sequence Clustering viewer. To make everything clear we're supporting our project with the following example.

I. Creating DMX Queries:

Figure 5 displays two tables: Customer and ClickPath.
- Customer table contains customer profiles about Web usage.
- ClickPath is a transaction table. It contains three columns:
  * CustomerGuid is the foreign key to the Customer table.
  * SequenceID is an integer that determines the Web click sequence number 1, 2, 3 . . . *n*.
  * URLCategory is the state of the sequence.

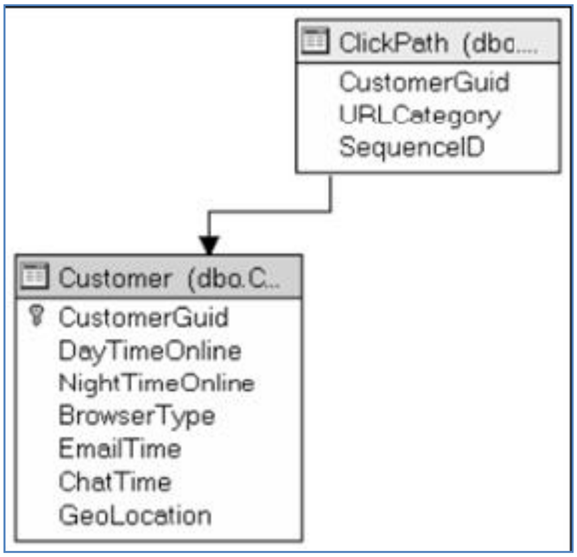


Figure 5 Customer and ClickPath tables.

In this model the sequence is the series of Web clicks on the URLCategory such as News ➪ News ➪ Sports ➪ News ➪Weather. The length of the sequence is variable from costumer to another.

1) Creating the model:

The following statement creates a mining model using the Sequence Clustering algorithm. **Sequence data must be stored in a nested table.**

The Microsoft Sequence Clustering algorithm doesn't support multiple sequence tables in a model and doesn't support more than one nonkey attribute in the sequence table.

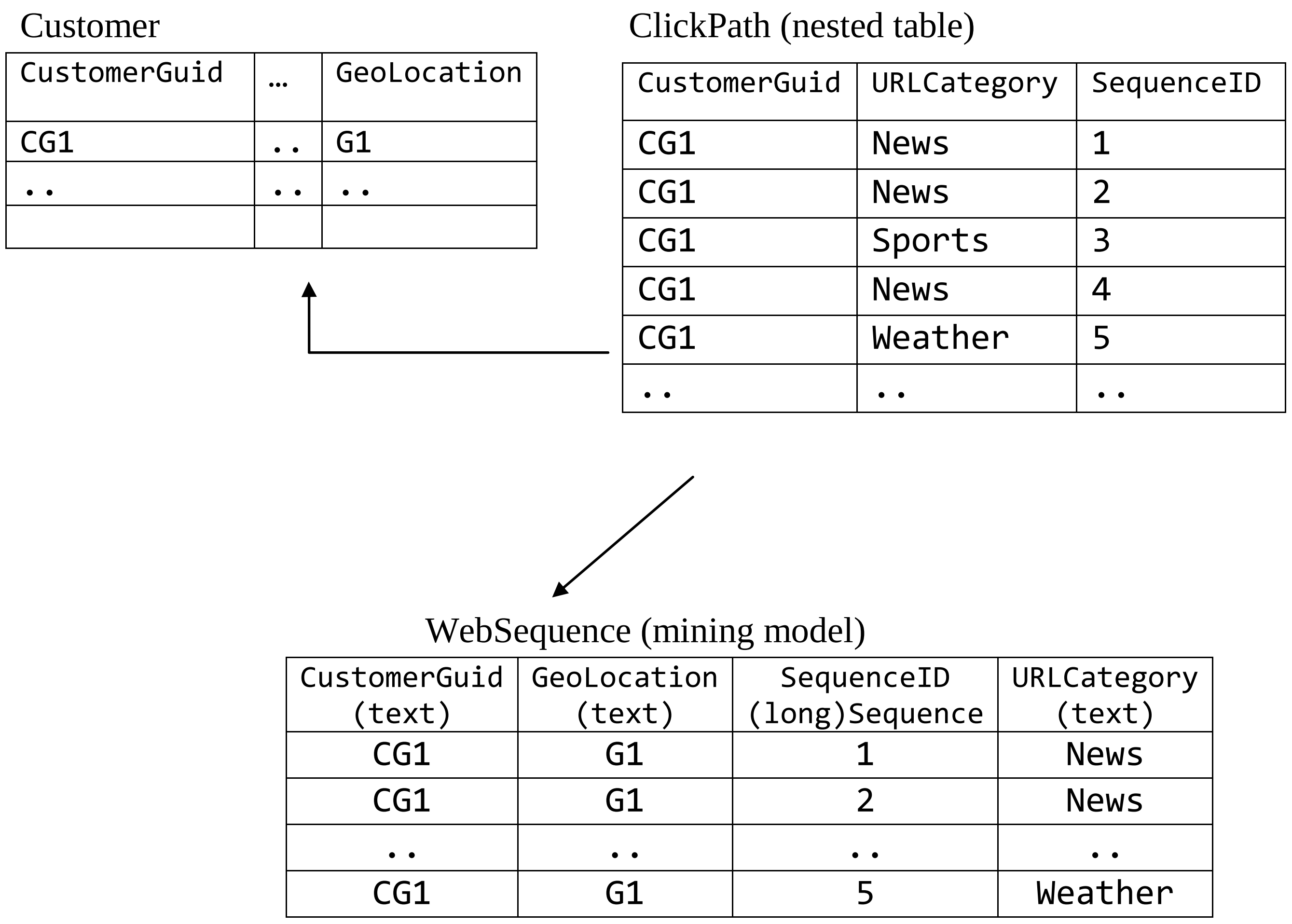
In this model the nested table, that contains the sequence data, is ClickPath and the nonkey attribute is "URLCategory".

```
Create mining model WebSequence (

 CustomerGuiId text key,
 GeoLocation text discrete,
 ClickPath table Predict (    // ClickPath is the nested table that contains
                                        //the sequence data
      SequenceID long key Sequence,
        URLCategory text,   //this is the nonkey attribute in the nested table.
     )
 )
Using Microsoft_SequenceClustering_Algorithm
```

The nested table, ClickPath, or the nonkey attribute in the nested table (URLCategory) may be specified as predictable.

In this case, when the model is processed, wee see customer segments based on their Web clicks and geolocations. We can also use the model to predict the next *n* sequence states for a given customer.



Customer

CustomerGuid	…	GeoLocation
CG1	..	G1
..	..	..

ClickPath (nested table)

CustomerGuid	URLCategory	SequenceID
CG1	News	1
CG1	News	2
CG1	Sports	3
CG1	News	4
CG1	Weather	5
..	..	..

WebSequence (mining model)

CustomerGuid (text)	GeoLocation (text)	SequenceID (long)Sequence	URLCategory (text)
CG1	G1	1	News
CG1	G1	2	News
..	..	..	..
CG1	G1	5	Weather

2) Charging the model:

The following Insert into statement trains the sequence model:

```
Insert into WebSequence

(      CustomerGuid, GeoLocation,
      ClickPath (SequenceID, URLCategory)
)

OPENROWSET (‘MSDataShape’, ‘data provider=Microsoft.Jet.OLEDB.4.0; data
source=C: \data\webclick.mdb’, ‘SHAPE
     {Select CustomerGuid, GeoLocation from Customer}
     Append (
          {Select CustomerGuid, SequenceID, URLCategory from ClickPath}
          Relate CustomerGuid To CustomerGuid
          )
     As ClickPath’
 )
```

3) Using Cluster function:

Sequence Clustering algorithm supports prediction. For cluster membership prediction, we can use the Cluster() function, which returns the cluster ID for each case.

The following query returns the cluster ID for each input case:

```
 SELECT t.CustomerGuid, Cluster ()
 From WebSequence PREDICTION JOIN

 SHAPE {
     OPENROWSET (‘SQLOLEDB.1’, ‘Integrated Security=SSPI; Initial
Catalog=Sequence; DataSource=localhost‘, ‘SELECT CustomerGuid,
GeoLocation FROM Customer ORDER BY CustomerGuid’)
     }

 APPEND ({
    OPENROWSET (‘SQLOLEDB.1’, ‘Integrated Security=SSPI; Persist
Security Info=False; Initial Catalog=Sequence; Data Source=localhost‘,
‘SELECT SequenceID, CustomerGuid, URLCategory FROM ClickPath ORDER BY
CustomerGuid’)
    }
    RELATE CustomerGuid TO CustomerGuid
    )

 AS ClickPath AS t

 ON
     WebSequence.CustomerGuid = t.CustomerGuid AND
     WebSequence.GeoLocation = t.GeoLocation AND
     WebSequence.ClickPath.URLCategory = t.ClickPath.URLCategory AND
     WebSequence.Click Path.SequenceID = t.ClickPath.SequenceID
```

The previous query returns a table of two columns: CustomerGuid and the predicted cluster number for each case.

NOTE: The Select query shown here uses the Shape provider from Analysis Services instead of the standard Microsoft Data Access Component (MDAC) Shape provider. The syntaxes of these two Shape providers are slightly different. The Shape provider of Analysis Services put the Shape command outside two Openrowset statements. It requires that both input rowsets be sorted on the same join key and in the same order. The Analysis Services Shape essentially does a merge operation and is much more scalable.

4) Using PredictSequence function:

Because the nested table ClickPath is predictable, it is possible to use the Sequence Clustering algorithm to predict the subsequent states of a given sequence.
There is a new prediction function called PredictSequence, which has the following syntax:

`PredictSequence(ClickPath)` (Returns the next predicted sequence state for a given sequence. The result is in a table form.)

`PredictSequence(ClickPath, 3)` (Returns the next three predicted sequence states for a given sequence. The result is in a table form.)

When the prediction returns a number of consequence steps, the probability of P_n is always less than P_{n-1}, where n is the step number. The formula to calculate of P_n is the following:

$$P_n = P_{n-1} * p\left(\frac{S_n}{S_{n-1}}\right)$$

Where $p\left(\frac{S_n}{S_{n-1}}\right)$ is the probability from state S_{n-1} to S_n in the closest cluster for the case. In our example URLCategory represents the state.

The following query predicts the next three steps for each customer.

```
Select CustomerId, PredictSequence (ClickPath, 2) as Sequences
From WebSequence Prediction Join ...
…
```

It returns the results shown in Table 3. The predicted sequence states are stored in a nested table.

There are three columns in the nested table:

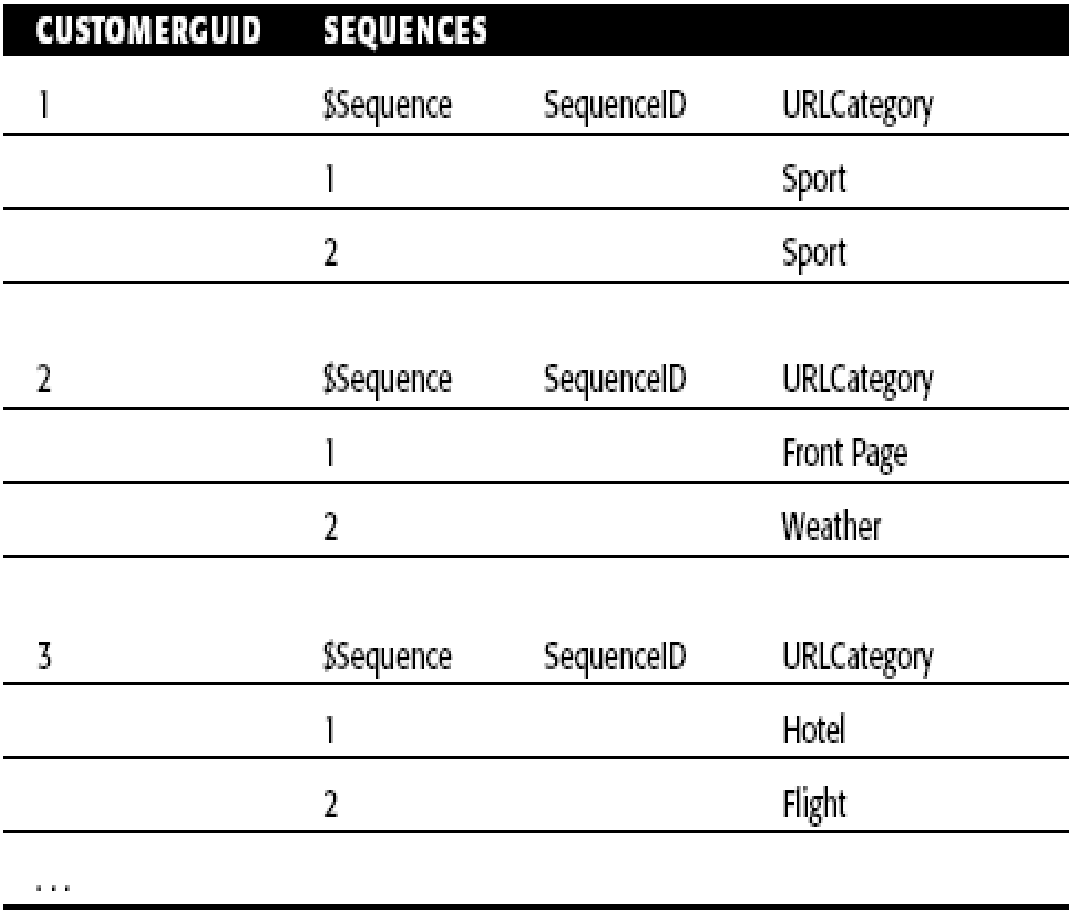
i) $Sequence is the generated column:

* It is an integer.
* It indicates the future steps 1, 2, 3 . . . . "1" means the next step.

ii) The Sequence ID

* It has the same data type as the sequence column.
* If the sequence key is date type, it returns the consequent dates.
* Sequence Clustering algorithm doesn't fill this column.

iii) URLCategory is the predicted state of the sequence.



CUSTOMERGUID	SEQUENCES		
1	$Sequence	SequenceID	URLCategory
	1		Sport
	2		Sport
2	$Sequence	SequenceID	URLCategory
	1		Front Page
	2		Weather
3	$Sequence	SequenceID	URLCategory
	1		Hotel
	2		Flight
. . .			

Table 3 Prediction query result with sequences.

We can also use a subselect statement on the nested table produced by PredictSequence. For example:

```
Select CustomerGuid, (Select $Sequence, URLCategory
          From PredictSequence(ClickPath, 2)) as Sequences
From WebSequence Predict Join ...
...
```

5) Using PredictProbability function:

To get the probability of each predicted sequence state, we can use the PredictProbability function:

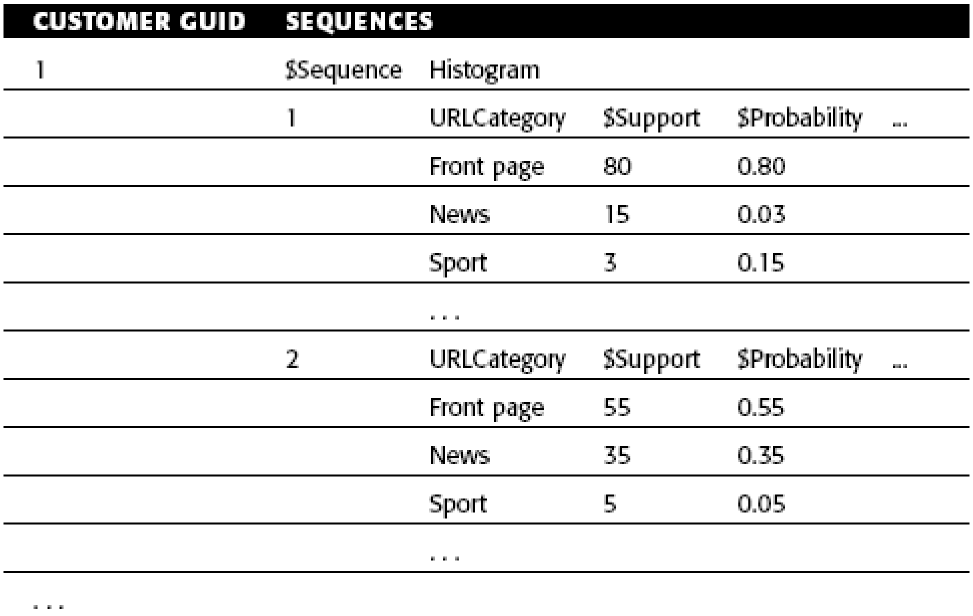
```
Select CustomerGuid, (Select $Sequence, URLCategory,
          PredictProbability (URLCategory)
    From PredictSequence (ClickPath, 2)) as Sequences
From WebSequence Predict Join ...
...
```

6) Using PredictHistogram function:

Sometimes, we want to have a histogram of the probability for each sequence state at each step. We can use the PredictHistogram function on the sequence state column. For example:

```
Select CustomerGuid, (Select $Sequence,
          PredictHistogram (URLCategory)as Histogram
    From PredictSequence(ClickPath, 2)) as Sequences
From WebSequence Predict Join ...
...
```

The result of this query contains two levels of nesting: one level is generated by PredictSequence, and another level is generated by PredictHistogram. The result format is displayed in Table 4.



CUSTOMER GUID	SEQUENCES				
1	$Sequence	Histogram			
	1	URLCategory	$Support	$Probability	...
		Front page	80	0.80	
		News	15	0.03	
		Sport	3	0.15	
		. . .			
	2	URLCategory	$Support	$Probability	...
		Front page	55	0.55	
		News	35	0.35	
		Sport	5	0.05	
		. . .			
. . .					

Table 4 Query Result with PredictHistogram Function.

In a Web click scenario, we know the Web visitor's navigation sequence within a session and we may want to predict his or her next few possible clicks in real time so that we can provide a personalized guide for the visitor. The click path is not yet recorded in database. In this case, we can use singleton query to make our prediction:

```
Select
    PredictSequence(ClickPath,3)
From
    [WebSequence]
Natural Prediction Join
    (Select (Select 1 As SequenceID,
         'Baseball' As URLCategory
       Union Select 2 As SequenceID,
    'Business' As URLCategory) As ClickPath) As t
```

II. Model content:

The content of a sequence clustering model is laid out in four levels, as illustrated in Figure 6.

The root node represents the model. The second level is the cluster level; each node except the last one represents a cluster discovered by the algorithm. The last node in the second level is a transition matrix, which represents the state transition probabilities of the overall population. The transition matrix has a set of children; each represents a row in the transition matrix. Due to content size, the matrix stores only those items with a probability greater than 0. Each cluster node also has a transition matrix as its child, which represents the transition probability of the given cluster. Therefore, there are four levels in the content of a sequence clustering model.

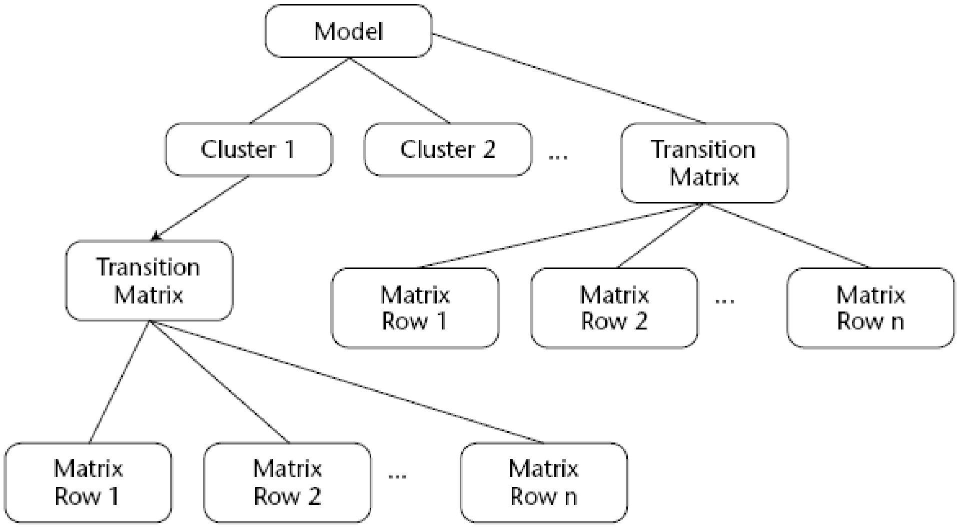


Figure 6 Content of Sequence Clustering model.

Explaining & understanding the model

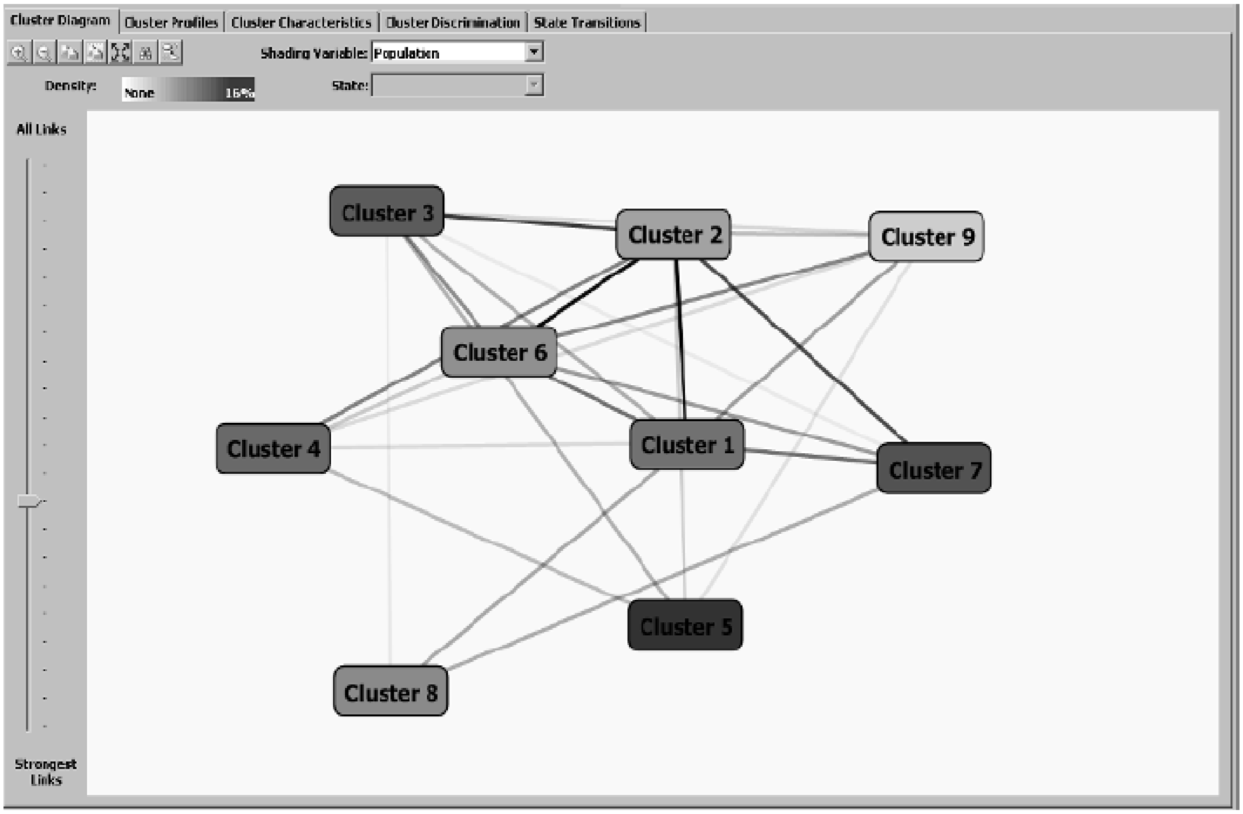
I. Cluster Diagram:

Figure 7 displays the Cluster Diagram pane. This tab is the same as in the Clustering viewer.

* Similar clusters (clusters with similar probability distributions, such as clusters 1, 5, and 7 in the figure) are closer to each other.

* The node background represents the size of the cluster. For example, Cluster 5 is a large cluster and Cluster 9 is much smaller.

* We can also use the node color-coding to represent other attribute values, including a sequence state, for example, Weather. The clusters with high probabilities of clicking on the Weather page are highlighted with a darker color.



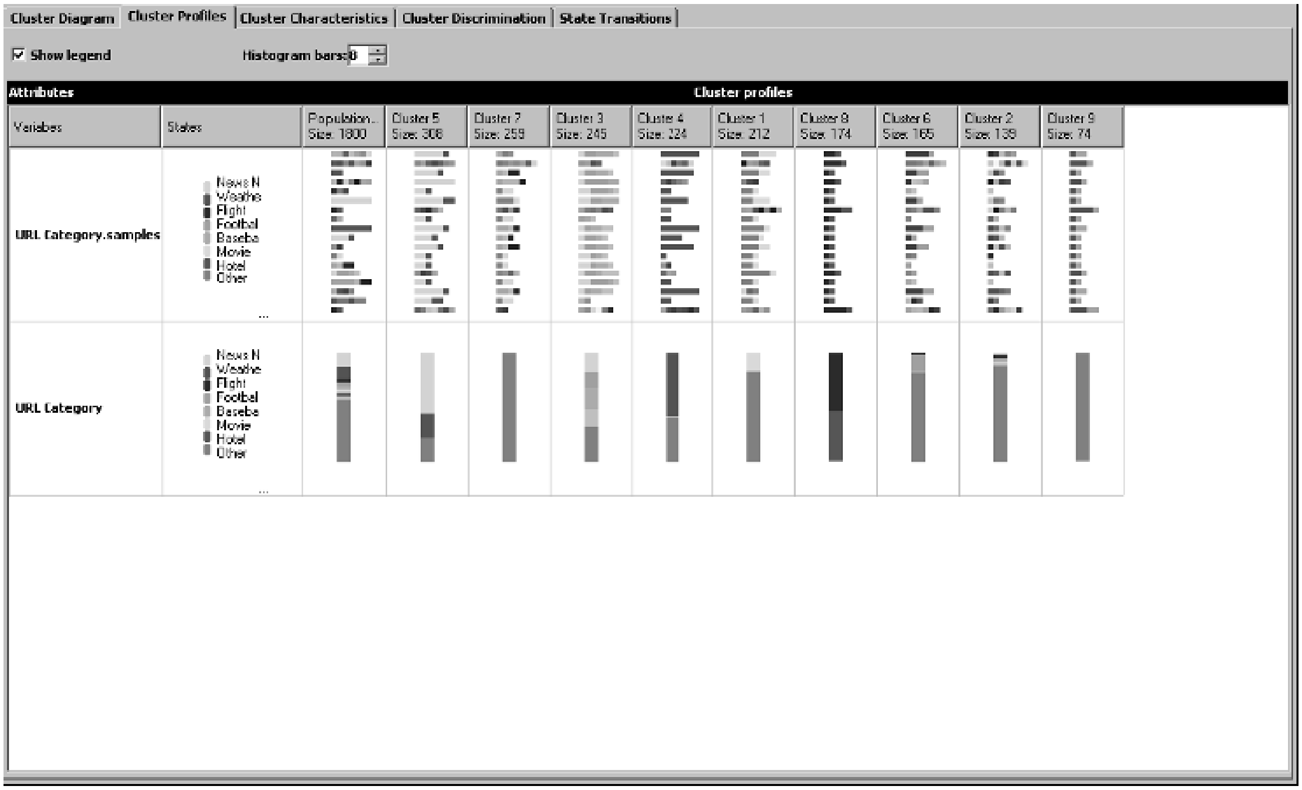
Shading Variable: Population

State:

Figure 7 Cluster Diagram.

II. Cluster Profiles:

Figure 8 displays the cluster profile.
* Each column represents a cluster.
* Each row represents an attribute.
* The URLCategory row represents the sequence attribute. Each cell in this row contains a histogram of sequences.
* Each line in the histogram represents a sample case in this cluster, and a line is composed of a series of sequence states.
* Each sequence cell displays about 20 cases.



☑ Show legend

Histogram bars: 8

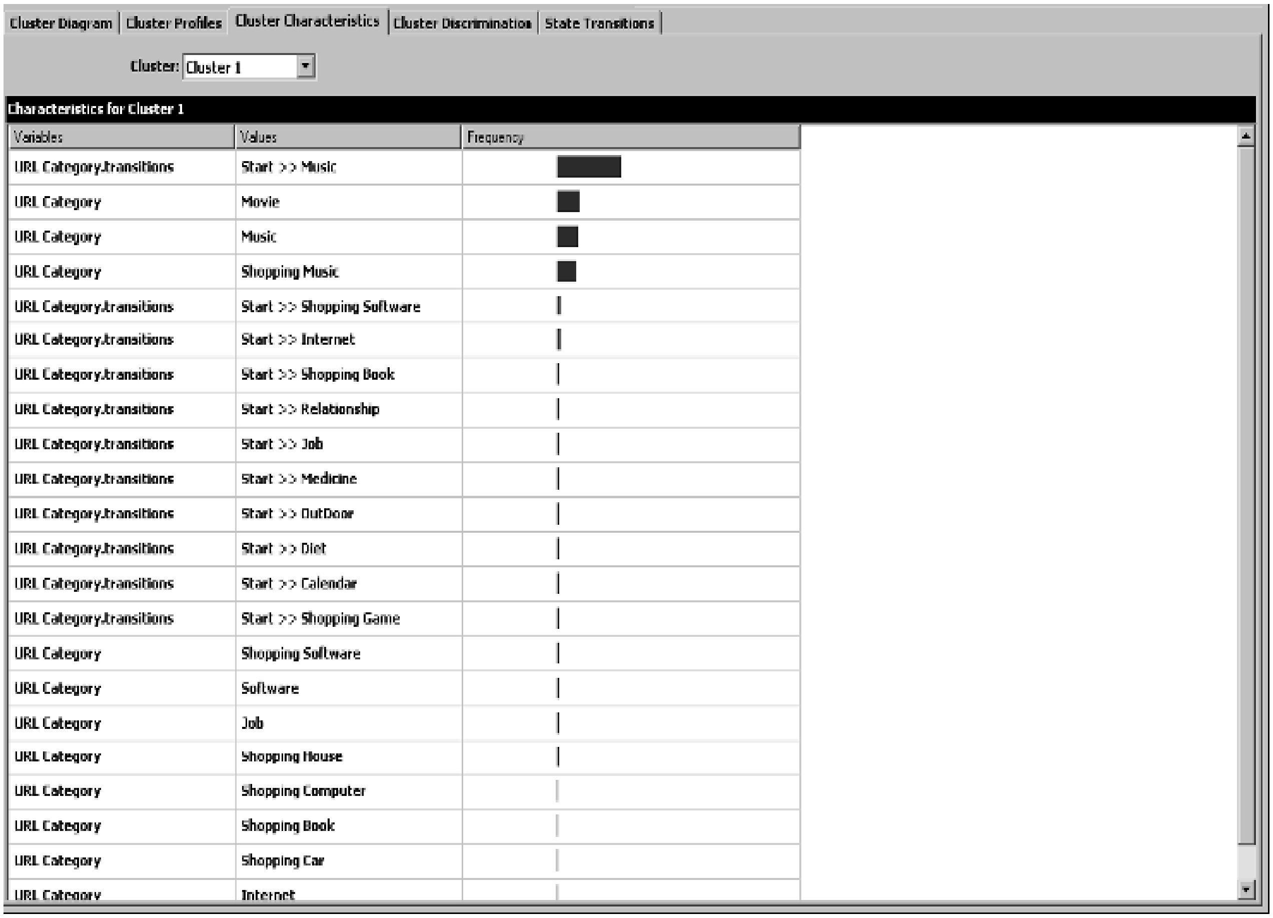
Attributes		Cluster profiles									
Variabes	States	Population... Size: 1800	Cluster 5 Size: 308	Cluster 7 Size: 259	Cluster 3 Size: 245	Cluste 4 Size: 224	Cluster 1 Size: 212	Cluster 8 Size: 174	Cluster 6 Size: 165	Cluster 2 Size: 139	Cluster 9 Size: 74
URL Category.samples	News N Weathe Flight Footbal Baseba Movie Hotel Other ...										
URL Category	News N Weathe Flight Footbal Baseba Movie Hotel Other ...										

Figure 8 Cluster profile.

III. Cluster characteristics:

Figure 9 displays the characteristics of each cluster.
* Each row represents the probability (frequency) of an attribute/value pair in the selected cluster.
* Each sequence state is considered a distinct value for the sequence attribute.
* The list of attribute values is sorted based on the frequency (order by frequency).

For example, the most likely attribute value in cluster1 is Start ➪ Music, which means that most of the Web visitors in cluster 1 start with the Music page. Movie is another popular URL that cluster 1 individuals like to visit.



Cluster: Cluster 1

Characteristics for Cluster 1

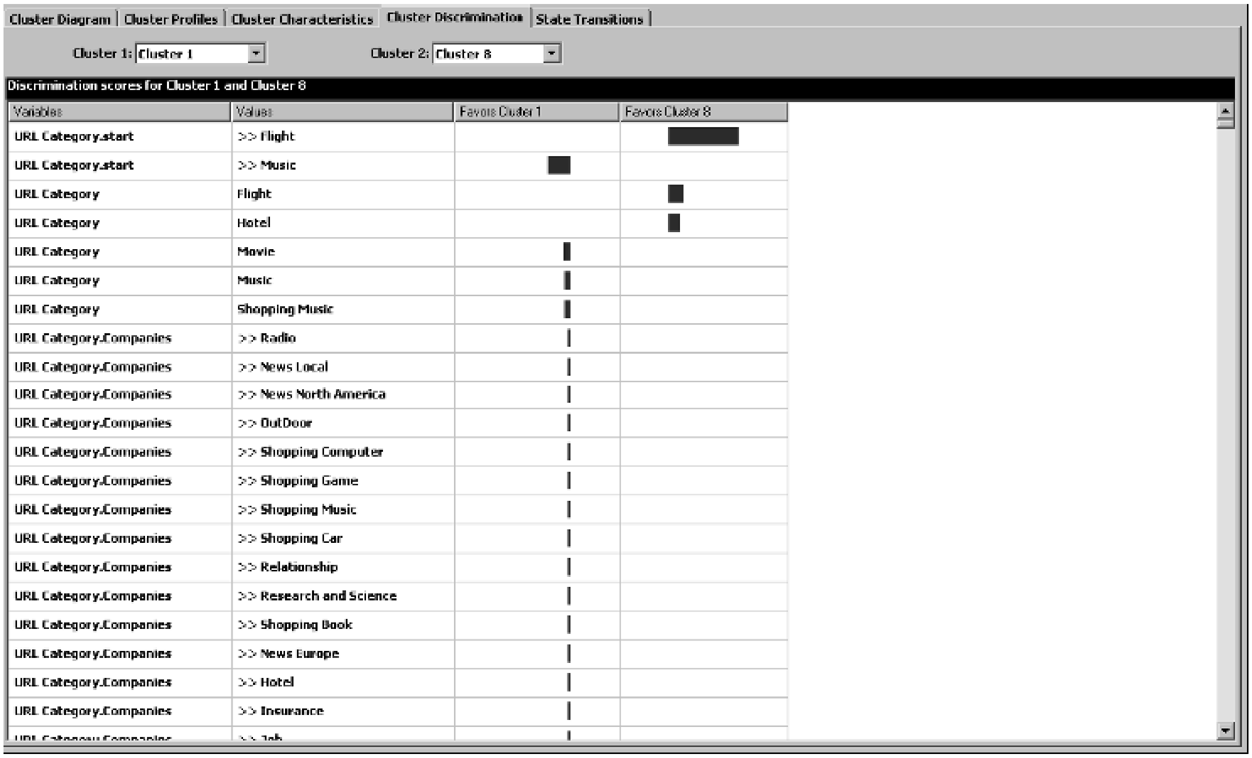
Variables	Values	Frequency
URL Category.transitions	Start >> Music	
URL Category	Movie	
URL Category	Music	
URL Category	Shopping Music	
URL Category.transitions	Start >> Shopping Software	
URL Category.transitions	Start >> Internet	
URL Category.transitions	Start >> Shopping Book	
URL Category.transitions	Start >> Relationship	
URL Category.transitions	Start >> Job	
URL Category.transitions	Start >> Medicine	
URL Category.transitions	Start >> OutDoor	
URL Category.transitions	Start >> Diet	
URL Category.transitions	Start >> Calendar	
URL Category.transitions	Start >> Shopping Game	
URL Category	Shopping Software	
URL Category	Software	
URL Category	Job	
URL Category	Shopping House	
URL Category	Shopping Computer	
URL Category	Shopping Book	
URL Category	Shopping Car	
URL Category	Internet	

Figure 9 Cluster characteristics.

IV. Cluster Discrimination:

Figure 10 shows the Cluster Discrimination pane.
This pane is designed to compare any two clusters, or to compare a cluster with the whole population or its complement.

From the figure, we can see that the biggest difference between cluster 1 and cluster 8 is that:
* cluster 1 customers end their navigation at a Music site while cluster 8 customers end their navigation at the Flight site.
* Cluster 1 customers like to go to Music and Movie sites, while cluster 8 customers like to visit Flight and Hotel URLs.



Cluster 1: Cluster 1 Cluster 2: Cluster 8

Discrimination scores for Cluster 1 and Cluster 8

Variables	Values	Favors Cluster 1	Favors Cluster 8
URL Category.start	>> Flight		████
URL Category.start	>> Music	█	
URL Category	Flight		█
URL Category	Hotel		▌
URL Category	Movie	▌	
URL Category	Music	▌	
URL Category	Shopping Music	▌	
URL Category.Companies	>> Radio	|	
URL Category.Companies	>> News Local	|	
URL Category.Companies	>> News North America	|	
URL Category.Companies	>> OutDoor	|	
URL Category.Companies	>> Shopping Computer	|	
URL Category.Companies	>> Shopping Game	|	
URL Category.Companies	>> Shopping Music	|	
URL Category.Companies	>> Shopping Car	|	
URL Category.Companies	>> Relationship	|	
URL Category.Companies	>> Research and Science	|	
URL Category.Companies	>> Shopping Book	|	
URL Category.Companies	>> News Europe	|	
URL Category.Companies	>> Hotel	|	
URL Category.Companies	>> Insurance	|	
URL Category.Companies	>> Job	|	

Figure 10 Cluster discrimination.

V. Cluster Transition:

Figure 11 shows the Cluster Transition pane.
It’s designed to display the sequence navigation patterns of each cluster.
* Each node is a sequence state.
* Each edge is the transition between these two states. It has a direction and weight.
* The weight is the transition probability.

From the figure, we can see that the main activities of customers in cluster 1 are Music, Shopping Music, and Movie, because those nodes are colored with the highest density.

There is a strong link from Music toward Shopping Music. Between those customers who are in the Shopping Music URL category, 64% will click on a Movie site next. About 45% of the customers in the cluster start with a Music page in the portal site.

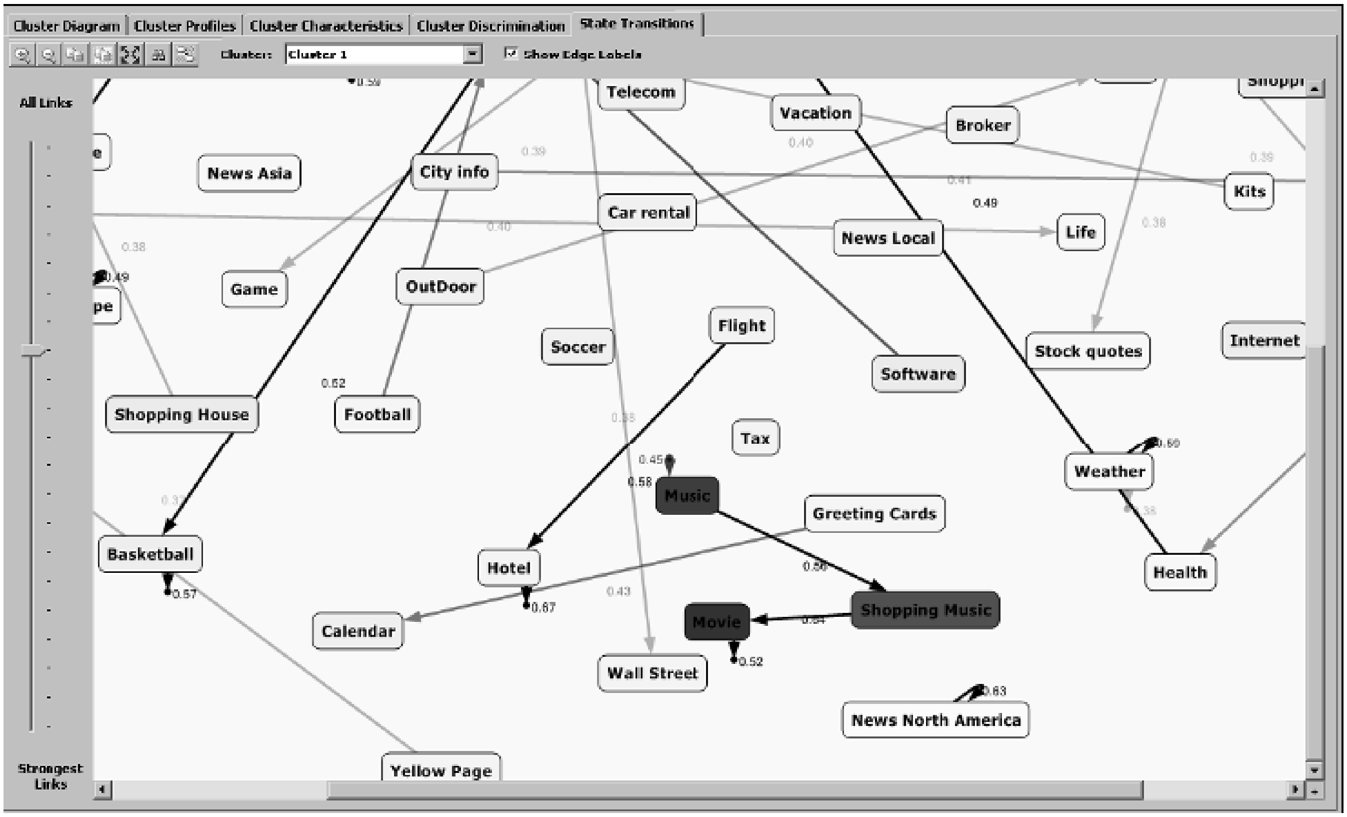

Cluster: Cluster 1
☑ Show Edge Labels

Figure 11 Cluster transitions.

Summary

In this project, we have seen the basic concepts of the Markov model and its application on sequence data. We also saw the principles of clustering based on sequenced attributes and nonsequenced attribute.

Lots of information in real life can be modeled as sequences, including weather, Web clicks, purchases, and so on. This project educated us how to build clustering models on these sequence data.
`PredictSequence` is a new DMX function introduced to predict the following states of a sequence attribute. We have seen the syntax and query result format of this function.

With careful preparation Sequence clustering algorithm will prove to be a good tool for modeling different types of sequences of actions and for making explicit the differences between these types.
Also, the fact that Markov chains have probabilities associated with transitions gives us a tool to deal with noise.

However, the algorithm has some limitation that, we hope, can be corrected in future versions. The first problem is its tendency for putting different types of sequences in the same cluster.

Finally the Sequence clustering viewer is a very powerful tool to help us exploring the sequence clustering model. The state transition tab of the viewer provides us with an easy way to understand the state transition matrix of each cluster.

List of figures

List of tables

Webographic references

- http://www.sqlserverdatamining.com/ssdm/Home/FAQ/tabid/55/Default.aspx
- http://msdn.microsoft.com/en-us/library/ms175462.aspx
- http://everything2.com/title/sequence%2520clustering
- http://www.sqlserverdatamining.com/ssdm/

Bibliographic references

- **Data mining with SQL server 2005:** written by ZhaoHui Tang and Jamie MacLennan and Published on 2005 by Wiley Publishing, Inc.

- **Process mining with sequence clustering:** written by students in the "Tecnica de lisboa" university on july 2007.