Sequence mining

Introduction

Sequences are an important type of data which occur frequently in many scientific, medical, security, business and other applications. For example, DNA sequences encode the genetic makeup of humans and all species, and protein sequences describe the amino acid composition of proteins and encode the structure and function of proteins. Moreover, sequences can be used to capture how individual humans behave through various temporal activity histories such as weblogs and customer purchase histories. Sequences can also be used to describe how organizations behave through sales histories such as the total sales of various items over time for a supermarket, etc.

Huge amounts of sequence data have been and continue to be collected in genomic and medical studies, in security applications, in business applications, etc. In these applications, the analysis of the data needs to be carried out in different ways to satisfy different application requirements, and it needs to be carried out in an efficient manner. Sequence data mining provides the necessary tools and approaches for unlocking useful knowledge hidden in the mountains of sequence data. The purpose of this book is to present some of the main concepts, techniques, algorithms, and references on sequence data mining.

This introductory chapter has four goals. First, it will provide some example applications of sequence data. Second, it will define several basic/generic concepts for sequences and sequence data mining. Third, it will discuss the major issues of interest in data mining research. Fourth, it will give an overview of the entire book.

1.1 Examples and Applications of Sequence Data

This section describes typical applications and common types of sequence data. It will demonstrate the richness of the types of sequence data, and serve as illustration of some formal concepts to be given in the next section.

1.1.1 Examples of Sequence Data

Biological Sequences: DNA, RNA and Protein

Biological sequences are useful for understanding the structures and functions of various molecules, and for diagnosing and treating diseases. Three major types of biological sequences are deoxyribonucleic acid (DNA) sequences, amino acid (also called peptide or protein) sequences, and ribonucleic acid (RNA) sequences. Figures 1.1 and 1.2 show respectively a part of a DNA sequence and a part of a protein sequence. RNA sequences are slightly different from DNA sequences. Below we briefly discuss some background information on these biological sequences.

The complete set of instructions for making an organism is called the organism's genome. A genome is often encoded in the DNA, which is a long polymer1 made from four types of nucleotides: adenine (abbreviated as A), cytosine (abbreviated as C), guanine (abbreviated as G) and thymine (abbreviated as T). The DNA contains both the genes, which encode the sequences of proteins, and the non-coding sequences.

GAATTCTCTGTAACACTAAGCTCTCTTCCTCAAAACCAGAGGTAGATAGA
ATGTGTAATAATTTACAGAATTTCTAGACTTCAACGATCTGATTTTTTAA
ATTTATTTTTATTTTTTCAGGTTGAGACTGAGCTAAAGTTAATCTGTGGC

Fig. 1.1. A DNA sequence fragment.

Proteins are polymers made from 20 different amino acids, using information present in genes. Genes are transcribed into RNA; RNA is then subject to post-transcriptional modification and control, resulting in a mature messenger RNA (mRNA); the mRNA is translated by ribosomes into the amino acids of the corresponding proteins. Each amino acid is the translation of a sequence interval of length 3 in the mRNA, which is also called a codon. The 20 amino acids are abbreviated as A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, and Y, respectively. RNA is made from four types of nucleotides: adenine (A), guanine (G), cytosine (C), and uracil (U). The first three are the same as those found in DNA, and uracil replaces thymine as the base complementary to adenine.

There are many data analysis problems of biological interest. Some examples include

• identifying genes and gene start sites from DNA sequences;

• identifying intron/exon splice sites from DNA sequences;

• identifying transcription promotors etc from DNA sequences;

```
SSQIRQNYSTEVEAAVNRLVNLYLRASYTYLSLGFYFDRDDVALEGVCHFFRELAEEKREGAERLLKMQNQRGGRAL
FQDLQKPSQDEWGTTPDAMKAAIVLEKSLNQALLDLHALGSAQADPHLCDFLESHFLDEEVKLIKKMGDHLTNIQ
RLVGSQAGLGEYLFERLTLKHD
```

Fig. 1.2. A protein sequence fragment.

• identifying non-coding RNA (also called small RNA) etc from RNA sequences;

• analyzing the structure and function of proteins from protein sequences;

• identifying the characteristic (motif) patterns of families of DNA, RNA or protein sequences;

• identifying useful sequence families; and

• comparing sequence families (e.g. comparing families associated with different species/diseases).

Advances on these problems can help us to better understand life and diseases.

Event Sequences: Weblogs, System Traces, Purchase Histories and Sales Histories

A major category of sequences are event sequences. Such sequences can be used to understand how the underlying actors (namely the objects which generated the event sequences) of the event sequences behave and how to best deal with them. The following are examples of event sequences.

A weblog is a sequence of user-identifier and event pairs (and perhaps other relevant information). An event is a request of some web resource such as a page (usually identified by the URL of the page) or a service. For each page requested, some additional information may be available, such as the type and the content of the page, and the amount of time the user spent on the page. The events in a weblog are listed in the timestamp ascending order. Figure 1.3 shows an example weblog, where a, b, c, d, e are events, and 100, 200, 300, and 400 are user identifiers. A weblog can also be restricted to a single user.

100, a, 100, b, 200, a, 300, b, 200, b, 400, a, 100, a, 400, b,300, a, 100, c, 200, c, 400, a, 400, e

Fig. 1.3. A weblog sequence.

System traces are similar to weblogs in form. They are sequences of records concerning operations performed by various users/processes to various data and resources in one or more systems.

Customer purchase histories are sequences of tuples, each consisting of a customer identifier, a location, a time, and a set of items purchased, etc. Figure 1.4 shows an example.

223100, 05/26/06, 10am, CentralStation, {W holeMealBread, AppleJuice},

225101, 05/26/06, 11am, CentralStation, {Burger, P epsi, Banana},

223100, 05/26/06, 4pm, W alMart, {M ilk, Cereal, V egetable},

223100, 05/27/06, 10am, CentralStation, {W holeMealBread, AppleJuice},

225101, 05/27/06, 12noon, CentralStation, {Burger, Coke, Apple}

Fig. 1.4. A customer purchase history.

Storewide sales histories are sequences of tuples, each consisting of a store ID, a time (period), the total sales of individual items for the time (period), and other relevant information. Such histories can also contain customer group information and some other information for the sales. Figure 1.5 shows an example.

97100, 05/06, {Apple : $85K, Bread : $100K, Cereal : $150K, ...},

90089, 05/06, {Apple : $65K, Bread : $105K, Diaper : $20K, ...},

97100, 06/06, {Apple : $95K, Bread : $110K, Cereal : $160K, ...},

90089, 06/06, {Apple : $66K, Bread : $95K, Diaper : $22K, ...}

Fig. 1.5. A storewide sales history.

1.1.2 Examples of Sequence Mining Applications

We now discuss some example data mining applications on event sequences.

Mining Frequent Subsequences

Ada is a marketing manager in a store. She wants to design a marketing campaign which consists of two major aspects. First, a set of products should be identified for promotion. Hopefully, for promoting those products, customers will be retained, and sales on other products will be stimulated. Second, a set of customers should be targeted so that the promotion information should be delivered.

To start with, Ada has the transactions of customers in the past. Each transaction includes the customer-id, the products bought in the transaction, and the timestamp of the transaction. Grouping transactions by customers and sorting them in the timestamp ascending order, Ada can get a purchase sequence database where each sequence records the behavior of a customer.

Ada may want to find frequent subsequences that are shared by many customers. As patterns, those frequent subsequences can help her to understand the behavior of customers. She can also identify products to be promoted according to the purchase patterns, and the target customers.

Classification of Sequences

Bob is a safety manager in an airline in charge of braking systems in airplanes. A sequence of status records is maintained for each aircraft. Maintaining the braking system of an airplane in a hub airport of the airline is highly desirable since maintenance cost is often several times higher when the job is done in a guest airport. On the other hand, being too proactive in maintenance may also lead to unnecessary cost since parts may be replaced too early and are not fully used.

Therefore, Bob is facing such a question: given an airplane's sequence of status records, predict in high confidence whether the plane needs a maintenance before it goes to the next hub airport. This is a classification problem (or as known as supervised learning) since the prediction is made based on some historical data, that is, some records of previous maintenances collected for references.

Clustering of Sequences

Carol is a medical analyst in charge of analyzing patients' reactions to a new drug. For each patient taking the drug (which is referred to as a case), she collects the sequence of reactions of the patient such as the changes in temperature, blood pressure, and so on. Typically, there are a good number, from 20 to more than 100, of such test cases. In order to summarize the results, she needs to categorize the cases into a few groups – all cases in a group are similar to each other, and the cases in different groups are substantially different from each other.

This is a clustering task (or as known as unsupervised learning), since the sequences are not labeled and the groups should be defined by Carol based on the similarity among sequences.

Other Examples

It is easy to name another dozens of examples of sequence data mining. For, example, by mining music sequences, we can predict the composers of music pieces. As another example, an interactive computer game can learn from players' behavior sequences to make it more intelligent and more fun.

The point we want to illustrate here is that sequence data mining is very practical in our lives, which makes it attractive for many researchers and developers.

1.2 Basic Definitions

This section defines the concepts of sequences, sequence types, sequence patterns2, sequence models, pattern matching, and support; it also discusses major characteristics of sequence data. Some of the definitions are generic, because there is considerable variation between specific instances in different applications. Examples of sequences were given in the previous section, and examples of the other concepts will be given in later chapters.

1.2.1 Sequences and Sequence Types

There is a rich variety of sequence types, ranging from simple sequences of letters to complex sequences of relations. Here we provide a very general definition which can capture most practical examples.

For a given application, sequences are constructed from members of some appropriate element types.

Definition 1.1. Element types are data types constructed from simple data types using several constructs; some common examples are the following:

• An item type is a finite set Σ of distinct items. Each x ∈ Σ is a member of the type. For example, the DNA sequences are constructed from the item type of Σ = {A, C, G, T }. We will frequently refer to the items as letters or symbols.

• A set type has the form 2τ , where τ is an element type. A member of this type is a finite set of members of type τ. In particular, for each finite set Σ of distinct items, 2Σ is a set type commonly referred to as a basket type. For example, market basket sequences are constructed from the element type of 2Σ, where Σ is a fixed set of items.

• A tuple type has the form τ = τ1, ..., τk
, where each τi is an element type, an ID type, a time type3 (such as Date and Time), or an amount type. The members of τ are precisely those tuple objects x1, ..., xk
where each xi is a member of τi. For example, weblog sequences can be constructed from the tuple of Date, T ime, URL
, where URL is a finite set of URLs.

Clearly, using set types and tuple types one can define types for relations.

Definition 1.2. A sequence over an element type τ is an ordered list4 S =

s1...sm, where

• each si (which can also be written as S[i]) is a member of τ, and is called an element of S;

• m is referred to as the length of S and is denoted by |S|;

• each number between 1 and |S| is a position in S.

A consecutive interval of sequence positions of the form [i, j], where 1 I j m is a window of the sequence; j − i + 1 is referred to as the length of the window.

Parenthesis and commas may be added to make sequences more readable.

Example 1.3. DNA sequences such as those shown in Figure 1.1 are sequences over {A, C, G, T }. The DNA sequence S = AT GT AT A has length 7, each number between 1 and 7 is a position in S, and S[3] is the letter G.

Protein sequences such as those shown in Figure 1.2 are sequences over {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y }.

Weblogs such as those shown in Figure 1.3 are over Date, T ime, URL
.

Customer purchase histories such as those shown in Figure 1.4 are over τ = CustomerID, Date, T ime, Location, 2I———————————————————————————————
, where the domain of Location is a (simple type) set of locations, and I is the set of product items. Storewide sales histories are similar.

The order among the elements of a sequence may be implied by time order as in event histories, or by physical positioning as in biological sequences.

The following general concepts are frequently used in biological sequence analysis:

• A site in a sequence (as in transcription binding site) is a short sequence window having some special biological property/interest. A site can be described by a start position and window length, or just a position. A site is usually characterized by the presence of some sequence pattern.

• Given a sequence S = s1...sm and a position i of S, the prefix s1...si−1 is often referred to as the upstream of i and the suffix si+1...sm is referred to as the downstream of i. The concepts are defined similarly for a window [i, j] (or site) of S, with s1...si−1 as the upstream, and sj+1...sm as the downstream. It is common to refer to position i − k of S as the −k region of position i, and to refer to position i + k as +k region of position i.

1.2.2 Characteristics of Sequence Data

Sequence data have several distinct characteristics, which lead to many opportunities, as well as challenges, for sequence data mining. These include the following:

• Sequences can be very long (and hence sequence datasets can have very high dimensionality), and different sequences in a given application may have a large variation in lengths. For example, the length of a gene can be as large as over 100K, and as small as several hundreds.

• Absolute positions in sequences may/may not have significance. For example, sequences may need to be aligned based on their absolute positions and there can be a penalty on position changes through insertion/deletions. In certain situations, one may just want to look for patterns which can occur anywhere in the sequences.

• The relative ordering/positional relationship between elements in sequences is often important. In sequences, the fact that one element occurs to the left of another is usually different from the fact that the first element occurs to the right of the second. Moreover, the distance between two elements is also often significant. The relative ordering/positional relationship between elements is unique to sequences, and is not a factor for relational data or other high dimensional data such as microarray gene expression data.

• Patterns can be substrings or subsequences. Sometimes a pattern must occur as a substring (of consecutive elements) in a sequence, without gaps between elements. At other times, the elements in a pattern can occur as a subsequence (allowing gaps between matching elements) of a sequence.

1.2.3 Sequence Patterns and Sequence Models

We now discuss sequence patterns, sequence models5, and related topics such as pattern matching and pattern support in sequence data. Due to the characteristics of sequence data discussed above, there are many possibilities for defining sequence patterns and sequence models. The purpose of this section is to provide a high-level unifying overview and show the many possibilities, rather than the detailed instances, of sequence patterns and sequence models. The detailed instances will be discussed in the subsequent chapters.

Roughly speaking, a sequence pattern/model consists of a number of single-position patterns plus some inter-positional constraints. A singleposition pattern is essentially a condition on the underlying element type. A sequence pattern may contain zero, one, or multiple single-position patterns for each position, where the single-position patterns for a given position are perhaps associated with a probability distribution; inter-positional constraints specify certain linkage between positions; such linkage can include conditions on position distance, and perhaps also include transition probabilities from position to position when two or more single-position patterns are present for some position. Below we give more details on these variations, together with some examples.

• A single-position pattern is a condition on the underlying element type defined recursively as follows: If τ is an item type, then a condition on τ can be “?” or “∗” or “·” (all denoting a single position wildcard or don’t care), an element of τ, a subset of τ, or an interval of τ when τ is an ordered type. If τ is a set type of the form {ψ}, then a condition on τ is a finite set of conditions on ψ. If τ is a tuple type of the form $\tau_1, ..., \tau_k$, then a condition on τ is an expression of the form $c_1, ..., c_k$, where c_i is a condition on τ_i. Since patterns are used to capture behavior in data, it may not make sense to have non-? conditions on ID types.

For example, if τ = {A, B, C, D}, {E, F, G}, int, real, then a singleposition condition can be A, {E,G}, ?,(20, 45]. If τ = {A, C, G, T }, then a single-position condition can be ?, C, {A, C} etc.

While it is possible to use the Boolean operators “AND” and “OR” to construct more complex conditions, this is seldom done since data mining of patterns must deal with a huge search space even without these Boolean operators. The intervals for ordered attributes are usually determined through a binning/discretization process.

• A sequence pattern is a finite set of single-position patterns of the form {$c_1, ..., c_k$}, together with a description of the positional distance relationships on the c_i’s and some other optional specifications. This formalization is general enough to include frequent sequence patterns, periodic patterns, sequence profile patterns, and Markov models. Below we give an overview of each of these.

A first representative sequence pattern type is the frequent sequence patterns. Each such a pattern consists of one single-position pattern for each position. For DNA sequences, an example of such a pattern is AT C. In the simplest case, the positions of the single-position patterns are a consecutive range of the positive integers – this is assumed when nothing is said about the relationships between the positions; in general, constraints on the positions (often referred to as gap constraints) can be included. For example, for the simplest case, A, T and C are at consecutive positions so that T ’s position is after A’s position and C’s position is after T ’s; for the general case, we may say that T ’s position is at least 2 and at most 5 positions after the position of A, and that C’s position is at most 3 positions after T ’s position. One can also add a window constraint to restrict the difference between the positions of the last and the first single-position patterns to be at most, for example, 7.

Frequent sequence patterns can be viewed as periodic sequence patterns. We will discuss some distinctions between frequent sequence patterns and periodic sequence patterns below.

A second representative sequence pattern type is the sequence profile patterns. Such a pattern is over a set of positions, and it consists of a set of single-position pattern plus a probability distribution. Examples will be given in Chapter 4.

A third representative sequence pattern type is the Markov models. Such a model consists of a number of states plus probabilistic transitions between states. In some cases each state is also associated with a symbol emission probability distribution. Examples will be given in Chapter 4.

A fourth representative sequence pattern type is the partial order models. Each such a model contains a set of single-position patterns associated with a partial order on these patterns. In a sense, the position distance between pairs of single-position patterns is in the range of [1, ∞). Such a model can capture a temporal event ordering on the events. Examples will be given in Chapter 5.

• In addition to sequence pattern mining discussed above, classification and clustering are also useful data mining tasks for sequence data. Neither these tasks nor their products fall under the general definition of sequence patterns given above. The characteristics of sequence data lead to new questions for these two tasks. For example, there are more possibilities for feature construction from sequence data. Moreover, in sequence data one may want to predict the "class" of a location in a long sequence, which does not have a counterpart for conventional relational/vector data. More details will be provided in Chapters 3 and 4.

We now turn to the issues regarding pattern matching and sequence pattern support in sequence data. We first need several definitions.

A match between a sequence pattern p = p1...pk and a sequence s = s1...sn is a function f from {1, ..., k} to {1, ...m} such that the condition pi is satisfied in sf(i) and the associated constraints on p are satisfied. The concept of satisfaction is defined in the natural manner.

For each match between a sequence pattern and a sequence, let the match interval be defined as [low, high], where low is the smallest position, and high is the largest position, in the sequence for the match. We note that, for sequence patterns with gaps, it is possible that the matching interval of one match is properly contained in the matching interval of a second match.

Several possibilities exist regarding which matches can contribute towards the count/support of a pattern:

• One sequence contributes at most one match and the support/count of pattern is with respect to the whole dataset. This simple case is very similar to the conventional transactional data case.

• One sequence contributes multiple matches and the count of pattern is with respect to one sequence. Three options exist: (b) Different contributing matches are completely disjoint, in the sense that the matching intervals of different contributing matches must be completely disjoint. (b) Different contributing matches are sufficiently disjoint, in the sense that the matching intervals of different contributing matches must not overlap more than some given threshold. (c) All matches are counted. For options (a) and (b), it may be computationally expensive to determine the highest possible number of matches of a pattern in a sequence.

A sequence model can be used as a generative device. For example, one can compute the most likely sequence that can be generated by a Markov model.

Some distinctions can be made between sequence patterns and sequence models, similar to the distinctions between general patterns and general models. A pattern is usually partial (or local) in the sense that it may occur only in a subset of the sequences under consideration. On the other hand, a model is usually total (or global) in the sense that it can be applied to every sequence under consideration.

1.3 General Data Mining Processes and Research Issues

In this section we give a brief high level overview of the general data mining process, and the general issues of interest in data mining research and applications. More details on these can be found in general data mining texts.

The typical steps of the data mining process are the following:

• Understanding the application requirements and the data. In this step the analyst will need to understand what is important, and how such importance is reflected in data.

• Preprocessing of the data by data cleaning, feature/data selection, and data transformation. Data cleaning is concerned with removing inconsistency in data, with integrating data from heterogeneous sources etc. Feature selection is concerned with selecting the more useful features (for a particular data mining task) from a large number of candidate features. Feature construction is about producing new features from existing features. Data transformation is concerned with mapping data from one form to another. Discretization (also called binning) is a common approach of data transformation, where one maps an attribute with a large domain into an attribute with a smaller domain. Common discretization methods include equi-width binning, equi-density binning, and entropy-based binning.

• Mining the patterns/models. This is done by running some data mining algorithms on the data produced from the last step above.

• Evaluation of the mining result. In this step the data analyst will apply various measures to evaluate the goodness of the mined patterns or models for the application under consideration.

These steps may be iterated to improve the quality of the mining result. Improvement is possible since one’s understanding of the data/application deepens after one or more iterations of working through the data.

Naturally, data mining research should address issues of practical/ theoretical interest, and solving important problems, in data mining applications. Data mining research often considers the following technical issues:

• Formulating useful new concepts that have high potential to lead to advances of research in the field.

• Designing novel techniques for efficiency and scalability in computational space/time, for dealing with large volume of data and with high dimensionality of data. The techniques should address the unique challenges and take advantage of the unique opportunities of the underlying application/data.

• Optimizing cluster/classification quality under measures such as accuracy, precision and recall, and cluster quality (intra-cluster similarity and intercluster dissimilarity).

• Optimizing pattern interestingness under appropriate measures, such as support/confidence, surprise, lift/novelty and actionability.