

Algorithms of Sequential Pattern Generation with Noise using Stochastic and Fuzzy Models

Maria A. Butakova, Andrey V. Chernov^(✉), and Alexander N. Guda

Rostov State Transport University, Rostov-on-Don, Russia
{butakova,avcher,guda}@rgups.ru

Abstract. A task of sequential pattern generation can be considered as a problem which is inverse to sequential pattern mining. This paper presents two novel approaches to the sequential pattern generation with noise, namely the approach based on stochastic automata and context-free grammars and the approach based on Hidden Markov model. The distinctive feature of these methods is the suitability to produce an output in the noisy and fuzzy input data. Also, we present the detailed calculation algorithms to the proposed approaches.

Keywords: Sequential pattern · Stochastic automata model · Hidden markov model · Fuzzy pattern

1 Introduction and Preliminary Work

We live in the real world which can be characterized as Big Data era. Intelligent systems, network devices traffic, Internet of Things, Machine-to-Machine interaction and social networks creates unprecedented huge data arrays. The data itself has a heterogeneity and in some sense the unlimited volume. Under this circumstances, the traditional statistical verification-driven Data Mining are giving way to discovery-driven Data Mining [1]. One of the basic elements of modern discovery-driven Data Mining is a pattern concept. Patterns represent some regularities in acquired data and can be obtained in compact form and understandable to people. It is worth to notice that raw data can contain some level of incompleteness, vagueness, uncertainty and other factors making them non-crisp data. The most prominent methods to describe these phenomena, surely the probability theory, fuzzy sets theory, rough sets theory and numerous their extensions.

Patterns discovery is being implemented by methods are not framed any a priori assumptions about the sample structure, probability laws, fuzziness or roughness of sets of values. The important position of modern Data Mining methods is the non-triviality of patterns to be discovered. The found patterns

The work was financially supported by Russian Foundation for Basic Research (projects 15-01-03067-a, 16-01-00597-a, 15-08-01886-a).

should reflect non-obvious, unexpected regularities in the data that make up the so-called hidden knowledge. One of the promising approaches to Data Mining is Sequential Pattern Mining (SPM). The SPM attracts much research attention [2, 3] in recent years. It has broad applications, including behaviour analysis [4], analysis of web-access patterns [5], discovering the patterns of transactions in databases [6] and patterns of computer attacks [7] and intrusion detection [8].

The problem of sequential pattern mining formally defined as follows. A set of sources and a set of items $I = \{i_1, i_2, \dots, i_n\}$ exists. At some time moment we register an event $e \subseteq I$ that is a collection of items. An ordered list of events defines a sequence $s = \langle s_1, s_2, \dots, s_k \rangle$. In general case a sequence s can be stored in a database D with corresponded indication of event source. The pattern is a some user-defined subsequence $s_p \subseteq S$ that is formed depending on $X_i(S, D)$ indicator variable, whose value equal 1, if $s_p i \leq D_i$, where D_i is a i th source sequence in database D , and 0 otherwise. The mining task is finding all $s_p i$ patterns suited to some user-defined constrains, for example, patterns with specific al values, or related to user-defined threshold.

Further extensions of SPM problem we can divide into three categories: the extensions that concern to adding an uncertainty [9, 10], the extensions that propose improvements concerning to the fuzzification [11, 12] and the pattern discovery in a noisy environment [13]. An uncertainty in SPM problem concerns to the possibility of getting inaccurate data from sources or attribute level uncertainty [14]. Also, an event level uncertainty exists and described in [15].

Fuzzification procedures for input values can be implemented in different ways. A simplest procedure is based on the measure of difference between variables x_i and y_i in a, b interval one can define with two parameter membership

$$\text{function: } f_i(x_i, y_i) = \begin{cases} 0, & |x_i - y_i| \leq a \\ \frac{|x_i - y_i| - a}{b - a}, & a < |x_i - y_i| < a + b \\ 1, & |x_i - y_i| \leq a + b \end{cases}$$

General considerations about noise structure we describe in the following way. Let's denote $E = (E_i)_{i \geq 1}$ is a stochastic process with $E_i \in \{0, 1\}$ binary random variables. Also we mean that $E_i \in GF(2)$ with standard addition and multiplication, and further we mean this variables as presence or absence of noise in the generated sequence. As a noise model base we consider $\{\Omega, F, (F_i)_{i \geq 0}, P\}$ stochastic basis, where Ω is a probability space (a state of modelling subsequence), F is a sigma algebra subsets of Ω , (sequence of random events in modelling subsequence), F_i is non-decreasing sequence sigma algebras of F and element F_i is a information about state available up to and including the time i . The initial sigma algebra is $F_0 = \sigma\{\Omega, \emptyset\}$ a trivial sigma algebra $F = \bigcup_0^\infty F_i$.

The paper contains two sections that present main results. Section 2 proposes a novel algorithm to sequential pattern generation using a stochastic automata model. In Sect. 3 we develop an algorithm for sequential pattern generation based on Hidden Markov model.

2 Fuzzy Sequential Pattern Generation Based on Stochastic Automata Model

In this section we propose a novel algorithm for fuzzy sequential pattern generation and first of all we would notice that fuzzification stage is not being discussed. The pattern, as it is discussed in the previous section consists of events series, that modelling two fuzzy states: a “bad” state and a “good” state. We use of quasi-periodic processes is based on the assumption that the same events can be stored on a set of contiguous positions, that is, a series of events distinguish between a series of “bad” events and a series of “good” events in the pattern. Thus, it is possible to distinguish alternating events, the events, in which the probabilities of errors are significantly different. Let’s define λ is a number of erroneous “bad” events, and μ is a number of “good” events without errors. Obviously, the lengths of series of “bad” events and “good” events are random variables with probability distributions $f_\lambda(\tau)$ and $f_\mu(\tau)$ respectively, $\tau = 0, 1, \dots, n, \dots$. The we consider a sequence $\langle \lambda_i, \mu_i \rangle_{i \geq 0}$. Random variables λ_i and μ_i are independent. Random variables $\lambda_0, \lambda_1, \dots$ are independent and identically distributed with distribution $f_\lambda(\tau)$ and random variables μ_0, λ_1, \dots are independent and identically distributed with distribution $f_\mu(\tau)$. A sequence $\langle \lambda_i, \mu_i \rangle_{i \geq 0}$ can be divided into two non-overlapping intervals as follows:
 if pattern starts with “good” state

$$[0, \mu_0 - 1], [\mu_0, \mu_0 + \lambda_0 - 1], \dots, \left[\sum_{i=0}^n \mu_i + \sum_{i=0}^{n-1} \lambda_i, \sum_{i=0}^n \mu_i + \sum_{i=0}^n \lambda_i - 1 \right], \dots,$$

and if pattern starts with “bad” state

$$[0, \lambda_0 - 1], [\lambda_0, \lambda_0 + \mu_0 - 1], \dots, \left[\sum_{i=0}^n \lambda_i + \sum_{i=0}^{n-1} \mu_i, \sum_{i=0}^n \lambda_i + \sum_{i=0}^n \mu_i - 1 \right], \dots$$

Then, let’s consider a noise as a binary sequence E which is divided into segments by stochastic intervals E^0, E^1, \dots and obtain a concatenation of quasi-periods. Based on widely known average moving models we denote M_1 is a model related to generation of “bad” erroneous events sequence and M_2 is a model related to generation of “good” error-free events sequence. We use following generation rule: if the segment of generated pattern is related to interval

$$\left[\sum_{i=0}^n \mu_i + \sum_{i=0}^{n-1} \lambda_i, \sum_{i=0}^n \mu_i + \sum_{i=0}^n \lambda_i - 1 \right]$$

then it is created M_1 model; if the segment of generated pattern is related to interval

$$\left[\sum_{i=0}^n \lambda_i + \sum_{i=0}^{n-1} \mu_i, \sum_{i=0}^n \lambda_i + \sum_{i=0}^n \mu_i - 1 \right]$$

then it is created M_2 model.

The more general generation model can be obtained using attribute grammars [16] in the following way. A context-free grammar is a quad-tuple $G = \langle S, N, T, P \rangle$, where S is starting symbol, N is alphabet of non-terminal symbols, T is alphabet of terminal symbols, P is set of inference rules. We assume that in grammar there are no non-terminal symbols that are not belonging to any of the conclusions. Attribute grammar consists of context-free grammar G , which is called the base of attribute grammar, mappings Z and I , which match each character $X \in N \cup T$ disjoint sets $Z(X)$ and $I(X)$ of synthesized and inherited

attributes, as well as from sets $M(p)$ we obtain the sets of semantic rules (rules of calculation of attributes) for each rule $p \in P$.

2.1 Formal-Grammar Noise Model

Next we propose a modified attribute grammar and as a base we use a stochastic context-free grammar $G = \langle S, N, T, Q \rangle$. The element Q is a finite set of stochastic inference rules. The stochastic inference rule is $Y \xrightarrow{q} X$, where $Y \in N$, $X \in (N \cup T)^*$, q is a probability of application of inference rule. We assume that Q has a set of stochastic inference rules $Y \xrightarrow{q_1} X_1, Y \xrightarrow{q_2} X_2, \dots, Y \xrightarrow{q_n} X_n$, then $\sum_{i=1}^n q_i = 1$ and all $q_i > 0$. Mappings $Z(X)$ and $I(X)$ are defined for each $X \in T$.

The set of semantic rules is defined for each $X \in T$.

Next we interpret a terminal symbols alphabet. Attributes in alphabet A are $a(X), b(X), M(X)$. Attribute $a(X)$ is inherited attribute, attributes $b(X)$ and $M(X)$ are synthesized attributes. Variable $b(X)$ is a random variable with probability distribution $f_X(\tau)$. Attribute $M(X)$ is an average moving model related to pattern state (“bad” or “good”). Let’s X is included into subsequence $\lambda \in (N \cup T)^*$ and if Y is a symbol that previous to X in subsequence λ , then $a(X) = a(Y) + b(Y)$. If in the subsequence λ there is no terminal symbols preceding to X , then $a(X) = 0$. Hence, every terminal symbol creates stochastic interval $[a(X), a(X) + b(X) - 1]$ and defines the probability distribution for noisy segments in generated pattern corresponding to interval $[a(X), a(X) + b(X) - 1]$ using $M(X)$ model. Further, we will consider only automata attribute grammars with following inference rules: $Y \xrightarrow{q} ZX$, where $Z \in T$ and $X \in N$ or $Y \xrightarrow{q} Z$ or $Y \xrightarrow{q} \varepsilon$, where ε is an empty subsequence. The reasoning above allow us to propose the following Algorithm 1 for generating noisy segments of the pattern.

Algorithm 1. Generation of noisy segments of the pattern

- 1: **Initialization**
 - 2: $Gen1$ - random normal variable, $Gen2$ - uniform random variable in $[0,1]$
 - 3: Define length of segment. $i = 0$
 - 4: Generate initial terminal symbol X_0
 - 5: $a(X_0) = 0, b(X_0) \leftarrow Gen2, f_{X_0}(\tau), d(X_0) = \min(b(X_0) - 1, N), seq_0 \leftarrow Gen1, M(X_0)$
 - 6: **Iteration**
 - 7: **if** $d(X_i) = N$ **then** then 12
 - 8: $i = i + 1$
 - 9: Generate $X_i, a(X_i) = d(X_{i-1}) + 1, b(X_i) \leftarrow Gen2, f_{X_i}(\tau)$
 - 10: $d(X_i) = \min(a(X_i) + b(X_i) - 1, N)$
 - 11: $seq_i \leftarrow Gen1, M(X_i)$
 - 12: **Stop**
-

Next we give some comments about method above. A model is described by modified attribute grammar $N = \{S, A, B\}, T = \{a, b\}$, where symbol a is

a fuzzy “good” state, and symbol b is a fuzzy “bad” state and inference rules are $Q = \left\{ S \xrightarrow{q} aB, S \xrightarrow{1-q} bA, A \xrightarrow{1} aB, B \xrightarrow{1} bA \right\}$. Sequence can start on subsequence with “good” state or on a “bad” state with probabilities q or $1 - q$ respectively. The range of good states corresponds to the gap between groups of states. Within the group, the intervals of “bad” states alternate with intervals of “good” states. State groups alternate with intervals of “good” states. The number of “bad” intervals within a group is a random variable distributed geometrically.

Next, we propose more general Algorithm 2, which takes into account intermediate state of subsequence between “bad” and “good” intervals. In the following we use notation “a,b,c” for “bad”, “intermediate” and “good” intervals in generated sequence pattern respectively, $f_X(\tau)$ is a random variable that denotes length of interval. $M(\mathbf{X})$ is an moving average model the following type: $x_i = d_0 + d_1x_{i-1} + d_2x_{i-2}$, where instead of the coefficients we take $a_0, a_1, a_2, b_0, b_1, b_2$ and c_0, c_1, c_2 for “bad”, “intermediate” and “good” intervals respectively.

Algorithm 2. Generation sequential pattern with noise

- 1: **Start**
 - 2: **for** $i = 1, N$ **do**
 - 3: Generate random variable x_i
 - 4: Compute $E_i = \begin{cases} 0, & d_0x_i + d_1x_{i-1} + d_2x_{i-2} \leq \Theta \\ 1, & \text{otherwise.} \end{cases}$
 - 5: Forming a grammar $N = \{S, L, M, Q\}$. A set of terminal symbols is $\{a, b, c\}$, S is starting symbol.
 - 6: Inference rules are: $S \xrightarrow{P_1} cL, S \xrightarrow{P_2} cM, S \xrightarrow{P_3} cQ, S \xrightarrow{P_0} \varepsilon, \sum P_i = 1, M \xrightarrow{P_6} bM, M \xrightarrow{P_7} bQ, Q \xrightarrow{P_8} aS, Q \xrightarrow{P_9} aS.$
 - 7: Generate subsequence’s lengths k and obtain random variables $\xi_1, \xi_2, \dots, \xi_k, |\lambda| = k$
 - 8: **for each** i **do**
 - 9: interval 1: $[0, \xi]$;
 - 10: interval 2: $[\xi_1 + 1, \xi_1 + \xi_2]$;
 - 11: interval 3: $[\xi_1 + \xi_2 + 1, \xi_1 + \xi_2 + \xi_3]$;
 - 12: interval i : $\left[\sum_{j=1}^{i-1} \xi_j + 1, \sum_{j=1}^i \xi_j \right]$.
 - 13: Produce output sequence
 - 14: **Stop**
-

The algorithm above requires the data chosen from arrays must be deleted in order to avoid repetitions in generated sequence.

3 Sequential Pattern Generation with Fuzziness Based on Hidden Markov Model

Hidden Markov models, the doubly embedded stochastic process with an underlying not observable stochastic process now is a tool for SPM [17, 18].

Again, we will consider an additive noise $E = \{E_i\}$, as a random discrete sequence, and will add random state of the generated pattern interval $D = \{D_i\}$ with $D_i \in W_D$. The probability distribution of noise $E : P(E) = \sum_D P(E/D) P(D)$.

The presence of the conditional distribution $P(E/D)$ is due to the occurrence of interference is completely determined by the state of interval D . The independence condition leads us to the $P(E/D) = \prod_i P(A_i/D)$ probability distribution. From the casual condition it follows $P(E_i/D) = P(E_i/D^{(i)})$, where $D^{(i)} = (\dots, D_{i-1}, D_i)$. We decide that enough condition of the noise dependency on the state is $D^{(i)} = (\dots, D_{i-1}, D_i)$ which means that E_i depends on D_i only. This, the probability distribution law is:

$$P(E) = \sum_D \prod_i P(E_i/D_i) P(D) \tag{1}$$

If sequence D is a Markov sequence $P(D) = P(D_0) \prod_{i \geq 1} P(D_i/D_{i-1})$ than (1) has following form:

$$P(E) = \sum_{D_0} P(D_0) \sum_{D_1} P(E_1/D_1) P(D_1/D_0) \sum_{D_2} P(E_2/D_2) P(D_2/D_1) \dots, \tag{2}$$

$\sum_{D_n} P(E_n/D_n) P(D_n/D_{n-1})$, where $n = |E|$. We assume sequences E, D are finite. Equation (2) creates recurrent procedure calculation of $P(E)$ sequence of functions ϕ as follows:

$$\begin{aligned} \phi_n(E^n, D_{n-1}) &= \sum_{D_n} P(E_n/D_n) P(D_n/D_{n-1}); \\ \phi_k(E^k, D_{k-1}) &= \sum_{D_k} P(E_k/D_k) P(D_k/D_{k-1}) \phi_{k+1}(E^{k+1}, D_k); \\ k &= n - 1, \dots, 1; \\ P(E) &= \sum_{D_0} P(D_0) \phi_1(E^1, D_0). \end{aligned} \tag{3}$$

Hence, from (3) it is following that $E^k = \{E_i\}_{i=k}^n$, and particularly, $E = E^1$. Let's consider a model with $E_i \in \{0, 1\}, D_i \in \{0, 1\}$ interleaving "bad" and "good" states, where $P(E_i=1/0) = 0, P(E_i=1/1) = 1 - \varepsilon, P(0/0) = p_0, P(1/1) = 1 - p_1$. The proposed model is completely defined by three parameters ε, p_0, p_1 . Condition probability distributions is defined as:

$$\begin{aligned} P(E_i/0) &= -E_i + 1, P(E_i/1) = (1 - 2\varepsilon) E_i + \varepsilon, \\ P(D_i/0) &= (1 - 2p_0) D_i + p_0, P(D_i/1) = (2p_1 - 1) D_i + p_1 \end{aligned} \tag{4}$$

From (2) and (4) it follows

$$P(E) = \alpha \prod_{i=1}^n E_i + \sum_{i=1}^n \alpha_i \prod_{j=1, j \neq i}^n E_j + \sum_{i=1}^n \sum_{j=1, j \neq i}^n \alpha_{i,j} \prod_{k=1, k \neq i, k \neq j}^n E_k + \alpha_{1,2,\dots,n} \tag{5}$$

The Eq. (5) is not quite suited for $P(E)$ calculation, thus we use Eq. (3) in the following way:

$$\begin{aligned} \phi_n(E^n, 0) &= (-E_n + 1)p_0 + ((1 - 2\varepsilon)E_n + \varepsilon)(1 - p_0), \\ \phi_n(E^n, 1) &= (-E_n + 1)p_1 + ((1 - 2\varepsilon)E_n + \varepsilon)(1 - p_1), \\ \phi_k(E^n, 0) &= (-E_n + 1)p_0\phi_{k-1}(E^{k-1}, 0) + ((1 - 2\varepsilon)E_n + \varepsilon)(1 - p_0)\phi_{k-1}(E^{k-1}, 1), \\ \phi_k(E^n, 1) &= (-E_n + 1)p_1\phi_{k-1}(E^{k-1}, 0) + ((1 - 2\varepsilon)E_n + \varepsilon)(1 - p_1)\phi_{k-1}(E^{k-1}, 1), \end{aligned} \tag{6}$$

$$P(E) = P(0)\phi_1(E^1, 0) + (1 - P(0))\phi_1(E^1, 1).$$

Let's consider the estimation of hidden sequence D from observed sequence E . It should be noticed that sequence $D_i^* = 1$, if $E_i = 1$ and D_i^* is equal to 0 or 1, if D_i^* . Due to this circumstance we can divide our task into two subtasks:

$$\begin{aligned} &\max_{D_0^{i_1}} \left(\sum_{i=1}^{i_1} (\ln P(E_i=0/D_i) + \ln P(D_i/D_{i-1})) + \ln P(D_0) \right) \\ &\max_{D_{i_k}^{i_{k+1}}} \left(\sum_{i=i_k}^{i_{k+1}} (\ln P(E_i=0/D_i) + \ln P(D_i/D_{i-1})) \right) \\ &D_{i_k}^{i_{k+1}} = \{D_{i_k}, \dots, D_{i_{k+1}}\} \end{aligned}$$

where

$$k = 2, \dots, l.$$

With our parameters ε, p_0, p_1 we can apply following equation

$$\max_{\varepsilon} \sum_{i=1}^n \ln P(E_i/D_i, \varepsilon)$$

and

$$\max_{p_0, p_1} \sum_{i=1}^n P(D_i/D_{i-1}, p_0, p_1)$$

And, based on considerations above we propose Algorithm 3 for generating a sequence pattern using a Hidden Markov model. We also use two states of

generated interval: g is a “good” state and b is a “bad” state. The probabilities $P(E_i=1/g)$ and $P(E_i=1/b)$ is done. In preliminary we have a stochastic matrix $Q = \begin{pmatrix} p & 1-q \\ 1-p & q \end{pmatrix}$, where p is a probability of “good” state, and q is probability of “bad” state. Initial probabilities p_0 and $1-p_0$ also are defined.

Algorithm 3. Generation sequential pattern based on Hidden Markov model

- 1: **Start**
 - 2: Input N - the length of the sequence
 - 3: Generate random variables according $Gen1$, $Gen2$
 - 4: Generate initial values with **Gen1** and probabilities $P(E_i=1/g)$ or $P(E_i=1/b)$.
 - 5: $i = 0$
 - 6: $R = \begin{pmatrix} p_0 \\ 1-p_0 \end{pmatrix}$
 - 7: **if** $i \neq N$ **then**
 - 8: $i = i + 1$
 - 9: Calculation $i = i + 1$
 - 10: Generation of interval of sequence pattern with R and $Gen2$
 - 11: Generation of “good” state with probability $P(E_i=1/g)$
 - 12: Generation of “bad” state with probability $P(E_i=1/b)$
 - 13: Goto 6
 - 14: **Stop**
-

Generating a random binary sequence using a generator will be quasi-periodic with alternating “bad” and “good” intervals. The length of “bad” interval is a random variable distributed with $P(\lambda = i) = q^i(1-q)$. The length of “good” intervals is a random variable distributed with $P(\mu = i) = p^i(1-p)$.

4 Conclusions

In this paper, two novel approaches to sequential pattern generation problems have been proposed. In the first approach, authors have proposed a novel algorithm suited for fuzzy sequential pattern generation based on stochastic automata model. The distinctive feature of the presented algorithm is the possibility of modeling noisy segments with original formal-grammar noise model. The second novel approach for sequential pattern generation problem authors is based on Hidden Markov model. In this case, authors consider additive noise as a random discrete sequence and fuzzification procedure with stochastic matrix generating based on quasi-periodic intervals. Three detailed algorithms for implementing approaches above have been developed.

Acknowledgements. The authors gratefully acknowledge financial support from the Russian Foundation for Basic Research.

References

1. Sumathi, S., Sivanandam, S.N.: Introduction to Data Mining and its Applications, vol. 29, pp. 1–828. Springer, Heidelberg (2006). doi:[10.1007/978-3-540-34351-6](https://doi.org/10.1007/978-3-540-34351-6)
2. Li, T.R., Xu, Y., Ruan, D., Pan, W.: Sequential pattern mining. In: Intelligent Data Mining, vol. 5. Studies in Computational Intelligence, pp. 103–122 (2005). doi:[10.1007/11004011_5](https://doi.org/10.1007/11004011_5)
3. Shen, W., Wang, J., Han, J.: Sequential pattern mining. In: Frequent Pattern Mining, pp. 261–282 (2014). doi:[10.1007/978-3-319-07821-2_11](https://doi.org/10.1007/978-3-319-07821-2_11)
4. Wang, Y.L., Wen, L.M., Chen, T.S., Chen, R.C.: Using sequential pattern mining to analyze the behavior on the WELS. In: Qu, X., Yang, Y. (eds.) Information and Business Intelligence. CCIS, vol. 267, pp. 95–101 (2012). doi:[10.1007/978-3-642-29084-8_15](https://doi.org/10.1007/978-3-642-29084-8_15)
5. Zhou, B., Hui, S.C., Fong, A.C.M.: Efficient sequential access pattern mining for web recommendations. *Int. J. Knowl. Based Intell. Eng. Syst.* **10**(2), 155–168 (2006). doi:[10.3233/KES-2006-10205](https://doi.org/10.3233/KES-2006-10205)
6. Gouda, K., Hassaan, M.: Mining sequential data patterns in dense databases. *Int. J. Database Manag. Syst. (IJDMS)* **3**(1), 179–194 (2011). doi:[10.5121/ijdms.2011.3112](https://doi.org/10.5121/ijdms.2011.3112)
7. Sun, H., Sun, J., Chen, H.: Mining frequent attack sequence in web logs. In: Huang, X., Xiang, Y., Li, K.C. (eds.) Green, Pervasive, and Cloud Computing. LNCS, vol. 9663, pp. 243–260 (2016). doi:[10.1007/978-3-319-39077-2_16](https://doi.org/10.1007/978-3-319-39077-2_16)
8. Song, S.J., Huang, Z., Hu, H.P., Jin, S.Y.: A sequential pattern mining algorithm for misuse intrusion detection. In: Jin, H., Pan, Y., Xiao, N., Sun, J. (eds) Grid and Cooperative Computing - GCC 2004 Workshops, GCC 2004. LNCS, vol. 3252, pp. 458–465 (2004). doi:[10.1007/978-3-540-30207-0_57](https://doi.org/10.1007/978-3-540-30207-0_57)
9. Muzzamal, M., Raman, R.: Mining sequential patterns from probabilistic databases. *Knowl. Inf. Syst.* **44**(2), 325–358 (2015). doi:[10.1007/s10115-014-0766-7](https://doi.org/10.1007/s10115-014-0766-7)
10. Zhao, Z., Yan, D., Ng, W.: Mining probabilistically frequent sequential patterns in uncertain databases. In: Proceedings of the 15th International Conference on Extending Database Technology, EDBT 2012, pp. 74–85 (2012). doi:[10.1145/2247596.2247606](https://doi.org/10.1145/2247596.2247606)
11. Chen, R.S., Tzeng, G.H., Chen, C.C., Hu, Y.C.: Discovery of fuzzy sequential patterns for fuzzy partitions in quantitative attributes. In: Proceedings of ACS/IEEE International Conference on Computer Systems and Applications, Beirut, pp. 144–150 (2001). doi:[10.1109/AICCSA.2001.933967](https://doi.org/10.1109/AICCSA.2001.933967)
12. Chen, Y.L., Huang, T.C.: Discovering fuzzy time-interval sequential patterns in sequence databases. *IEEE Trans. Syst. Man Cybern. B Cybern.* **35**(5), 959–972 (2005). doi:[10.1109/TSMCB.2005.847741](https://doi.org/10.1109/TSMCB.2005.847741)
13. Wang, W., Yang, J., Yu, S.: Mining patterns in long sequential data with noise. *ACM SIGKDD Explor. Newsl. Spec. Issue Scalable Data Min. Algorithms* **2**(2), 28–33 (2000). doi:[10.1145/380995.381008](https://doi.org/10.1145/380995.381008)
14. Muzzamal, M., Raman, R.: Uncertainty in sequential pattern mining. In: MacKinnon, L.M. (eds.) Data Security and Security Data, BNCOD 2010. LNCS, vol. 6121, pp. 147–150 (2010). doi:[10.1007/978-3-642-25704-9_18](https://doi.org/10.1007/978-3-642-25704-9_18)
15. Zhao, Z., Yan, D., Ng, W.: Mining probabilistically frequent sequential patterns in large uncertain databases. *IEEE Trans. Knowl. Data Eng.* **26**(5), 1171–1184 (2014). doi:[10.1109/TKDE.2D13.124](https://doi.org/10.1109/TKDE.2D13.124)
16. Shen, A.: Context-free grammars. In: Algorithms and Programming, Part of the series Springer Undergraduate Texts in Mathematics and Technology, pp. 221–242 (2010). doi:[10.1007/978-1-4419-1748-5_15](https://doi.org/10.1007/978-1-4419-1748-5_15)

17. Jeung, H., Shen, H.T., Zhou, X.: Mining trajectory patterns using hidden Markov Models. In: Song, I.Y., Eder, J., Nguyen, T.M. (eds.) Data Warehousing and Knowledge Discovery, DaWaK 2007. LNCS, vol. 4654, pp. 460–480 (2007). doi:[10.1007/978-3-540-74553-2_44](https://doi.org/10.1007/978-3-540-74553-2_44)
18. Jaroszewicz, S.: Using interesting sequences to interactively build Hidden Markov Models. *Data Min. Knowl. Disc.* **21**, 186–220 (2010). doi:[10.1007/s10618-010-0171-0](https://doi.org/10.1007/s10618-010-0171-0)