International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Clickstream Data Analysis Using Data Mining in R

Anubhuti Singh[1], Sandhya Tarar[2]

[1, 2]Gautam Buddha University, School of Information and Communication Technology, Greater Noida, India

Abstract: ***Clickstream analysis is to understand how user moves through website and in what order. It helps to extract data-driven user personalities, predict their actions, and extract frequent sequential patterns using clickstream data. Many social networking platforms rely on the data generated based on user's click path. To start analyzing clickstream data, we first need to be able to capture step-by-step a user's activity on a web page or application. And that is of great value in the hands of any internet marketer. Getting a 360-degree view of a customer by knowing what they are and are not clicking can bring you a huge improvement in both your products and your customers' experience.***

Keywords: Clickstream analysis, Markov chain, Clustering, cSPADE, Frequent Sequential Pattern

1. Introduction

Clickstream may be a generic term to explain visitors' ways through one or a lot of internet sites, an info path a user leave behind whereas visiting a web site. It's generally captured in semi-structured web site log files. A series of sites requested by a traveler in an exceedingly single visit is noted as a session. Clickstream knowledge [1] in an exceedingly web site may be an assortment of sessions within the site. Clickstream knowledge may be derived from raw page requests (referred to as hits) and their associated info (such as timestamp, IP address, URL, status, range of transferred bytes, referrer, user agent, and, sometimes, cookie data) recorded in net server log files. Analysis of clickstreams shows however an internet website is navigated and employed by its guests. in an exceedingly visualize web site atmosphere, clickstreams knowledge provides info essential to understanding the user's behavior, like what the user visit on net, Their gender, however frequent they visit a page. Analyzing such info embedded in clickstream knowledge is essential to enhance the user expertise. Interest in deciphering net usage knowledge in net server log files has spawned a lively marketplace for blog analysis tools that analyze, summarize, and visualize net usage patterns. [2]
In this paper, we tend to propose ways for interactive visual analysis of clickstream information, with a spotlight on understanding frequent paths visited by users.

a) Data preparation

Within the information world, refinement includes processing, cleaning, and transformation of the initial information into one thing convenient for the analysis you're going to carry out.

b) Model construction

As in most cases, the ways that are deployed for solving this drawback are several. during this paper, we are about to evaluate 2 of them, as they're widely used and simple to understand: Markov Chains work with serial information and clustering clickstream data, an oversized variety of monitored clickstreams makes the analysis harder unless we tend to cluster along similar clickstreams and user profiles.

c) Data Mining

Rather than modelling click stream information as transition probabilities, we will represent them as sequent patterns, we will then mine sequent patterns for locating those patterns that have selected minimum support, or in different words, occur a tiny low variety of times in user's clickstream information. [3]

Figure 1: Flow chart of methodology

2. Environment Setup

Linux setup is used for clustering of database and running of project. Linux is a stable OS for analysis of log files. Ubuntu 16.04 in Virtual box on Windows 10 as host is used. RStudio server is only compatible on linux environment.

R 3.5 and RStudio (latest) server are the software running on linux machine. R language is used for coding at every stage and for visualization

This paper is done on 3.5.5 version of R and RStudio-server with version 1.1.463 on 64 bit system. All the visualization is done on either on RStudio and Xlstat version 2015.5 which is compatible for windows 10.

3. Data Collection

With the flexibility to trace each step of a user's journey across an internet site or app, we will really get a 360° read of a client. Often, organizations gather this data via multiple analytics tools, every representing a distinct side of user behavior. several of these rely on information generated from once a user does (or doesn't) "click." we tend to call that clickstream data.

Either you have data in your data warehouse, or you need to catch live data from user's clicks on your website, you need to have a way to collect and store data consistently into a database.

We have clicks path of the users on an ecommerce website. The raw data of user's click path had data elements data elements such as:

Session id- that uniquely identifies the user
Date and time stamp
The visitor's IP address
The URLs of the pages visited,
Category of the website, and many Extra information

	A	B	C	D	E
1	1331582487	3/12/2012 13:01	66.91.193.75	http://www.acme.com/SH55126545/VD55179433	shoes
2	1331584835	3/12/2012 13:40	173.172.214.24	http://www.acme.com/SH55126545/VD55179433	shoes
3	1331585550	3/12/2012 13:52	173.172.214.24	http://www.acme.com/SH55126545/VD55179433	shoes
4	1331583407	3/12/2012 13:16	68.109.255.230	http://www.acme.com/SH55126545/VD55179433	shoes
5	1331583457	3/12/2012 13:17	68.109.255.230	http://www.acme.com/SH55126545/VD55179433	shoes

Figure 2: Sample Raw Data

4. Data Preparation

Raw data is like crude oil; It need to be refined before being truly valuable. In the data world, refinement includes data processing, cleaning, and transformation of the initial data into something that can be processed for analysis. In this Project, we would like to have our data grouped into sessions. It would be sensible, too, if we may organize the events of every session in time order before moving to the actual analysis. In the above description, it's necessary to outline what will we mean by the term "session." A session involves the consideration of 1 of the following: The time between 2 resulting application begin events within the case of an application. The time from entry till logout or timeout (i.e. twenty minutes of no activity) within the case of a web page.

In distinction to different knowledge sequences, clickstream data will have variable lengths completely different for various} sessions and different users.

In order to rework the initially collected event log into clickstream data, we want to:

- Identify events/actions performed by constant user and cluster them along.
- Split them more into subgroups of events primarily based what was performed throughout an equivalent session in line with the session's definition given higher than.

The Dataset we'll be using in further analysis looks something like this:

Session 1: movies clothing movies games computer Defer
Session 2: games computer movies games Defer
Session 3: electronic movies home&garden handbages home&garden handbages shoes handbages Buy
Session 4: outdoors clothing tools accessories tools grocery Defer
Session 5: computers home&garden tools home&garden movies games Defer
Session 6: games grocery accessories computers accessories movies computers movies games Defer
Session 7: automotive electronics automotive shoes shoes electronics movies handbags Buy
Session 8: outdoors clothing movies outdoors games movies Defer
Session 9: electronics shoes electronics handbags computers movies home&garden computers Defer

Figure 3: Sample Refined Data

In this representation, each line corresponds to a session, where the first field is the session number and following field are the actions performed in a particular session.

5. Model Construction

The above problem of studying user behavior by merely their actions performed on the internet can be solved by numerous methods, multiple techniques and algorithm are available to find frequent sequence and predict pattern of search. In this project we'll be using two of them ,which are widely used and easy to understand.

1) Markov Chains

Markov chains work with sequential data, the type we'll be dealing with in this paper. Markov chain is a stochastic model describing sequence of possible events with the probability of the next event depends only on the present state of the system and not on any previous states.

$S(T_n = i_n \mid T_{n-1} = i_{n-1}) = S(T_n = i_n \mid T_0 = i_0, T_1 = i_1 \ldots T_{n-1} = i_{n-1})$

Order of the Markov chain means the dependency on the history of actions. Low order Markov chain means that next state only depends on the current state. The higher-order Markov Chain introduced by the Raftery (1985) means more history is used to predict the next state, By choosing higher order we increase the memory and complexity but will lead to more realistic model. At the same time, the parameters needed for the representation increase exponentially, so it is important to choose the order wisely.

Console Terminal

```
First-Order Markov Chain

Transition Probabilities:

Lag:  1
lambda:  1
             Buy Defer accessories automotive clothing computer electronics
Buy            0     0        0.00        0.0     0.00     0.00         0.0
Defer          0     0        0.00        0.0     0.00     0.29         0.0
accessories    0     0        0.00        0.0     0.00     0.14         0.0
automotive     0     0        0.00        0.0     0.00     0.00         0.2
clothing       0     0        0.00        0.0     0.00     0.00         0.0
computer       0     0        0.33        0.0     0.00     0.00         0.0
electronics    0     0        0.00        0.5     0.00     0.00         0.0
games          0     0        0.00        0.0     0.00     0.00         0.0
grocery        0     0        0.00        0.0     0.00     0.00         0.0
handbags       0     0        0.00        0.0     0.00     0.00         0.2
home&garden    0     0        0.00        0.0     0.00     0.14         0.0
movies         0     0        0.33        0.0     0.67     0.43         0.4
outdoors       0     0        0.00        0.0     0.00     0.00         0.0
shoes          0     0        0.00        0.5     0.00     0.00         0.2
tools          0     0        0.33        0.0     0.33     0.00         0.0
```

Figure 4: Transition probabilities

Fitting the Markov Chain model results in transition probabilities matrice and the lambda parameters of the chain for each one of the lags, lags depend on the order of the markov chain (first order gives one lag) along with the start and end probabilities.(In Fig 4). Start and end probabilities

correspond to the probability that a clickstream will start or end with this specific event(In Fig 5).

```
Start Probabilities:

 automotive    computer electronics       games      movies    outdoors
       0.11        0.11        0.22        0.22        0.11        0.22

End Probabilities:

  Buy Defer
 0.22  0.78
```

Figure 5: Start and end probabilities

```
First-Order Markov Chain with 15 states.
 The Markov Chain has absorbing states.

Observations: 69
LogLikelihood: -74.64877
AIC: 181.2975
BIC: 217.0432
```

Figure 6: Summary of Markov Chain

2) Fitting a Markov chain

Fitting of Markov chain will lead to transition matrix depicting transition from the current to the next state.

With the plot markov chain function in R depicts zone transition that represents the transition matrix that gives the possible movement from the current state to the next state (Fig 7).

The transition probability matrix can be represented as a heat map (Fig 8) where the current state is on the y-axis and the next state is on the x-axis. The more darker tone of the color red, the more transition possibility is. For example, the transition from outdoors to clothing is very likely while the transition from games to clothing is not.

Theoretically, a Markov chain is a probabilistic automaton. The probability distribution of state transitions is typically represented as the Markov chain's transition matrix. If the Markov chain has M possible states, the matrix will be an M x M matrix, such that entry (i, j) is the probability of transitioning from state i to state j. Additionally, the transition matrix must be a stochastic matrix, a matrix whose entries in each row must add up to exactly 1.

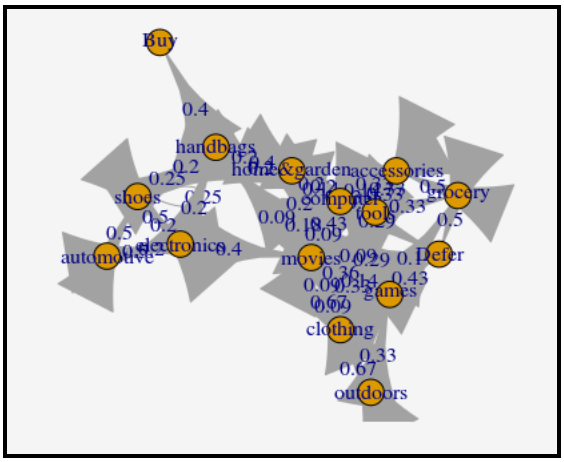

Figure 7: Zone transition

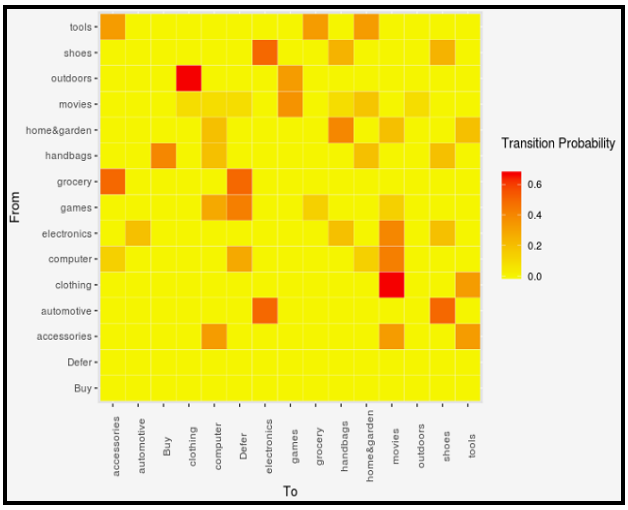

Figure 8: Heat map to show transition probability

A. Predicting clicks

In clickstream analysis, it is vital to predict the next click or final click (state) of a user given the current click path. This will give the detail study on the user's behavior on the net. Along with the click prediction, the transition probabilities are also considered.

```
Observations: 9

Click Frequencies:
accessories  automotive         Buy    clothing    computer       Defer
          3           2           2           3           7           7
electronics       games     grocery    handbags home&garden      movies
          5           7           2           5           5          11
   outdoors       shoes       tools
          3           4           3
```

Figure 9: Click frequencies

Fig 9 depicts the number of times a particular page is clicked by the user which shows the clicking frequencies.

```
> pattern<- new("Pattern", sequence= c("movies","outdoors"))
> resultPattern<- predict(mc, startPattern= pattern, dist=1)
> resultPattern
Sequence: clothing
Probability: 0.6666667
Absorbing Probabilities:
  None
1  NaN
```

Figure 10: Predicting the pattern

Fig 10 predicts the next sequence or item depending on the sequence or pattern provided. Here given sequence is movies and outdoors for which the resultant predicted pattern is clothing with probability being 0.66.

By analyzing the highest start probability and the transition with the most probability, we end up predicting the user behavior depending on this data.

B. Clustering clickstream data

In most cases, due to the complexity of websites or applications, probability of occurring same clickstreams is difficult as user can take multiple paths and it becomes difficult to study their behavior. Therefore we group together depending upon similar clickstreams and user profile.

Data mining techniques can be put to use to extract real time results from the large number of log files developed from the internet.

This process is beneficial for organization to study users similar interests and find customer segments.

In our paper, we performed a k-means clustering with two centers. A meaningful interpretation on what the clusters depict:

After clustering, we noticed that cluster by cluster, the common length of clickstreams increase. this suggests that k-means clustered the clickstreams supported the number of actions the user that made them performed throughout a session.

```
> clusters
[[1]]
Clickstreams

Session3: electronics movies home&garden handbags home&garden handbags shoes handbags
Buy
Session7: automotive electronics automotive shoes shoes electronics movies handbags Bu
y
Session9: electronics shoes electronics handbags computer movies home&garden computer
Defer

[[2]]
Clickstreams

Session1: movies clothing movies games computer Defer
Session2: games computer movies games Defer
Session4: outdoors clothing tools accessories tools grocery Defer
Session5: computer home&garden tools home&garden movies games Defer
Session6: games grocery accessories computer accessories movies computer movies games
Defer
Session8: outdoors clothing movies outdoors games movies Defer
```

```
Total SS: 27.72222
Within SS: 7 15.16667
Total Within SS: 22.16667
Between SS: 5.555556
```

Figure 11: Clustering

In Fig 11 after clustering through K-means algorithm we can see that 2 clusters are made depending on the type of clicks.

6. Data Mining

For better understanding of user's actions we can model clickstream data as sequential patterns instead of transition probabilities. By mining sequential patterns we can find patterns with a particular minimum support as we mention, that is the small number of times that pattern occurs in user's clickstream data. The SPADE (**S**equential **Pa**ttern **D**iscovery using **E**quivalence classes) algorithm uses a vertical id-list database format, where we associate each sequence a list of objects in which it occurs. Then, frequent sequences can be found efficiently using intersections on id-lists. The method also reduces the number of databases scans, and therefore also reduces the execution time, the algorithm computes the frequencies of sequences with only one item, in the second step with two items and so on.

Figures below gives details of the minimum support taken which is 0.4.

```
parameter specification:
support : 0.4
maxsize :  10
maxlen  :  10

algorithmic control:
bfstype  : FALSE
verbose  :  TRUE
summary  : FALSE
tidLists :  TRUE

preprocessing ... 1 partition(s), 0 MB [0.053s]
mining transactions ... 0 MB [0.083s]
reading sequences ... [0.085s]

total elapsed time: 0.221s
```

Figure 12: Support details

```
> summary(sequences)
set of 23 sequences with

most frequent items:
      Defer    computer       games      movies
         11          10          10           8
home&garden     (Other)
          2           4

most frequent elements:
   {Defer} {computer}    {games}   {movies}
        11          10          10           8
{clothing}    (Other)
         2           4

element (sequence) size distribution:
sizes
 1  2  3  4
 7 10  5  1
```

Figure 13: Sequence summary

```
sequence length distribution:
lengths
 1  2  3  4
 7 10  5  1

summary of quality measures:
    support
 Min.   :0.4000
 1st Qu.:0.4000
 Median :0.4000
 Mean   :0.4957
 3rd Qu.:0.6000
 Max.   :0.8000

includes transaction ID lists: TRUE

mining info:
  data ntransactions nsequences support
 trans            34          5     0.4
```

Figure 14: Summary of frequent sequences

The above figure shows the details of the frequent sequence produce by mentioning most frequent items , most frequent sequence , sequence size and length distribution and other quality measures.

Further n-sequences can be formed by combining (n-1)-sequences using their id-lists. The size of the id-lists can be seen by the number of sequences having that item. If this number is greater than minSup (minimum support), then the sequence is said to be frequent. The algorithm terminates when no further frequent sequences is to be found. The algorithm can either use breadth first search(BFS) or depth first search(DFS) to find frequent sequences.

It is observed that cSPADE found many nonzero sequences from user actions.

For example, it has found many singular sequences, such as <{Movies}>, <{games}>, <{Computer}>, among others. These singular sequences are the ones which are frequently used, but they may not be useful in the particular application, referred to as Tag Recommendation.

With this algorithm any organization can predict, understand user's path through its website or application.
The patterns with minimum support are extracted:

	sequence	support
1	<{clothing}>	0.4
2	<{computer}>	0.6
3	<{Defer}>	0.8
4	<{games}>	0.6
5	<{home&garden}>	0.4
6	<{movies}>	0.8
7	<{tools}>	0.4
8	<{computer},{movies}>	0.4
9	<{home&garden},{home&garden}>	0.4
10	<{computer},{games}>	0.4
11	<{movies},{games}>	0.6
12	<{computer},{movies},{games}>	0.4
13	<{clothing},{Defer}>	0.4
14	<{computer},{Defer}>	0.6
15	<{games},{Defer}>	0.6
16	<{movies},{Defer}>	0.6
17	<{tools},{Defer}>	0.4
18	<{movies},{games},{Defer}>	0.6
19	<{computer},{movies},{Defer}>	0.4
20	<{games},{computer},{Defer}>	0.4

Figure 15: Support details

Fig 14 shows the 23 sequences produced with their minimum support.

For a given sequence pattern X, we can predict the next click by searching for the pattern sequence with the highest support that starts with X.

For example, after having just performed (Computer), the most probable next action is (Defer) according to pattern sequence 14 with support 60% which is higher than pattern sequences of 8, 10, 12, 19 being 40% each as shown in Fig 14.

Of course, lowering the support would give us pattern sequences that are less frequent in our clickstreams.

7. Result Analysis

The markov chains are fitted on each clickstreams and found out the transition probabilities of the elements. By K-means clustering algorithm we clustered the clickstreams into two clusters, Users that are frequently visiting the website and the users who surf less on a particular website. Through clickstream data we can also predict the user's next click based on the frequent sequences.

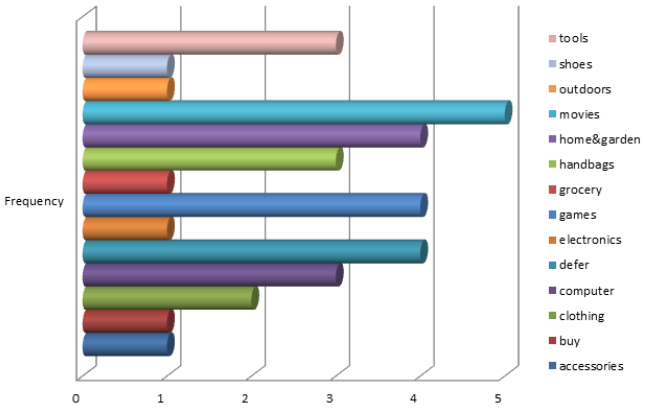

Figure 16: Most frequent items

In the above figure It shows the most frequent items in the sequences. As shown movies are most frequently visited, grocery page is one of the least visited.

Instead of representing clickstream data into transition probabilities we can demonstrate by sequential patterns, through cSPADE mining technique keeping minimum support to 0.4 we developed sequence patterns and their corresponding minimum support and predict the most favorable next click for a particular item by going to its next highest minimum support.

SPADE algorithm is concerned with finding relevant sequential pattern between data samples where the values are represented as sequences.

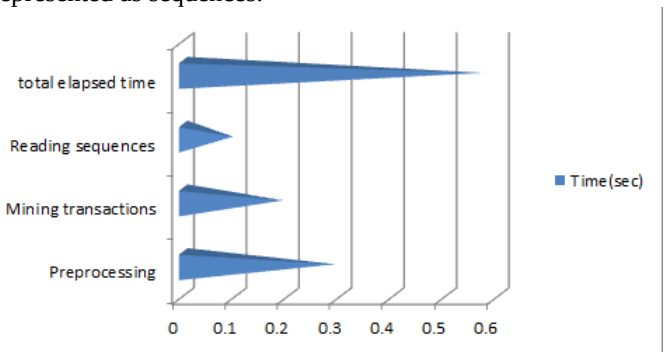

Figure 17: Average performance of cSPADE

The above figure shows the average performance of cSPADE algorithm on the given dataset to mine the frequent sequential patterns with their support values. cSPADE algorithm in web click log analysis is most used domain of frequent sequential mining. It has an advantage over apriori based algorithms as it reduces the database scans and hence reduces the execution time. Thus increasing its efficiency. This technique can help multiple organization which are user driven to understand their user's and visitors can know more about their product as to which pages are visited more and which are less travelled.

8. Conclusion

In this paper we explored an ecommerce website by the clickstream data on the website through Markov chain and mined using cSPADE algorithms on the clickstream sequence patterns. By implementing the proposed model, we

can conclude that: Extract user's behavior for the most frequent click path on the website or application, Predicted next click on the bases of first order Markov chain, Finally through mining extracted frequent sequential patterns. Based on the above result we can carry out rigorous process of reviewing the data driven behavior of the customers by initializing strategies independent of the web or application design.

References

[1] X. Cheng, R.Wu, Clustering path profiles on a website using rough k-means method, Journal of Computational Information Systems 8 (14) (2012) 6009–6016.
[2] X. Cheng, R.Wu, Clustering path profiles on a website using rough k-means method, Journal of Computational Information Systems 8 (14) (2012) 6009–6016.
[3] P. Lingras, R. Yan, C. West, Comparison of conventional and rough k-means clustering, Proceedings RSFDGrC – 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, LNAI, vol. 2639, Springer-Verlag, Berlin, Germany, 2003, pp. 130–137.
[4] Scholz, Michael 2017/01/01, R Package Clickstream-Analyzing Clickstream Data with Markov Chains, forth-coming 10.18637/jss.v074.i04, Journal of statistical software.
[5] J.Jarvis and D.Shier. Graph-theoretic analysis of finite markov chains, In Discrete Mathematics and Its Applications, CRC Press, nov 1999.
[6] A.Spedicato. markovchain: an R Package to Easily Handle Discrete Markov Chans, 05 2017. Rpackage version0.6.5.
[7] Visser and M.Speekenbrink. depmixS4: AnRPackage for hidden markov models. Journal of Statistical Software, 36(7),2010.
[8] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis), 2011.
[9] D. Oelke, D. Spretke, A. Stoffel, and D. A. Keim. Visual readability analysis: How to make your writings easier to read. IEEE Transactions on Visualization and Computer Graphics, 18(5):662–674, 2012.
[10] M. J. Zaki, "Parallel and distributed association mining: A survey," IEEE Concurrency, vol. 7, no. 4, pp. 14–25, Oct./Dec. 1999.
[11] Goulet, C. Dutang, M. Maechler, D. Firth, M. Shapira, M. Stadelmann, and expm-developers@lists.R-forge.R-project.org. *expm: Matrix Exponential*, 2015.
[12] G. Cassandras. Discrete event systems: modeling and performance analysis. CRC, 1993. Chambers, J.M. Software for Data Analysis: Programming with R. Statistics and computing. SpringerVerlag, 2008. ISBN 9780387759357.
[13] Peng, W.-C., Liao, Z.-X., 2009. "Mining sequential patterns across multiple sequence databases". Data Knowl. Eng. 68 (10), 1014–1033.
[14] Li, K.-C., Chang, D.-J., Rouchka, E. C., Chen, Y. Y., 2007. "Biological sequence mining using plausible neural network and its application to exon/intron boundaries prediction". In: CIBCB. IEEE, pp. 165–169.
[15] Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: Agrawal, R., Srikant, R. (eds.) Proceedings of 1995 International Conference Data Engineering, ICDE 1995, pp. 3–14 (1995).
[16] Yu, X., Li, M., Gyu Lee, D., Deuk Kim, K., Ho Ryu, K.: Application of Closed Gap-Constrained Sequential Pattern Mining in Web Log Data.
[17] Richard Serfozo (24 january 2009).*Basics of Applied Stochastic Processes.* Springer Science & Business Media.p. 2. ISBN 978-3-540-89332-5. Archived from the original on 23 March 2017.
[18] Wan, Lijie; Lou, Wenjie; Abner, Erin; Kryscio, Richard J. (2016)."A comparison of time-homogeneous Markov chain and Markov Process multi-state models". *Communications in Statistics: Case Studies, Data Analysis and Applications.*