Synthetic Sequence Generator for Recommender Systems – Memory Biased Random Walk on a Sequence Multilayer Network

Nino Antulov-Fantulin[1], Matko Bošnjak[2,⋆], Vinko Zlatić[3], Miha Grčar[4], and Tomislav Šmuc[1]

[1] Laboratory for Information Systems, Division of Electronics, Rudjer Bošković Institute, Zagreb, Croatia
nino.antulov@irb.hr
[2] Department of Computer Science, University College London, London, UK
[3] Theoretical Physics Division, Rudjer Bošković Institute, Zagreb, Croatia
[4] Department of Knowledge Technologies - E8, Jožef Stefan Institute, Ljubljana, Slovenia

Abstract. Personalized recommender systems rely on each user's personal usage data in the system, in order to assist in decision making. However, privacy policies protecting users' rights prevent these highly personal data from being publicly available to a wider researcher audience. In this work, we propose a memory biased random walk model on a multilayer sequence network, as a generator of synthetic sequential data for recommender systems. We demonstrate the applicability of the generated synthetic data in training recommender system models in cases when privacy policies restrict clickstream publishing.

Keywords: biased random walks, recommender systems, clickstreams, networks.

1 Introduction

Recommender systems provide a useful personal decision support in search through vast amounts of information on the subject of interest [1, 2] such as books, movies, research papers, and others. The operation and the performance of recommender systems based on collaborative data [3, 4] are necessarily tied to personal usage data, such as users' browsing and shopping history, and to other personal descriptive data such as demographical data. These data often conform to privacy protection policies, which usually prohibit their public usage and sharing, due to their personal nature. This, in turn, limits research and development of recommender systems to companies in possession of such vital data, and prevents performance comparisons of new systems between different research groups.

⋆ Part of this research was done while the author was at the Rudjer Bošković Institute.

S. Džeroski et al. (Eds.): DS 2014, LNAI 8777, pp. 25–36, 2014.

In order to enable data sharing and usage, many published data sets were anonymized by removing all the explicit personal identification attributes like names and demographical data, among others. Nevertheless, various research groups managed to successfully identify personal records by linking different datasets over quasi-personal identifiers such as search logs, movie ratings, and other non-unique data, revealing as a composition of identifiers [5]. Due to successful privacy attacks, some of the most informative data for recommendation purposes, such as the personal browsing and shopping histories, are put out of the reach of the general research public. In their original form, usage histories are considered personal information, and their availability is heavily restricted. However, even with the personal information obfuscated, they remain a specific ordered sequence of page visits or orders, and as such can be uniquely tied to a single person through linkage attacks.

With usage histories often rendered unavailable for public research, recommender systems researchers have to manage on their own and often work on disparate datasets. Recently, a one million dollar worth Overstock.com recommender challenge released synthetic data, which shares certain statistical properties with the original dataset. The organizers noted that this dataset should have been used only for testing purposes, while the code itself had to be uploaded to RecLabs[1] for model building and evaluation against the real data. The challenge ended with no winner since no entry met the required effectiveness at generating lift. It would be useful both for contestants and the companies, if the synthetic data could be used for recommendation on real users.

We propose an approach to synthetic clickstream generation by constructing a memory biased random walk model (MBRW) on the graph of the clickstream sequences, which is a subclass of Markov chains [6, 7]. Random walks [8–10] have been used for constructing recommender systems on different types of graph structures originating from users' private data, but not to generate synthetic clickstreams. In this work we show that the synthetic clickstreams generated by the MBRW model share similar statistical properties to real clickstream. In addition, we use the MBRW model to generate synthetic clickstreams for the VideoLectures.NET[2] dataset from the ECML/PKDD 2011 Discovery Challenge [11] and publish it on-line. Finally, we demonstrate that synthetic data could be used to make recommendations to real users on the Yahoo! Music dataset released for the KDDCup challenge for the year 2011 [12] and the MovieLens dataset [3].

2 Methodology

The biased random walk on a graph [13, 14] is a stochastic process for modelling random paths on a general graph structure In our case, the graph we refer to is constructed from users' interaction history, i.e. clickstreams. Clickstream is a sequence of items (path on graph) $c^i = \{u_1^i, u_2^i, u_3^i, ..., u_n^i\}$, such as web pages,

[1] http://code.richrelevance.com/reclab-core/
[2] http://videolectures.net
[3] http://grouplens.org/datasets/movielens/

movies, books, etc., a user i interacted with. The set of all the clickstreams in a system is $C = \{c^1, c^2, ..., c^i, ..., c^m\}$. This set is usually used to generate an item history matrix, which is used by a recommender system algorithm for recommendation learning. In our work, we use two characteristic data generator matrices, obtained from the real clickstream data: the Direct Sequence matrix (DS) and the Common View Score matrix (CVS). The element $DS[m, n]$ of the matrix DS denotes the number of clickstreams in C in which the web page m immediately follows the web page n. The element $CVS[m, n]$ of the matrix CVS denotes the number of occurrences in which the web page m and the web page n belong to the same clickstream in C. In order to reconstruct synthetic clickstreams from these matrices, we introduce the memory component to the biased random walk [13], and obtain the memory biased random walk model.

The MBRW model is a discrete time Markov chain model, with a finite memory of m past states. Biases from the DS graph are the connecting probability of choosing the next item with respect to the current item, while biases from the CVS graph are the connecting probability of choosing the next item with respect to the the past m items in a clickstream. The initial vertex for the random walk can be chosen by either a stochastic or a deterministic rule.

Given an initial vertex u_1, the probability of choosing an adjacent vertex u_2 equals:

$$P(u_2|u_1) = \frac{DS_{u_2,u_1}}{\sum_k DS_{k,u_1}} \tag{1}$$

which, generates a clickstream $c^i = \{u_1, u_2\}$. The third vertex, u_3 in the clickstream is chosen with a probability of:

$$P(u_3|u_2, u_1) = \frac{DS_{u_3,u_2} CVS_{u_3,u_1}}{\sum_k DS_{k,u_1} CVS_{k,u_1}} \tag{2}$$

thus generating a clickstream $c^i = \{u_1, u_2, u_3\}$. Using a finite memory of size m, we choose the vertex u_n with the probability of:

$$P(u_n|u_{n-1}, ..., u_{n-m-1}) = \frac{DS_{u_n,u_{n-1}} \prod_{k=1}^{m} CVS_{u_n,u_{n-k-1}}}{\sum_j DS_{j,u_{n-1}} \prod_{k=1}^{m} CVS_{j,u_{n-k-1}}} \tag{3}$$

thus generating a clickstream $c^i = \{u_1, u_2, u_3, ..., u_n\}$ at the n-th step of the random walk.

The intuition behind (3) is that the probability of choosing the next item should be proportional to the product of direct sequence frequency DS and common view score frequencies CVS in the clickstream data. Direct sequence frequency DS measures the tendency of the current item preceding the next item in the clickstream data. The product of common view score frequencies measures the tendency of the next item appearing together with all the other items in a currently generated clickstream. The denominator of (3) is the normalization expression. In Figure 1, we demonstrate the transition probability calculation on a simple example.

We use the aforementioned MBRW model in a generative manner for constructing synthetic clickstreams. The procedure of generating a single clickstream

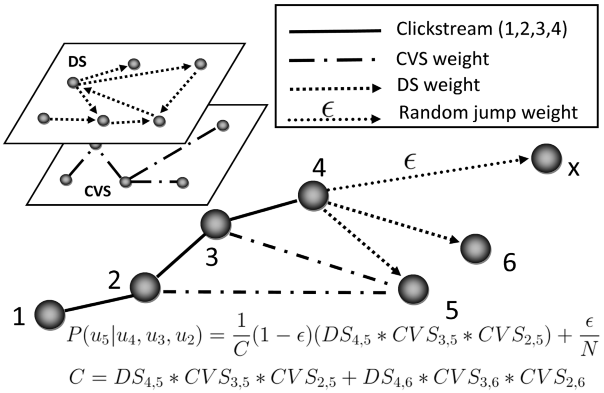

Fig. 1. Simple example: at the current step the MBRW model ($m = 2$) has created a clicktream (u_1, u_2, u_3, u_4) and node u_4 has two neighbouring nodes u_5 and u_6 at the DS graph. The transition probability (see formula 3) to node u_5 is given, where the ϵ transition denotes the probability of a jump to some arbitrary node u_x, the C denotes the normalization, the factor $\frac{\epsilon}{N}$ denotes the probability of random jump back to node u_5 and N denotes total number of nodes in DS graph.

starts by randomly generating the first item. We then sample the length of the clickstream l from a discrete probability distribution L like Poisson, negative binomial, geometric, or from the real clickstream length distribution, if available. The next step is to iteratively choose the next $l-1$ items with the MBRW model. In order to ensure additive smoothing over transition probabilities in the MBRW walk, we introduce a small ϵ probability of a random jump. At each step in the clickstream generation process a random walker produces a jump at some random item with the probability ϵ. This ϵ-smoothing technique turns all possible clickstreams to become non-forbidden in generation process. At the end of this process, the random walk path $c^i = \{u_1^i, u_2^i, ..., u_l^i\}$ presents one clickstream which is then appended to the synthetic clickstream set C^*. This whole clickstream generation process is iterated in K independent iterations to produce K synthetic clickstreams. The pseudo code for Memory Biased Random Walks with Random Jumps is provided in Algorithm 1. The code for the MBRW model is available on GitHub.[4]

3 Evaluation and Results

We analyse the statistical properties as well as the utility of the synthetic data in training recommender system models. In our experiments we used three datasets: (i) the Yahoo! Music dataset released for the KDDCup challenge for the year 2011 [12], (ii) the MovieLens 1M dataset and (iii) the VideoLectures.NET dataset

[4] `http://github.com/ninoaf/MBRW`

Algorithm 1. Memory Biased Random Walks with Random Jumps

Input: DS - Direct Sequence matrix, CVS - Common View Score matrix, K - number of synthetic clickstreams, ϵ - probability of random jump, m - memory length from prob. distr. M, L - clickstream length distribution
Output: $C^* = \{c^{*1}, c^{*2}, ..., c^{*K}\}$ synthetic clickstream set
$C^* = \emptyset$
for $i = 1 : \mathrm{K}$ **do**
  $c_i^* \sim \{u_1, u_2, ..., u_k\}$ // sample the initial item;
  $l \sim L$ // sample the clickstream length
  for $j = 2 : \mathrm{l}$ **do**
    with $1 - \epsilon$ probability choose the next item u_j with MBRW walk on DS and CVS by using (3), otherwise with ϵ probability choose the next item u_j with a random jump;
    append new item: $c_i^* = c_i^* \cup u_j$
  end for
  append new synthetic clickstream: $C^* = C^* \cup c_i$
end for

from the ECML/PKDD 2011 Discovery Challenge [11]. As the privacy policies did not restrict publishing user preference data to particular items in both the KDDCup challenge 2011 and the MovieLens 1M dataset, we used them in our study as an experimental polygon to measure the performance of recommender systems models trained on synthetic data. Contrary, in the ECML/PKDD 2011 Discovery Challenge [11], only the content data and clickstrem statistics could be published but not the actual clickstreams. Therefore, we used our methodology on the VideoLectures.NET dataset to create and publish synthetic clickstream data. The first dataset used in our experiments is a subset of the Yahoo! Music dataset released for the KDDCup challenge for the year 2011 [12] which contains user preferences to particular musical items in a form of ratings, along with an appropriate time stamp. We extracted from this dataset a subset that represents a very good proxy for a set of sequential activities (clickstreams). For each user in our subset we retained a sequence of highly rated items in ascending order over time stamps (sequence activity or clickstream proxy). We limited the total number of items and users in our subset to 5000 and 10000 respectively, in order to be able to perform large set of computational experiments with resources on disposal. The reduced dataset, denoted with C, represents a set of clickstreams for 10000 users. This dataset reduction should not have any significant impact on the results and conclusions of the study. We will address this question later with the cross-validation technique. The second dataset contains approximately 10^6 anonymous ratings of approximately 3900 movies made by 6040 MovieLens users who joined MovieLens in 2000. Per each user, we extracted a sequence of highly rated items in the ascending order over time stamps from this dataset.

Our first hypothesis is that, given a sufficiently large synthetic dataset, basic statistical properties of DS^* and CVS^* matrices are preserved. We examined how statistical properties of the item preference matrix like DS and CVS are preserved in synthetic clickstream set, with respect to the original clickstream

set. We calculated the DS and CVS matrices from the C dataset and created the synthetic clickstream set C^* by using the MBRW model. Memory parameter m was sampled from the Gaussian distribution $\mathcal{N}(3, 2^2)$, number of random walk hops parameter l was sampled from $\mathcal{N}(9, 2^2)$ and number of synthetic clickstreams parameter K varying from 10^4 to 10^6. Upon obtaining the synthetic clickstream set C^*, we calculated the DS^* and CVS^* matrices, and compared their statistical properties to the original matrices DS and CVS. We used the Spearman's rank correlation [15] measure between the corresponding rows in (DS, DS^*) and (CVS, CVS^*).

Table 1. Average rank correlation between (DS,DS^*) and (CVS,CVS^*) for different sizes (K) of generated synthetic clickstream set. Synthetic clickstream set is created using parameter m sampled from $\mathcal{N}(3, 2^2)$, parameter l sampled from $\mathcal{N}(9, 2^2)$.

Size	$AVG[r(DS, DS^*)]$	$STD[r(DS, DS^*)]$
$K = 10^4$	0.5700	0.3210
$K = 10^5$	0.8914	0.2224
$K = 10^6$	0.9294	0.0590
	$AVG[r(CVS, CVS^*)]$	$STD[r(CVS, CVS^*)]$
$K = 10^4$	0.4545	0.2677
$K = 10^5$	0.6050	0.2120
$K = 10^6$	0.7361	0.1784

Due to the fact that these matrices are sparse and that in the process of recommendation only top ranked items are relevant, we limited the rank correlation calculation to the first $z = 100$ elements. Rank correlation between complete rows would be misleadingly high due to row sparsity. Average rank correlation coefficient $AVG[r(DS, DS^*)] = 0.92$ and $AVG[r(CVS, CVS^*)] = 0.73$ over all corresponding rows was obtained for the first z most important elements, with the above parameters and $K = 10^6$. The rank correlation coefficients for different values of parameter K can be seen in Table 1. This shows highly correlated statistical properties (DS, DS^*) and (CVS, CVS^*).

Now, we analyse the ability to learn recommender system models from synthetic data and apply this model on real users. We measure and compare the recommender system models learned on real, synthetic and random data, and their corresponding performance on recommending items to real users. We take the standard Item-Knn [16] recommender system as a representative of similarity-based techniques and a state-of-the-art matrix factorization technique, namely Bayesian Personalized Ranking Matrix Factorization Technique [17]. We hypothesise that learning recommender systems models even from the synthetic data can help making predictions to real users.

In order to create proper training, query and test data for testing of our hypotheses, we create two splits: a vertical and horizontal split. The horizontal split of the clickstream dataset C randomly divides them to two disjoint,

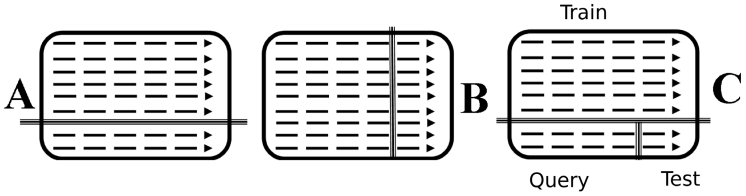

Fig. 2. Three ways of splitting the original clickstream set used in computational experiments: A - Horizontal split, B - Vertical split and C - Horizontal and vertical split

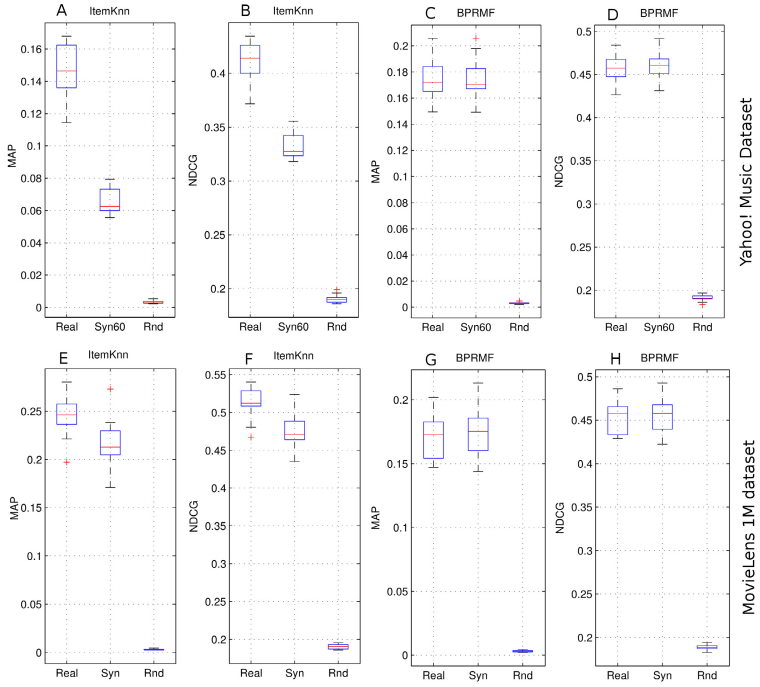

Fig. 3. Results for 10-folds cross-validation for MAP and NDCG measures for different datasets with Item-Knn [16] (plots A, B, E and F) and BPRMF [17] algorithm (plots C, D, G and H). Label "Real" represents performance on real dataset. Label: "Syn" represents synthetic data using MBRW with m sampled from $\mathcal{N}(3, 2^2)$, l sampled from real length distribution, $\epsilon = 0.0001$. Label "Rnd" represents random data generated by random jumps $\epsilon = 1.0$ on item graph. Plots: A, B, C and D are experiments on the Yahoo! Music dataset and plots: E, F, G and H are experiments on the Movielens 1M dataset. One can notice that results obtained with the Item-Knn and synthetic data are much lower than using real training data. This behaviour is more pronounced for the Yahoo! Music dataset, which is more sparse than the MovieLens 1M dataset. This confirms the hypothesis that Item-Knn is more sensitive to changes to the noise in local data distribution.

fixed-size clickstream sets C_{train} and C_{test}. Using the horizontal split on the Yahoo! Music dataset and the Movielens 1M dataset, we produced a training set C_{train} and then used the vertical split on the rest of the data to get the query set (C_{query}) and a test set (C_{test}). The vertical split, divides clickstream in C into two sets: first 50% of items are appended to first set C_{query}, whereas the rest of the clickstream items belong to a second set C_{test}. These splits are graphically represented in Figure 2. Experimental procedure is the following. We extract DS and CVS statistics from C_{train} and generate synthetic C^*_{train} with the MBRW model. The baseline random synthetic dataset C^*_{RND} is created by setting the parameter $\epsilon = 1$ (random jump model). Now, we create three different recommender system models: M (real model), M^* (synthetic model), and M_{RND} (random model) from the C_{train}, C^*_{train} and C^*_{RND}, respectively. Then recommender models for the input of real users C_{query} produce recommendations which are compared to C_{test} (ground truth). The performance on C_{test} is measured with the standard information retrieval measures: MAP [15] (Mean Average Precision) and NDCG [15] (Normalized Discounted Cumulative Gain: ranking measure). In order to estimate how performance results can generalize to independent datasets we use 10-fold cross-validation. Then in each cross-validation round we generate C^i_{train}, C^i_{test} and C^i_{query}. For each C^i_{train} we generate synthetic C^{i*}_{train} and random dataset C^{i*}_{RND}. Note that in each round recommender algorithms learn model on C^i_{train}, C^{i*}_{train} and C^{i*}_{RND} but their performance is measured for new users C^i_{query} on C^i_{test}. In Figure 3, we observe that BPRMF and Item-Knn models performed significantly better than baseline random models. We used the recommender system[5] implementation from the Recommender System extension [18–20] in RapidMiner. Furthermore, we notice that Item-Knn recommender is more sensitive to synthetic data than the BPRMF recommender system. Detailed analysis of this effects are out of scope of this work, but our hypothesis is that this behaviour of the Item-Knn algorithm is a consequence of well known high sensitivity of nearest neighbour approach to local properties of data and noise in the data. Contrary to this, the BPRMF algorithm is based on a low rank matrix factorization approximation which seems to produce same latent factors from synthetic and real data.

In the end, we focus on the ECML/PKDD 2011 Discovery Challenge [11], where the privacy policies have restricted public availability of users clickstream data on the VideoLectures.Net. Note, that here we did not have real clickstreams but only DS and CVS statistics. This challenge provided rich content data about items in a system and different statistics about users clickstream sequences. This motivated us to use the direct sequence statistics and common view statistics as generators of synthetic clickstreams with the proposed MBRW model. Direct sequence graph DS from this dataset consists of 7226 vertices in a single large, weakly connected component and common view score undirected graph CVS from this dataset consists of 7678 vertices in a large connected component.

[5] Item-Knn with $k = 20$ and BPRMF with num. factors: 10, user, item and negative Item regularization: 0.025, iterations: 30, learn rate: 0.05, initial mean: 0.0, initial std: 0.1 and fast sampling: 1024.

We produced and published[6] 20000 synthetic clickstreams for VideoLectures.net with the MBRW model with the memory parameter $m = 5$ and clickstream length L sampled from as a Geometric distribution with parameter 0.1 (expected length of clickstreams is 10).

4 Related Work and Discussion

The problems of privacy-preserving data publishing [21, 22] and privacy preserving data mining [23] are intensively researched within the database, the statistical disclosure, and the cryptography communities. Recently, a comprehensive survey [24] on the privacy challenges and solutions in privacy-preserving data mining has been published. Different privacy protection models already exists and here we will only mention the important ones.

Record linkage models like k-Anonymity model [25, 26] assure that the number of records with a quasi-identifier id is at least k and therefore assure the value of linkage probability of at most $1/k$. Attribute linkage models like L-diversity [27] are envisioned to overcome the problem of inferring sensitive values from k anonymity groups by decreasing the correlations between the quasi-identifiers and the sensitive values. Probabilistic models like ϵ-differential privacy model [28] ensure that individual's presence or absence in the database does not effect the query output significantly. Post-random perturbation (PRAM) methods [29, 30] change original values through probabilistic mechanisms and thus, by introducing uncertainty into data, reduce the risk of re-identification. Aggarwal et. al. [31] proposed an anonymization framework for string-like data. They used the condensation-based techniques to construct condensed groups and their aggregate statistics. From the aggregate statistics, they calculated the first and the second order information statistics of symbol distributions in strings, and generated synthetic, pseudo-string data. But still, many data-privacy researchers agree that high dimensional data poorly resist to de-anonymization [5] which poses privacy issues for companies, and prevent the usage of real-life datasets for research purposes.

Contrary to standard anonymization methods, synthetic data generation is an alternative approach to data protection in which the model generates synthetic dataset, while preserving the statistical properties of the original dataset. Several approaches for synthetic data generation have been proposed: (i) synthetic data generation by multiple imputation method [32], (ii) synthetic data by bootstrap method [33] (estimating multi-variate cumulative probability distribution, deriving similar c.d.f., and sampling a synthetic dataset), (iii) synthetic data by Latin Hypercube Sampling [34], (iv) and others such as a combination of partially synthetic attributes and real non-confidential attributes [35, 36]. These synthetic data generation strategies were mostly developed for database records with a fixed number of attributes but not for sequence data.

We proposed a novel approach for synthetic sequence generation by constructing the memory biased random walk (MBRW) model on the multilayer network

[6] `http://lis.irb.hr/challenge/index.php/dataset/`

of user sequences. Moreover, we demonstrated that this synthetic data can be used for learning recommender models which can be useful for applications on real users.

What are the potential privacy breach problems of our approach? Our method is based on the assumption that the sequence statistics: direct sequence DS and common view score CVS can be publicly available without breaking privacy of particular user. Why this is the case? We can view the clickstreams as a different way of writing the sequence statistics like finite state machines represent finite way of coding the infinite set of word from some regular language [37]. Note that the privacy breach can occur in a situation when the attacker can claim that individual unique synthetic subsequences could only be generated by using the unique transitions from particular user u. This is the reason why we need smoothing procedure (ϵ jumps) or k-anonymity filtering over the transition matrices DS and CVS. The ϵ random jumps in the generation process with small ϵ probability correspond to the additive smoothing of transition probabilities in MBRW model. Let us define the set of all possible combinatoric combinations of clickstreams with arbitrary length from set of items with Ω (infinite). Note that when $\epsilon = 0$ the MBRW model cannot create arbitrary clickstreams from the space of all clickstream combinations Ω due to the existence of zero values in DS and CVS matrices. As the additive smoothing technique turns all combinatoric clickstreams from Ω set possible, the attacker cannot claim that a certain unique user subsequence was used in the generation process. K-anonymity filtering can also be applied to CVS and DS directly by filtering all frequencies that are lower than k. This filtering enables that the presence or absence of individual transitions in DS or CVS cannot be detected. Therefore if the DS and CVS statistics can be publicly available without breaking privacy, our methodology can be applied.

5 Conclusion

The principle aim of our work was to construct a generator of real-like clickstream datasets, able to preserve the original user-item preference structure, while at the same time addressing privacy protection requirements. With respect to this aim, we investigated properties of the memory biased random walk model. We demonstrated that the basic statistical properties of data generators DS and CVS matrices are preserved in the synthetic dataset if we generate sufficiently large datasets. In addition, we demonstrated that the synthetic datasets created with it can be used to learn recommender system models applicable on real users.

Acknowledgments. This work was partially supported by the European Community 7^{th} framework ICT-2007.4 (No 231519) "e-LICO: An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science", partially by the EU-FET project MULTIPLEX (Foundational Research on MULTIlevel comPLEX networks and systems, grant no. 317532) and partially by the Croatian Science Foundation under the project number I-1701-2014.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Rendle, S., Tso-Sutter, K., Huijsen, W., Freudenthaler, C., Gantner, Z., Wartena, C., Brussee, R., Wibbels, M.: Report on state of the art recommender algorithms (update). Technical report, MyMedia public deliverable D4.1.2 (2011)
3. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
4. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW 1994, pp. 175–186 (1994)
5. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: Proceedings of the 2008 IEEE Symposium on Security and Privacy, SP 2008, pp. 111–125 (2008)
6. Feller, W.: An introduction to probability theory and its applications, vol. 2. John Wiley & Sons (2008)
7. Kao, E.: An introduction to stochastic processes. Business Statistics Series. Duxbury Press (1997)
8. Bogers, T.: Movie recommendation using random walks over the contextual graph. In: Proceedings of the 2nd Intl. Workshop on Context-Aware Recommender Systems (2010)
9. Fouss, F., Faulkner, S., Kolp, M., Pirotte, A., Saerens, M.: Web recommendation system based on a markov-chain model. In: International Conference on Enterprise Information Systems, ICEIS 2005 (2005)
10. Gori, M., Pucci, A.: Research paper recommender systems: A random-walk based approach. In: Web Intelligence, pp. 778–781 (2006)
11. Antulov-Fantulin, N., Bošnjak, M., Žnidaršič, M., Grčar, M., Morzy, M., Šmuc, T.: ECML/PKDD 2011 Discovery Challenge overview. In: Proceedings of the ECML-PKDD 2011 Workshop on Discovery Challenge, pp. 7–20 (2011)
12. Dror, G., Koenigstein, N., Koren, Y., Weimer, M.: The Yahoo! music dataset and kdd-cup'11. In: Proceedings of KDD Cup 2011 (2011)
13. Zlatić, V., Gabrielli, A., Caldarelli, G.: Topologically biased random walk and community finding in networks. Phys. Rev. E 82, 066,109 (2010)
14. Newman, M.: Networks: An Introduction. Oxford University Press, Inc. (2010)
15. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval Cambridge University Press (2008)
16. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Transactions on Information Systems 22(1), 143–177 (2004)
17. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009, pp. 452–461 (2009)
18. Gantner, Z., Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Mymedialite: A free recommender system library. In: Proceedings of the Fifth ACM Conference on Recommender Systems, pp. 305–308 (2011)
19. Mihelčić, M., Antulov-Fantulin, N., Bošnjak, M., Šmuc, T.: Extending rapidminer with recommender systems algorithms. In: Proceedings of the RapidMiner Community Meeting and Conference, pp. 63–75 (2012)

20. Bošnjak, M., Antulov-Fantulin, N., Šmuc, T., Gamberger, D.: Constructing recommender systems workflow templates in RapidMiner. In: Proc. of the 2nd RapidMiner Community Meeting and Conference, pp. 101–112 (2011)
21. Chen, B.C., Kifer, D., LeFevre, K., Machanavajjhala, A.: Privacy-preserving data publishing. Foundations and Trends in Databases 2(1-2), 1–167 (2009)
22. Fung, B.C., Wang, K., Fu, A.W.C., Yu, P.S.: Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques, 1st edn. Chapman & Hall/CRC (2010)
23. Aggarwal, C.C., Yu, P.S. (eds.): Privacy-Preserving Data Mining. Models and Algorithms. Springer (2008)
24. Berendt, B.: More than modelling and hiding: towards a comprehensive view of web mining and privacy. Data Mining and Knowledge Discovery 24(3), 697–737 (2012)
25. Kenig, B., Tassa, T.: A practical approximation algorithm for optimal k-anonymity. Data Mining and Knowledge Discovery 25(1), 134–168 (2012)
26. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering 13(6), 1010–1027 (2001)
27. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: L-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data 1(1) (2007)
28. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
29. Wolf, P.P.D., Amsterdam, H.V., Design, C., Order, W.T.: An empirical evaluation of PRAM statistics. Netherlands Voorburg/Heerlen (2004)
30. Wolf, P.P.D., Gouweleeuw, J.M., Kooiman, P., Willenborg, L.: Reflections on PRAM. Statistical Data Protection, Luxembourg, pp. 337–349 (1999)
31. Aggarwal, C.C., Yu, P.S.: A framework for condensation-based anonymization of string data. Data Mining and Knowledge Discovery 16(3), 251–275 (2008)
32. Raghunathan, T., Reiter, J., Rubin, D.: Multiple imputation for statistical disclosure limitation. Journal of Official Statistics 19(1), 1–16 (2003)
33. Fienberg, S.: A radical proposal for the provision of micro-data samples and the preservation of confidentiality. Technical report, Department of Statistics, Carnegie-Mellon University (1994)
34. Dandekar, R.A., Cohen, M., Kirkendall, N.: Sensitive micro data protection using latin hypercube sampling technique. In: Domingo-Ferrer, J. (ed.) Inference Control in Statistical Databases. LNCS, vol. 2316, pp. 117–125. Springer, Heidelberg (2002)
35. Dandekar, R.A., Domingo-Ferrer, J., Sebé, F.: LHS-based hybrid microdata vs rank swapping and microaggregation for numeric microdata protection. In: Domingo-Ferrer, J. (ed.) Inference Control in Statistical Databases. LNCS, vol. 2316, pp. 153–162. Springer, Heidelberg (2002)
36. Reiter, J.: Inference for partially synthetic, public use microdata sets. Survey Methodology 29(2), 181–188 (2003)
37. Brookshear, J., Glenn, H.: Theory of Computation: Formal Languages, Automata, and Complexity. Benjamin/Cummings Publish Company, Redwood City (1989)