ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Non-Linear Recommender System using Stochastic Chaining and N-Gram Model

Venkata Sai Sriharsha Sammeta[1], Navya Yenuganti[2], K Jairam Naik[3]

Computer Science, Vasavi College of Engineering, Osmania University & Intern@Oracle India [1]

Computer Science, Vasavi College of Engineering, Osmania University, India [2]

Ph.D. in Computer Science, Professor, Vasavi College of Engineering, Osmania University, India[3]

ABSTRACT: Typical recommender systems accept a static view of the recommendation process and treat it as a declaration made in advance problem. We argue it is more suitable to view the problem of generating recommendations as a subsequent optimization problem and, consequently, that stochastic process support a more appropriate model for recommender systems. Stochastic process introduce two benefits: they take into account the long-term belongs of each recommendation and the normal value of each recommendation. To accomplish in practice, stochastic process-based recommender system must apply a strong initial model, must be soluble quickly, and should not deplete too much memory. In this paper, we explain our particular model, its primary using a predictive model, the result and advance algorithm, and its actual accomplishment on a commercial site. We also explain the particular predicting model we used which outperforms previous models. Our arrangement is one of a small number of commercially expanded recommender systems. As far as we know, it is the first to article experimental analysis performed on a real commercial site. These results approve the commercial value of recommender systems, and in exact, of our stochastic process approach.

KEYWORDS: Recommender Systems, N-Gram Model, Stochastic Chaining;

I. INTRODUCTION

In many markets, customers are faced with a wealth of products and data from which they can select. To relieve this problem, many web sites effort to help users by incorporating a recommender system that supports users with a list of items and/or webpages that are likely to interest them. Once the customer makes her preferred, a new list of recommended elements is presented. Thus, the recommendation action is a sequential process. Moreover, in many domains, user choices are subsequent in nature – for example, we buy a book by the author of an advance book we liked.

The subsequent nature of the recommendation process was noticed in the past. Taking this concept one step farther, we suggest that recommendation is not simply a subsequent prediction problem, but rather, a subsequent decision problem. This conclusion should take into account the subsequent process involved and the optimization test suitable for the recommender system, such as the profit created from selling an item. Thus, we plan the use a well-known stochastic model of sequential decisions.

In specific, an optimal policy will take into account the likelihood of a recommendation to be approved by the user, the actual value to the site of such an acceptance, and the long-term suggestions of this on the user's future choices. These applications are taken with the appropriate balance to assure the generation of the maximal expected reward stream.

The assets of an MDP-based recommender system discussed above are offset by the fact that the model limits are unknown. Standard support for learning techniques that learn optimal behaviour's will not do they take extensive time to converge and their primary behaviour's random. No commercial site will expand a system with such behaviour. Thus, we must find ways for creating good primary estimates for the MDP parameters. The way we suggest initializes a predictive model of user behaviour using data gathered on the site prior to the exercise of the recommender system. We then use the predicting model to provide initial parameters for stochastic chaining.

Our initialization process can be accomplished using any predictive model. In this paper we suggest a specific model that outperforms previous ways. The predictive model we explained is motivated by our subsequent view of the recommendation process, but comprises an independent grant. The sample can be thought of as an n-gram mode a Markov chain in which states correspond to subsequent of events. In this paper, we affirm the latter interpretation due to its natural relationship with stochastic chaining. We note that have explained the use of simple n-gram models for predicting web pages.

Certifying recommender system algorithms is not simple. Most suggested systems, such as reliance networks, are tested on historical data for their predictive exactness. That is, the system is coaching using historical data from sites that do not support recommendations, and tested to see whether the recommendations adapt to actual user behaviour. We present the solutions of a similar test with our system showing it to perform better than the previous leading path.

The main improvements of this paper are: (1) A novel way to recommender systems depend on an stochastic chaining model together with appropriate initialization and result techniques. (2) A novel predictive sample that outperforms past predictive models. (3) One of a small number of commercial applications depends on stochastic chaining model. (4) The first experimental inquiry of a commercially deployed recommender system.

Stochastic chaining model, the process of customer navigating within airport. The state of this stochastic chaining was the consumer’s position and rewards were obtained when the customer entered a store or bought an item. Recommendations were circulated on a palm-top, indicating routes and stores to visit. However, the stochastic chaining model was hand-coded and experiments were attended with students rather than real users.

A) Recommender Systems:

Early when the Internet became broadly used as a source of information, data explosion became a problem that needed addressing. Many web sites presenting a broad variety of content discovered that users had issues finding the items that interested them out of the entire selection. Recommender Systems help users’ constraint their search by supplying a list of elements that might interest a specific user. Different ways were suggested for supplying meaningful recommendations to customers and some were used in modern sites.

B) N-gram Models:

N-gram models derive in the field of language modelling. They are utilized to predict the next word in a sentence given the last n − 1 words. In the easiest form of the method, probabilities for the next word are supposed via maximum likelihood; and many methods continuing for improving this simple path including skipping, clustering, and smoothing. Skipping accepts that he probability of the later word xi depends on words other than just the previous n − 1. An estimate model is built using skipping and then linked with the normal n-gram model. Clustering is path that groups some states together for intentions of predicting next states. Grouping helps to identify the problem of data scarcity. Smoothing is a normal name for samples that modify the assumption of probabilities to achieve higher accuracy by adjusting zero or low anticipation upward. One type of smoothing is finite combining modelling, which combines multiple models via a convex mixture. In specific, given k component models for xi given a prior subsequent X—pM1(xi|X),..., pMk(xi|X)—we can explain the k-component combination model p(xi|X) = π1 · pM1(xi|X) + ···+ πk · pMk(xi|X), where ∑ki=1 πi = 1 are its combined weights. Details of these and other models.

C) MDPs:

An MDP is a model for subsequent stochastic decision problems. An stochastic chaining is by definition a four-tuple: hS,A,Rwd,tri, where S is a function of states, A is a set of actions, Rwd is a reward function that allows a real value to each state/action combination, and tr is the state-transition function, which supports the probability of a transition between every pair of states given each action.Various accurate and approximate algorithms exist for automating an optimal policy. Below we concisely review the algorithm known as policy-iteration, which we utilize in our implementation. A basic concept in all ways is that of the value function. The value function of a policy π, denoted Vπ, accredit to each stat s a value which conform to the expected infinite horizon discounted sum of accolade obtained when using π starting from s. This function contents the following recursive equation:

$$V^{\pi}(s) = Rwd(s,\pi(s)) + \gamma \sum_{s_j \in S} tr(s,\pi(s),s_j)V^{\pi}(s_j) \qquad (1)$$

where 0 < γ < 1 is the discount factor.2 An optimum value function, denoted V∗, accredit to each state s its value given approval to an optimal policy π∗and satisfies

$$V^*(s) = \max_{a\in A}[Rwd(s,a)) + \gamma \sum_{s_j\in S} tr(s,a,s_j)V^*(s_j)]. \quad (2)$$

To assert a π∗and V∗ using the policy-iteration algorithm, we explore the space of possible policies. We initiate with an primary policy π0(s) = argmaxa∈ARwd(s,a). At each step we automate the value function depend on the former policy and update the policy given the new value function:

$$V_i(s) = Rwd(s,\pi_i(s)) + \gamma \sum_{s_j\in S} tr(s,\pi_i(s),s_j)V_i(s_j), \quad (3)$$

$$\pi_{i+1}(s) = \operatorname*{argmax}_{a\in A}\,[Rwd(s,a) + \gamma \sum_{s_j\in S} tr(s,a,s_j)V_i(s_j)]. \quad (4)$$

II. The Predictive Model

Our initial step is to construct a predictive model of user purchases, that is, a method that can predict what element the user will buy next. This model does not take into account its consequence on the user, as it does not model the recommendation method and its effects. Suitable formulation of the state space, as our model. In Section 4 we shall show that our predictive model outperforms former models, and in Section 5 we shall start our stochastic chaining-based recommender system using this predictive model.

Some Improvements.

We experimented with several improvements to the maximum-likelihood n-gram model on data different from that used in our formal assessment. The improvements explained and used here are those that were found to work well.

One improvement is a form of skipping and is depend on the observation that the occurrence of the series x1, x2, x3 lends some possibility to the series x1, x3. That is, if a person bought x1, x2, x3, then it is possible that someone will buy x3 after x1. The specific skipping method that we found to work well is an easy additive model. First, the count for each state transition is introduced to the number of noticed transitions in the data. Then, given a user sequence x1, x2,...,xn, we add the fractional count ½ (j−(i+3)) to the transition from (xi, xi+1, xi+2) to (xi+1, xi+2, xj), for all i+3 < j ⩽ n. This fractional count compares to a diminishing probability of skipping a large number of transactions in the subsequent. We then normalize the counts to collect the transition probabilities:

$$tr_{MC}(s,s') = \frac{count(s,s')}{\sum_{s'} count(s,s')} \quad (6)$$

where count(s,s') is the count associated with the transition from s to s'.

A second improvement is a form of clustering that we have not found in the literature. Motivated by characteristics of our domain, the way exploits similarity of sequences. For example, the state hx,y,zi and the state hw,y,zi are identical because some of the items arrived in the former appear in the latter as well. The essence of our way is that the possibility of transition from s to s' can be predicted by developments from t to s', where s and t are identical. In particular, we describe the similarity of states si and sj to be

$$sim(s_i,s_j) = \sum_{m=1}^{k} \delta(s_i^m, s_j^m)\cdot(m+1) \quad (7)$$

where δ(·,·) is the Kronecker delta function and sim is the mth element in state si. This identical is arbitrary up to a constant. In addition, we define the identical count from state s to s'to be

$$simcount(s,s') = \sum_{s_i} sim(s,s_i)\cdot tr_{MC}^{old}(s_i,s') \quad (8)$$

where original transition function, with or without skipping. The advance transition probability from s' to s is then given by 5.

$$tr_{MC}(s,s') = \frac{1}{2}tr_{MC}^{old}(s,s') + \frac{1}{2}\frac{simcount(s,s')}{\sum_{s''} simcount(s,s'')} \quad (9)$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

A third improvement is the use of finite mixture modeling.6 identical methods are used in n-gram models, where—for example—a trigram, a bigram, and a unigram are combined into a single method. Our mixture model is encouraged by the fact that larger values of k lead to states that are more data whereas smaller values of k advantage to states on which we have more statistics. To balance these constraints properties, we mix k models, where the ith model looks at the final i transactions. Thus, for k = 3, we mix three methods that predict the next transaction depend on the last transaction, the final two transactions, and the last three transactions. In general, we can learn mixture weights from information. We can even allow the combined weights to depend on the given case Nonetheless, for easy, we use π1 = ··· = πk = 1/k in our experiments. Because our initial model is based on the k last items, the creation of the models for smaller values entails little automation overhead.

III. Evaluation of the Predictive Model

Initial incorporating our predictive model into stochastic chaining-based recommender system, we assessed the accuracy of the predictive model. Our assess used data corresponding to user behavior on a web site and engaged the evaluation metrics commonly utilized in the collaborative filtering literature. In Section 6 we assess the stochastic chaining-based way using an experimental path in which recommendations on an e-commerce site are manipulated by our algorithms.

A) Data Sets:

Two data sets were utilized: one supporting user transactions and one supporting user browsing paths obtained from web logs. We separated out items that were bought/visited less than 100 times and users who bought/browsed no more than one item as is commonly done when assessing predictive models. We were left with 116 items and 10820 customers in the transactions data set, and 65 items and 6678 customers in the browsing data set.7 In our browsing data, no cookies were utilized by the site. If the same customer visited the site with a new IP address, then we would treat her as a recent user. Also, activity on the same IP address was ascribing to a new user whenever there were no requests for two hours. These data sets were randomly divided into a training set (90% of the users) and a test set (10% of the users).

The rational for avoiding items that were rarely bought is that they cannot be reliably predicted. This is a conservative path which implies, in practice, that a rarely visited item will not be recommended by the system, at least exceptionally.

We assessed predictions as follows. For every user subsequent t1,t2,..,tn in the test set, we created the following test cases:

ht1i,ht1,t2i,...,htn−k,tn−k+1,...,tn−1i (10)

For each case, we then utilized our different models to determine the probability allocation for ti given ti−k,ti−k+1,...,ti−1 and ordered the items by this allocation. Finally, we utilized the ti actually noticed in conjunction with the list of recommended items to compute a score for the list.

B) Evaluation Metrics:

We used two scores: Recommendation Score (RC) and Exponential Decay Score (ED) with slight changes to fit into our sequential domain.

C) Recommendation Score:

For this measure of accuracy, an approval is deemed successful if the noticed item is among the top m recommended items. The score RC is the percentage of cases in which the declaration made in advance is successful. This score is useful for commerce sites that require a short list of approvals and therefore care little about the organizing of the items in the list.

D) Exponential Decay Score:

This measure of accuracy is depend on the position of the observed ti on the approvals list, thus processed not only the content of the list but also the order of items in it. The underlying expectation is that users are more likely to select a approval near the top of the list. In specific, it is assumed that a user will normally see the mth item in the list with probability

$$p(m) = 2^{-(m-1)/(\alpha-1)}, (m \geq 1) \qquad (11)$$

where α is the half-life parameter—the index of the element in the list with anticipation 0.5 of being seen. The score is given by

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

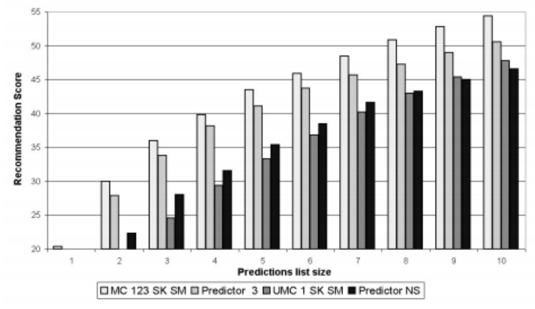
$$100 \cdot \frac{\sum_{c\in C} p(m = pos(t_i|c))}{|C|} \qquad (12)$$

where C is the set of all cases, c = ti−k,ti−k+1,...,ti−1 is a case, and pos(ti|c) is the position of the noticed item ti in the list of approved items for c. We used α = 5 in our experiments in order to be constant with the experiment. The relative performance of the samples was not sensitive to α.

IV. Comparison Models

A) UNORDERED MCS:

We also processed a non-sequential method of our predictive model, where subsequent such as hx, y,ziand hy,z, xi are mapped to the same state. If our expectation about the sequential nature of approvals is incorrect, then we should expect this model to accomplish better than our model,as it learns the probabilities using more coaching data for each state, combining all the ordered information into one unordered set. Skipping, clustering, and mixture modeling were contained as described in section 2. We call this model UMC**.**



(a) Transactions data set.

Variations of the MC Model

In order to measure how each n-gram enhancement affected predictive accuracy, we also processed models that excluded some of the enhancements. In noticing our results, we refer to a method that uses skipping and identical clustering with the terms SK and SM, respectively. In addition, we use numbers to denote which combination elements are used.

Figure 1(a) and figure 1(b) show the exponential decay score for the better samples of each type. It is valuable to note that all the MC models using skipping, clustering, and mixture modeling yielded better solutions than every one of the Predictor-k models and the non-sequential Predictor model. We see that the subsequent-sensitive models are better predictors than those that avoid sequence information.

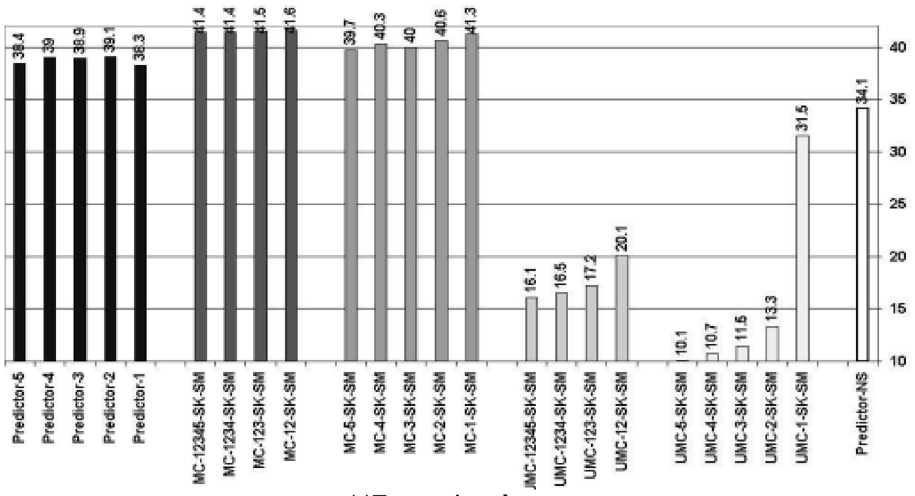
Figure 2(a) and Figure 2(b) show the approval score as a function of list length (m). Once again, subsequent models are superior to non-subsequent models, and the Markov chain models are major to the Predictor models.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

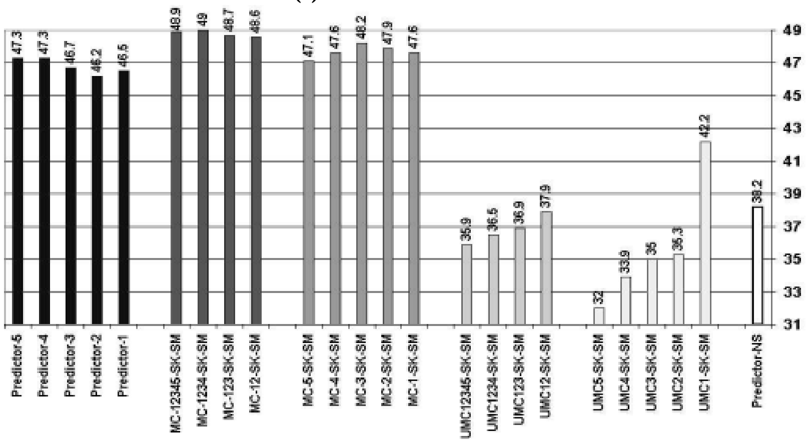
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015



(a)Transactions data set.



(b)Browsing data set.

Figure 1: Exponential decay score for different models.

Figure 1(a) and Figure 1(b) show how various versions of the Markov chain accomplished underthe exponential decay score in both data sets. We see that multi-element models out-perform single-component methods, and that similarity clustering is beneficial. In difference, we find that escaping is only beneficial for the transactions information set. Perhaps users tend to follow the same approaches in a rather moderate manner, or site structure does not support users to "jump ahead". In either case, once approvals are available in the site, skipping may confirm beneficial.

V. An MDP-Based Recommender Model

The predictive model we explained above does not attempt to capture the short and long-term effect of recommendations on the customer, nor does it try to optimize its behavior by taking into account such belongings. We now move to stochastic chaining model that explicitly models the approval process and attempts to optimize it. The predictive model plays major role in the construction of this model.

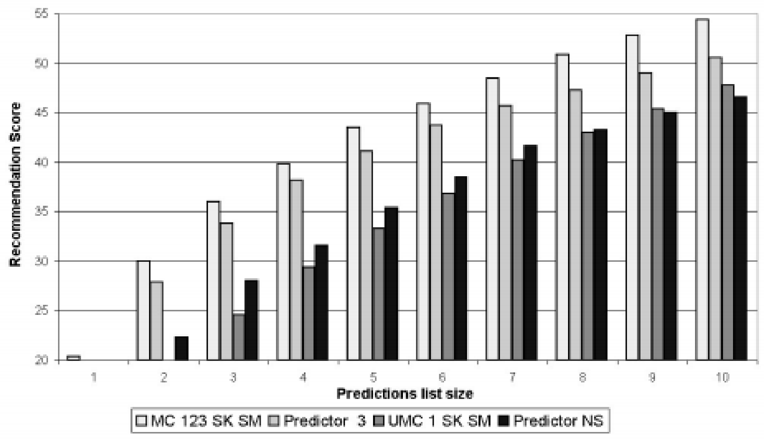
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

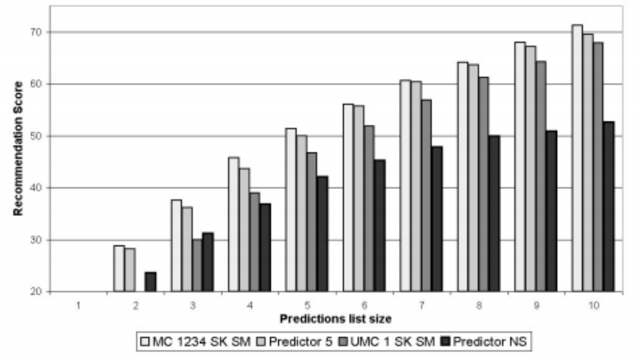
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

We expect that we are given a set of cases explaining user behavior within a site that does not support recommendations, as well as a probabilistic predictive sample of user acting without recommendations created from this data. The set of cases is needed to provide some of the approximations we make, and in specific, the lazy initialization way we take. The predictive model supports the probability the user will purchase a specific item x given that her sequence of past purchases is x1,...,xk. We denote this value by Prpred(x|x1,...,xk), where k = 3 in our case. It is important to stress that the path presented here is independent of the specific technique by which the above predictive value is estimated. Naturally, in our exercise we used the predictive model generated in Section 3, but there are other ways of constructing such a model.





(a)Transactions data set.

(b) Browsing data set.
Figure 2: Recommendation score for various models.

Defining the MDP:
Recall that to explain stochastic chaining; we need to support a set of states, actions, transition function, and aaccolade function. We now describe each of these elements. The states of the stochastic chaining for our recommender system are k-tuples of items, some prefix of which may provide null values corresponding to missing items. This supports us to model shorter sequences of purchases.
The actions of the stochastic chaining correspond to a recommendation of an item. One can contemplate multiple recommendations but, to keep our presentation easy, we start by discussing single approvals.
Benefits in our stochastic chaining encode the utility of selling an item as described by the site. Because the state encodes the list of elements purchased, the reward based on the last item describing the current state only. For example, the reward for state hx1, x2, x3i is the reward created by the site from the sale of item x3. In this paper, we utilize net profit for benefit.
The state following each recommendation is described by the user's response to that approval. When we recommend an element x', the customer has three options:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

• Obtain this recommendation, thus transferring from state hx1, x2, x3i into hx2, x3, x0i
• Elite some non-recommended item x", thus transferring the state (x1, x2, x3) into (x2, x3, x").
• Elite nothing in which case the system remains in the identical state.

Thus, the stochastic element in our method is the user's actual choice. The transition function for the stochastic chaining model is the probability that the customer will select element x" given that element x' is approved in state (x1, x2, x3i). We write tr1MDP to denote that only single element recommendations are used.

$$tr^1_{MDP}(\langle x_1, x_2, x_3\rangle, x', \langle x_2, x_3, x''\rangle) \qquad (13)$$

VI. GENERATING MULTIPLE RECOMMENDATIONS

When moving to multiple approvals, we make the estimation that recommendations are independent. Namely we estimate that for every pair of sets of recommended items, R,R', we have that

$$(r \in R \wedge r \in R') \vee (r \notin R \wedge r \notin R') \implies tr_{MDP}(s, R, s \cdot r) = tr_{MDP}(s, R', s \cdot r) \qquad (21)$$

This assumption might show to be false. It appears reasonable that, as the list of approvals grows, the probability of selecting any element reduces. Another more subtle example is the case where the system "thinks" that the customer is concerned in an inexpensive cooking book. It can then advice a few very costly cooking books and one is reasonably priced cooking book. The fairly priced book will seem like a bargain compared to the expensive ones, thus making the customer more likely to buy it.

Nevertheless, we make this estimation so as not to be forced to generate a larger action space where functions are ordered combinations of recommendations. Taking the simple way for representing the transition action we defined above, we still keep only two values for each state–itempair:

$$tr_{MDP}(s, r \in R, s \cdot r) = tr^1_{MDP}(s, r, s \cdot r), \qquad (22)$$

the probability that r will be bought if it came into view in the list of recommendations; and

$$tr_{MDP}(s, r \notin R, s \cdot r) = tr^1_{MDP}(s, r', s \cdot r) \text{ for all } r' \neq r, \qquad (23)$$

the probability that r will be bought if it did not came into view in the list.

As before, trMDP(s,r∈/ R,s·r) does not based on r, and will not base on R in the discussion that allows. We note again, that these values are merely reasonable primary values and are adapted by our system based on actual user behaviour.

The main of this work is that approve should be viewed as a subsequent optimization problem, and stochastic chaining provide an adequate model for this view. This is to be compared with previous systems which used predictive models for creating recommendations. In this section, we present an empirical confirmation of our thesis. We compare the accomplishment of our stochastic chaining-based recommender system with the accomplishment of a recommender system based on our predictive model as well as other variants.

Users accepted approvals when adding items to the shopping cart.12.The approvals were depend on the last k items added to the cart organized by the time they were added. Every time a user was presented with a list of approvals on either page, the system stored the approvals that were presented and recorded whether the user purchased an approved item. Cart deletions were rare and avoided. Once every two or three weeks, a method was run to update the model given the data that was composed over the latest time period.13

The stochastic chaining model was built utilizing data gathered while the model was running in the site with incremental updates for almost a year. We distinguished four policies; where the first policy uses data about the effect of recommendations, and the remaining policies are depend on the predictive model solely:

• Optimal – recommends elements based on optimal policy for the stochastic chaining.
• Greedy – recommends elements that maximize Pr(x|h)· R(x) (where Pr(x|h) is the probability of buying element x given user history h, and R(x) is the value of x to the site.
• Most likely – recommends elements that maximize Pr(x|h).
• Lift – recommends elements that maximize Pr(x|h)Pr(x), where Pr(x) is the prior probability of buying element x.

To process the different policies we ran a simulation of the cooperation of a user with the system.During the simulation the system created a list of recommended items R, from which the simulated customer selected the next item, utilizing the distribution tr(s,R,s· x)—the probability that the next chosen item is x given the current state s and the recommendation list R, pretending the purchaseof x by the user. The length of customer session was taken from the learned distribution of customer sessionlength in the normal site. We ran the pretend for 10,000 iterations for each policy, and calculatedthe average gathered reward for user session.

VII. CONCLUSION

This paper explains a new model for recommender systems based on a stochastic chaining. Our work presents one of a few examples of commercial systems that use stochastic chaining, and one of the first reports of the accomplishment of commercially expanded recommender system. Our experimental results approve both the utility of recommender systems and the utility of the stochastic chaining-based approach to recommender systems.
To support the kind of accomplishment required by an online commercial site, we utilized various approximations and, in specific, made heavy utilize of the special properties of our state space and its subsequent origin. Whereas the applicability of these techniques more than recommender systems is not clear, it shows an interesting case study of a successful real system. Moreover, the subsequent nature of our system stems from the fact that we need to maintain control of past purchases in order to acquire a state space. The need to record facts about the previous in the current state arises in different versions, and has been discussed in a number of papers on handling non-first-order
Another interesting technique is our use of off-line data to start a model that can provide adequate primary performance.
In the future, we to increase our transition function on those states that are infrequently encountered using generalization techniques, such as skipping and clustering that are identical to the ones we engaged in the predictive Markov chain model. Other potential improvements are utilized of a partially notable stochastic chaining to model the user. As a model, this is more estimate than a stochastic chaining, as it supports us to explicitly sample our uncertainty about the true state.
Weaknesses of our predictive model contain the use of ad hoc weighting functions for skipping and identical functions and the use of fixed mixture weights. Although the approvals that result from our current model are useful for ranking items, we have identified that the model probability distributions are not graduated. Learning the weighting functions and mixture weights from information should improve graduation. In addition, in informal experiments, we have seen proof that learning case-dependent mixture weights should increase predictive accuracy.

REFERENCES

1. C. Boutilier, F. Bacchus, and A. J. Grove. Rewarding behaviors. In Proceedings of the Thirteenth National convention on Artificial Intelligence, Vol. 2, pages 1160–1167, Portalnad, OR, 1996.
2. M. Balabanovic and Y. Shoham. Gathering content-based and collaborative approva. Communications of the ACM, 40(3):62—72, March 1997.
3. R. E. Bellman. Active Programming. Princeton University Press, 1962.
4. A. Jameson and T. Bohnenberger. When tactics are better than plans: decision-theoretic planning of recommendation subsequent. In IUI '01: Proceedings of the 6th international convention on intelligent user interfaces, pages 21–24, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-325-1.

BIOGRAPHY

Venkata Sai Sriharsha Sammeta is an undergrad Machine Learning researcher in Computer Science department of Vasavi College of Engineering, Osmania University. Previously, he did internship in Oracle R&D India Private Limited, developing machine learning models for Ordering and Service Management (OSM) systems to predict future orders and improved the performance of Order Lifecycle Management (OLM) significantly. His current research areas are in the fields of Artificial Intelligence, Machine Learning, Deep Learning, Natural Language Processing and Recommender Systems.

Navya Yenuganti is a computer Science student at Vasavi College of Engineering, Osmania University. Previously, she did an internship in Eidiko Systems Inc, an IBM business partner that develops mobile and cloud solutions for

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

enterprise. There, she worked on auto-reply functionality for messages using heuristics-based approaches. Her research areas are in the fields of Convolutional Neural Networks, Machine Translation and Natural Language Processing.

K Jairam Naik received his Ph.D in Computer Science from JNTU University, Hyderabad. He is well known for his recent work in grid computing defining novel ways to improve the performance of grid by using Ant-colony optimization for balanced job scheduling algorithm. He is currently Professor in Vasavi College of Engineering, Osmania University, Hyderabad teaching Machine Learning and advanced Statistical Learning. His current research includes Machine Translation, Natural Language Processing and Convolutional Neural Networks. He has presented and published over 30 research papers in National and international Conferences and Journals. He is also an editor and part of the selection-committee for journals like IJSETI and IJIEECR.