Web Pages New Emerging Trends Detection Using Distributed Learning Automata

Mahdi Bazargani, Ali Syed, BelindaFridey
Charles Sturt University
Faculty of Business, Melbourne, Australia

ABSTRACT: New emerging trend is an especial theme. It shows the upcoming trends at intervals. By knowing the new emerging trends,the upcoming tendencies will be discovered. In our paper, we discover the new trends in an online store. We use Distributed Learning Automata to discover the relation among products. Furthermore, we conclude the similarity network and obtain the new trends according to our declared measurements.
KEYWORDS: New Emerging Trends Detection; Learning Automata; Web Mining; Similarity Network.

Introduction

An emerging trend is a topic area that is growing in interest and utility over time. Indeed, by realizing emerging trends, new tendencies could be known. Human growth and life continuance certainly is impossible without understanding new trends, especially if these trends be minatory.

Mainly, in our paper, we focus on finding the new trends in web pages according to the users' interests. The methodology proposed in this paper is based on the utilization of distributed learning automata to trace user interests in web pages and then ranking them according to the users' interests.

We build a network representing the connection of pages and interests among them. Analyzing this network is the best way for determining the new trends in that. This network can be completely large and complicated. While, by considering some simple attribute, we could find the covert and overt relations among interest s of the users [MH06].

In the following sections, we discuss on Learning Automata concepts. Then we introduce some criteria for measuring and obtaining the new tendencies. Finally, we implement our approach on an online store and show our results.

1.Learning Automata

Learning Automata is an abstract model for interaction and is based on some finite actions which specify interaction with the environment and selection of the best according to feedback.

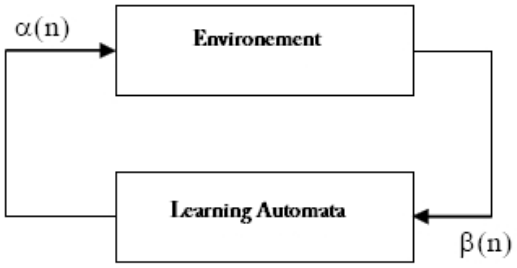


Figure 1. **Interaction between Environment and Learning Automata**

Figure 1 depicts the relation between environment and learning automata. Environment can be represented by a triple statement $E = \{\alpha, \beta, c\}$ in which α represents the input actions $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_r\}$, β represents the output actions $\beta = \{\beta_1, \beta_2, ...\beta_m\}$ and C represents the penalty factor $c = \{c_1, c_2, ..., c_r\}$.

β Can have two values: $p = 0$ for penalty factor and $p = 1$ for positive reaction. In static structure the penalty factors would remain fixed while in dynamic structure it would be changed according the learning algorithm and interaction with the environment.

2.Learning Automata with Variable Structure

These automata can be represented with a quadric statement $E = \{\alpha, \beta, p, T\}$ in which α represents the input actions $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_r\}$, β represents the

output actions $\beta = \{\beta_1, \beta_2, ...\beta_m\}$ and P represents the set of probabilities of each action $p = \{p_1, p_2, ..., p_r\}$ and also $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ is the learning algorithm. In this kind of automata if the action α_i would be selected in the step n and it resulted in a positive reaction the probability of action α_i would be increased and other actions' would be decreased. If the action receives a suitable response from the environment the probabilities would be changed as follow.

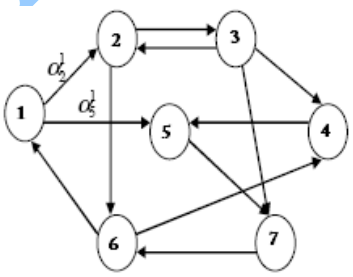
$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$p_j(n+1) = (1-a)p_j(n) \quad \forall j \qquad j \neq i$$

in which a is the encouragement factor, the sum of all new probability would be 1 after the changes.

3.Distributed Learning Automata

A DLA is a network of learning automata in which they cooperate with each other. Each time just one automaton is active, and the number of action one automata can perform is the same as the number of automata's connected to it. Consequently, every action triggers the peer automata.

A DLA can be represented with a graph, an edge (LA_i, LA_j) shows the action α_j^i in LA_i triggers the LA_j .the probability of actions would be represented $p^k = \{p_1^k, p_2^k, ...p_{r_k}^k\}$ in which p_m^k is the probability of action α_m^k which triggers LA_m. [TB87, BM11, MB01].



$DLA=(V,E)$
$V=\{LA_1, LA_2, ..., LA_n\}$
$E \subset V \times V$
$(LA_i, LA_j) \in E$

Figure 2. **A DLA with 7 Learning Automata's**

4.Determining the Structure of Web pages using DLAs

When a user request web pages consecutively, the probability is that those pages are correlated and address a similar subject. Such navigation can be recorded using Learning Automata. Each page is represented to the Learning Automata which computes the relativity to other pages. Over the Internet, the web pages and the users navigation acts as the environment and a Learning Automata including the DLA exists for each page. Every action in the DLA represents the navigation to other pages, and p_i illustrates relevancy between the destination and current page. Each transition that shows a positive reaction causes the probability to be increased. Saati and Meybodi in [SM05] introduced a new self organizing model to compute the updating of the probabilities. Suppose $p^k = \{p_1^k, p_2^k, \ldots p_{r_k}^k\}$ is the probability vector of LA_k which is devoted to page k and p_m^k is the probability of action α_m^k and r is the number of pages. If the user traverses from $D_k \to D_m$ (moving from page k to page m) the LA_k updates the probability vectors as follow

$$p_m^k(n+1) = p_m^k(n) + a_m^k\left[1 - p_m^k(n)\right]$$

$$a_m^k = \frac{E_m^k}{1 + E_m^k}$$

$$E_m^k = -\left(p_m^k \log p_m^k + \left(1 - p_m^k\right)\log\left(1 - p_m^k\right)\right)$$

The above value for E_m^k denotes the relation between page k and m. As the value of E_m^k increases, the relevancy of the page also increases. This proves that the encouraging factor changes as probability among the pages is updated. The algorithm of web relation is as follow [SM05].

1-Create a DLA for the current web pages
2- Initialize the probability vectors
3- For every user do the bellow steps
4- For every move from $D_k \to D_m$ update the probability vectors as stated above.

Traran et al, in [T+10], introduced an optimization on previous works done by Saati. They give rewards according to a transitive relation among pages traversed in a path traversed. In this case the first page receives reward for all the pages in the path. The closest pages to the first node result in more rewards. On the other hand, they consider a penalty factor, for cycle traverses. Cycle traversing implies and uncertainty among pages traversed by the user.

To simplify our implementation and enhance the efficiency for consecutive repetition of algorithm execution, we don't apply optimizations offered by Traran. As seen, we do not consider penalty for negative feed-back actions. This why, we use efficiency criteria compelling the sum of action not to be constant. Therefore, we did not consider the penalty factor for Learning Automata. On the other hand, positive reactions cause the increase of overall score of that Learning Automata.

For every $V = \{LA_1, LA_2, ..., LA_n\}$, we assign its score according to the frequency of the visits. For every LA_i if it is visited n_i we assign the score as follow.

$$S(LA_i) = \frac{n_i}{\sum_{i=1}^{n} n_i}$$

5.Network analysis

In this section, we introduce our criteria, for analyzing the graph which we gained in previous sections. For every two node we make the similarity graph as follow.

$$D(i,j) = \begin{cases} p_j^i & p_j^i > \frac{1}{n} \\ 0 & p_j^i \le \frac{1}{n} \end{cases}$$

where n is the number of nodes and p_j^i is the probability vectors of (LA_i, LA_j). Moreover, as defined in previous section, the sole score of $node_i$ is gained as follow.

$$S(i) = \frac{n_i}{\sum_{i=1}^{n} n_i}$$

in which n_i is the frequency of $node_i$ in dataset. More precisely, the number of hit frequency of the page caused by the users.

We define Average Degree Centrality for $node_i$ is as follow [WF94].

$$ADC(i) = \frac{\sum_{m=1}^{n} D(i,m)}{n \times (n-1)}$$

This measurement represents if the network is connected and correlated to each other strongly. In the other words, this measurement shows the average tendency of the users to the total web pages.

Another parameter is the Degree of centrality which defines the centrality of every node. The formula is as follow.

$$DC(i) = S(i) \times \sum_{m=1}^{n} D(i,m)$$

Another measurement for defining the centrality and importance of a node is Visibility defined as follow [Fre79].

$$V(k) = \sum_{i \neq j \neq k \in V} \frac{\sigma_{ij}(k)}{\sigma_{ij}}$$

$\sigma_{ij}(k)$ is the cost of the path from $node_i$ to $node_j$ which covers $node_k$ .On the other hand, σ_{ij} is the cost of all the paths from $node_i$ to $node_j$ .While ADC shows the total interest to the network, Centrality and Visibility show the strength of tendency for every node.

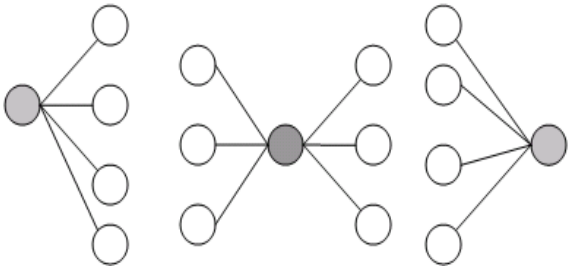


Figure 3. **Emerging trends shown with darker lights**

6.Results

For our implementation, we used BMS-WebView-1 dataset [K+00]. It contains several months' worth of click stream data from two ecommerce web sites. Each transaction in the two datasets is a web session consisting of all the product detail pages viewed in that session. In general, each customer can have multiple sessions. Each session can have multiple page views and multiple orders. Each order can have multiple order lines. Each order line is a purchase record of one product with a quantity of one or more.

Firstly, we consider 2873 transaction of dataset which had complete and integrate data. There were 247 different products in the transactions. This is obvious the number of products could be considered as the number of pages. We divided our transaction in four same size length and compute the *ADC* . The results have shown in the following table.

TABLE I. **ADC Parameter**

Sets	Number of Products	ADC
1	157	0.452
2	193	0.699
3	207	0.722
4	247	0.791

By traversing more transaction the *ADC* increases and shows the total tendency increased in the network.

We again repeat our experiments to see products which have more Degree of centrality. We have shown the 10 best products.

TABLE II. **Degree of centrality Parameter**

Ranks	Product ID	DC
1	10877	0.0653
2	10841	0.0525
3	12883	0.0508
4	10881	0.0501
5	12327	0.0455
6	12483	0.0342
7	35181	0.0331
8	12711	0.0290
9	12675	0.0267
10	12579	0.0265

We again repeat our experiments to see products which have more Visibility. We have shown the 10 best products.

TABLE III. **Visibility Parameter**

Ranks	Product ID	Visibility
1	10877	0.0283
2	12883	0.0257
3	10841	0.0250
4	35181	0.0239
5	12327	0.0228
6	12483	0.0219
7	12867	0.0213
8	12715	0.0174
9	12335	0.0116
10	12907	0.0105

Finally, we select the products resulted the best rank in our two measurements.

Conclusions

In this paper, we introduced a new method for discovering the new emerging trends. We made use of learning Automata to determine the relation among pages by the help of the intelligence of the people. Consequently, we made

the similarity network and introduced different measurements to determine the most upcoming trends and tendencies.

References

[BM11] **H. Beigy, M. R. Meybodi** - *Utilizing Distributed Learning Automata to Solve Stochastic Shortest Path Problem*, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, World Scientific Publishing Company, to appear.

[Fre79] **C. Freeman Linton** - *Centrality in social networks: I. Conceptual clarification*" Social Networks 1, Page 215 – 239, 1979.

[K+00] **R. Kohavi, C.E. Bradley, B. Frasca, L. Mason, Z. Zheng** - *KDD-Cup 2000 Organizers' Report: Peeling the Onion*. SIGKDD Exploration 2(2):86–93. 2000.

[MB01] **M. R. Meybodi, H. Beigy** - *Solving Stochastic Path Problem Using Distributed Learning Automata*, Proceedings of The Sixth Annual International CSI Computer Conference, CSICC2001, Isfahan, Iran, pp.70-86, Feb. 20- 22 , 2001.

[MH06] **N. Memon, L. L. Henrik** - *Practical Approaches for Analysis, Visualization and Destabilizing Terrorist Networks*. In the Proceedings of ARES 2006, pp. 906-913, 2006.

[SM05] **S. Saati, M. R. Meybodi** - *A Self Organizing Model for Document Structure Using Distributed Learning Automata,* Proceedings of the Second International Conference on Information and Knowledge Technology (IKT2005), Tehran, Iran, May 24-26.2005.

[TB87] **M. A. L. Thathachar, R. Harita Bhaskar** - *Learning Automata with changing number of actions*, IEEE Transaction on System, man and cybernetice, vol.SMC- 17, No.6, Nov.1987.

[T+10] **M. Traran, Sh. Motamedimehr, A. Hashemi, M. R. Meybodi** - *Identification of Web Communities using Distributed Learning Automata and Graph Partitioning*, Proceedings of the 4th Iran Data Mining Conference (IDMC'10), Sharif University, Tehran Iran, November 30-December 1, 2010

[WF94] **S. Wasserman, K. Faust** - *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.