

AJAX-Based Data Collection Method for Recommender Systems

ZSOLT NAGY

Institute of Mathematics and Computer Science

College of Nyiregyhaza

Nyiregyhaza, Sostoi u. 31/B

HUNGARY

info@nagyzsolt.hu

Abstract: - Recommender systems are powerful tools not only for e-commerce but for several areas of our Internet life. As the web has been developed, newer and newer technologies have become available to support data collection algorithms. In this paper, we present a new AJAX-based implicit data collection method which can invisibly change the user interface and collect user behavior data in real time. These two features can improve the user experience and the efficiency of the existing data collection methods.

Keywords: - implicit data collection, user profiling, AJAX, recommender system

1 Introduction

The World Wide Web contains more and more data, and it is getting more difficult for everyone to find relevant information. Web users cannot live without well-optimized search engines, as we constantly find ourselves in the presence of the intelligent web and its main engine, the recommender system. Recommender systems are used not only by e-Commerce systems [3], [8] but appear in many areas of Web activities, just to mention the most popular ones: e-Learning [6], entertainment [7] or social media [4], [5]. A recommender system is a collection of specific algorithms and technologies which offer relevant items and services for users or user groups that meet the user's preferences [11].

2 Recommender Systems

The first recommender systems appeared during the early 90's, mostly based on and expanding the terminology of collaborative filtering [2]. Later, when both the numbers of researches and the potential usage areas had grown, the scope of the terminology broadened. Recommender systems now use the following five technologies [14]:

- Collaborative Filtering Recommender Systems [12]: as there is a given user, find another user or user group who has similar behavior to predict items of interest;
- Content-based Recommender Systems [13]: technology usually employs a classifier to predict items' similarity; Recommend an item or a service to a user based upon the

properties of the item and a profile of the user's interests

- Demographic Recommender Systems [15]: compute users' similarity using demographic information (age, location, profession, education, etc.);
- Knowledge-based Recommender Systems [16], [17]: build a knowledge base with a model of the users and/or items in order to apply inference techniques and find matches between users' need and items' features.
- Utility based Recommender Systems: [14]: Utility-based recommenders make suggestions based on a computation of the utility of each object for the user.

Several researches mention another category of systems; we commonly call them Hybrid Recommender Systems [14], which attempts to combine the advantages of two or more of the techniques described above. However, naming conventions have a wide range depending on the research area. Melville et al. call their method Content-Boosted Collaborative Filtering [7], Sobecki [20] has researches on Consensus-based Recommendation Method, while Burke [14] simply uses the Hybrid Recommender System phrase.

There are several researches and surveys about which technique is better, however, regardless of what kind of systems we are talking about, all the recommendations use four basic steps:

1. collect data about users with explicit and implicit techniques
2. make user profiles

3. compute recommendations based on proper techniques, algorithms
4. present recommendations to users [1].

2.1 Explicit or Implicit Data Collection

Deriving user profiles from collected data, recommender systems apply two main different methods: explicit user inputs, like user ratings and implicit user feedbacks [16] such as which links they click on, how long they examine a document or their browsing history. Most existing studies are based on datasets with explicit ratings of items, however, in commercial practice it is much more common to have datasets based on implicit actions [2].

There are several ways to collect implicit feedbacks, thus they can be divided into several categories. Implicit data can be a purchase or a browsing history, users' geographical location, the name of his / her Internet Service Provider, the type of the users' browser, screen resolution and so on.

As it can be collected silently without disturbing users (and potentially in great quantities), the potential impact of implicit feedback is ultimately greater than the explicit method [18].

As Wei et al. write in their study, user preferences can be captured using four types of customer data: demographic data, rating data, browsing pattern data and transaction data. Demographic attributes are age, gender, profession, and education, rating data is a collection of user's feedbacks about rated items, the browsing pattern can be a user's viewing history and transaction data provides details about purchased items and can be used to compute future customer purchasing preferences [1].

2.2 User Behavior Studies

However, acquiring user's behavior is as important for user profiling as his/her demographic or transaction data. Basically, behavior data can be a part of the browsing pattern, but while browsing data mostly contains information about visited web pages, user behavior can give useful information about user activity. Oard & Kim [18] have made a framework for observable behavior which was extended by Kelly and Teevan [19]. Examining user behaviors can supply important information for user profiling, such as the following: the user can examine a document, select a text, print an article or refer it to other people [18].

There are several researches about collecting user behaviors. Claypool et al. [9] collect implicit

and explicit user feedback through a custom web browser. Hijikata [10] employs JavaScript, Applets and an embedded proxy server to record user interactions on a web page, while Goecks and Shavlik [8] use a browser plug-in to keep track of user behaviors. Although these methods are able to collect user behavior data, they require special client-side software components such as a plug-in, a custom web browser or a Java Applet.

Hofmann et al. [6] describe a method called Usertrack which uses JavaScript that attaches the recorded data to hidden form fields to submit the data to the server as part of a HTTP request. Usertrack does not require any special infrastructure, but recording user actions are only possible when a HTTP request has been initiated with a link or a submit button click.

3 New Implicit Method

In this article we present a new method for collecting user behavior. The technology combines the advantages of the previously explained explicit and implicit methods; we collect user inputs without asking or disturbing him / her with surveys or item ratings. During a shopping or booking transaction we continuously acquire the relevant user interactions in real time. The standard implicit data collection methods, such as server or browsing history logging, are also working in the background.

3.1 e-Tourism Case Study

Understanding the new AJAX method and comparing it to previous researches, we take an e-Tourism case as an example. As tourism industry constantly uses recommender systems, we mostly focus on tourism in our researches and use an online booking system as our test framework

An online hotel booking system is made up from the following single steps:

1. the user provides the arrival and departure date, the number and ages of the guests,
2. then the booking system searches and calculates for relevant offers: room types and prices.

Additionally, modern booking systems offer various hotel services and upgrades. Fruit in the room, car or bicycle rent, zoo, theatre or museum tickets, massage, bath or wellness services, even a room upgrade from a two-bed room to a presidential suite.

Of course, online booking systems apply the tools of artificial intelligence; they do not offer cots

and highchairs, but they do offer spa therapies for retired couples.

The simplest way to decide whether a user prefers the selected service is to make sure if that person chooses and pays for it. It is an explicit and exact feedback; we should recommend it later for this or that kind of user. But in the case we have offered, for example, chocolate massage therapies for the user and she did not choose it in the end, can we say that the fact that she did not choose the chocolate massage means that she does not like this service? What if she likes the massage but it is too expensive for her, and that is the reason why she did not choose that service? With the present technologies we cannot answer the question as we do not have enough information for it.

3.2 AJAX

These are the typical situations where AJAX technology can help for a recommender system. With the AJAX technology we can record and store all user interactions dynamically in real time, all the data collection process being invisible for users [21]. There is no page reloading, flickering, blank screen for seconds while the browser is waiting for server response; with this method, the user experience is the same as nothing had happened.

With this feature we can collect a wealth of information about what kind of products or services the user has flirted with, but finally did not order. We record if s/he selects a checkbox next to a zoo ticket or scrolls down to read its description, and we even record whether she unchecked it and whether she ordered it later. Even if she did not order it, a few days later our intelligent booking system can offer the service again at a cheaper price or in a package directly packed for the specific user.

Previous researches also dealt with this kind of data collection [6], [8], [9], [10] but they did not have appropriate technology to achieve the most important goal: collect great amounts of user behavior data without additional infrastructure and user disturbance.

The word AJAX, an acronym for *Asynchronous JavaScript And XML*, was first mentioned by Garrett in 2005 [23], however, the technology when a server-side application can exchange data without reloading the page had already existed as *remote scripting* in 2002 [24]. It became extremely popular when the engineers of Google started to use it in Google Suggest, Gmail and Google Maps.

Ajax is a collection of several existing technologies, but it is enough to use only three components of the originally described [21]

package. We need XMLHttpRequest [27] object to exchange data asynchronously with a server, JavaScript to display and DOM [28] to interact with the information.

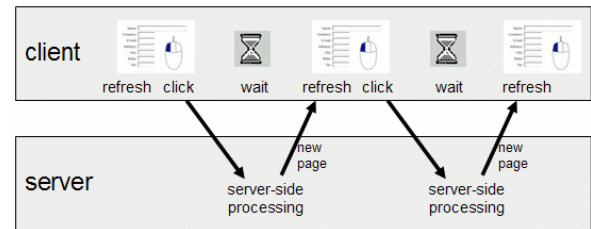


Fig. 1 Typical HTML user experience [22]

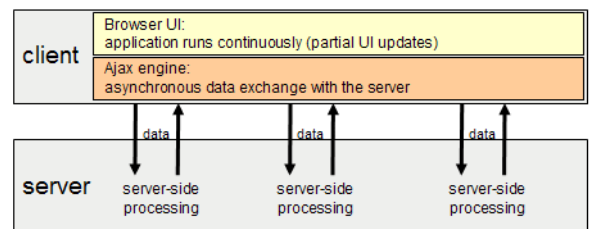


Fig. 2 AJAX powered user experience [22]

With Figure 1 we examine the typical web-based client-server communication stream: 1. a user clicks on a link or a button 2. the browser sends information to the web server 3. the user must wait for server response, and cannot do anything while the browser does not load the whole page. Figure 2 shows that client applications like web pages run continuously while the browser communicate with the server, so the user can do anything, and does not have to wait for any server response.

Although the client-server architecture is the same in both versions, there is one important difference. On the client side there is an Ajax engine (Fig. 2) which controls the necessary data exchange between the web server and the browser. The keyword is 'necessary' as we only send a small fraction of the whole web page whenever we need it. Because of the size of the exchanged data and the nature of the communication technology, in most cases all the operations are invisible for the user.

In implicit data collection, we use Ajax to send information from client to server without interrupting user activity, but Ajax has another benefit for recommender systems. It is able to change the web page dynamically in real time, based on the user's behavior or on a recommender algorithm.

3.3 Ajax-Based Data Collection

The essence of our new method comes from the nature of Ajax; we can recognize and collect user behavior information to our server-side profiling system on the fly, in real time. The efficiency of this method is much higher than the existing algorithms. Ajax-Based Data Collection (ABDC) owns all the benefits of previously discussed implicit methods, but gives the additional feature of an invisible data harvest process.

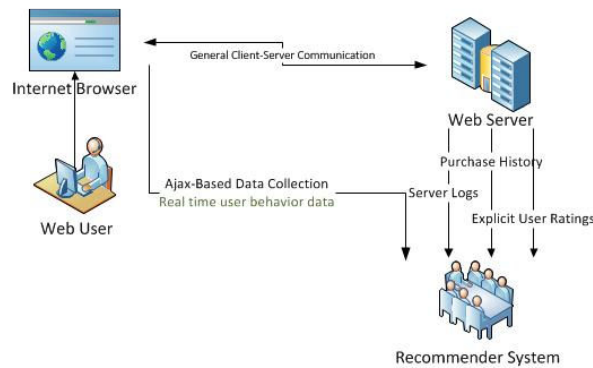


Fig. 3 ABDC Communication Schema

Figure 3 shows the simplified schema of our model. On the level of the client-server communication, we have added a parallel channel which ensures the asynchronous communication. Establishment of this channel requires some JavaScript codes as shown in Figure 3 and Figure 4.

```
22 request = new XMLHttpRequest();
```

Fig. 3: XMLHttpRequest object initialization

```
http_request.open("POST", url, true);
http_request.setRequestHeader("Content-Type", "application/x-www-form-urlencoded; charset=utf-8");
http_request.send(content);
```

Fig. 4 JavaScript methods for Ajax communication

In Figure 3 we initiate an Ajax communication with an XMLHttpRequest object, and then in Figure 4 we list those main JavaScript methods that are necessary for a successful asynchronous connection.

In our research, we apply this technology to an online booking system (Figure 4).

Fig. 4: Live Booking System with ABDC

In this case we focus on the user's room and hotel service selection behavior; we store all her selections during the booking process. At the room type selection step, we store whether she chooses a given room, or whether she has changed her mind and chooses a different one. At the tourism services selection stage we do the same. Our Ajax-Based Data Collection method implicitly gathers all the selection changes and sends them immediately to the Recommender System to calculate user profiles and relevant page content.

4. Privacy

Talking about recommender systems and data collection privacy is always a very important question. Even if we talk about explicit data collection where a user knows that we collect and store data about him/her, we have to keep the appropriate directives [26]. It is even more important if we use the implicit data collection method, when a user does not know and does not recognize the information acquiring process. National and international regulations can be standards for storing and managing data, but it is difficult to resolve the permanent conflict: everyone knows that recommender systems are very useful; despite this, more than 82% of web users refuse to give personal information on web pages [25]. This distrust appears exponentially in the case of implicit data collection methods.

5 Conclusion

Efficient data collecting is essential for successful user profiling, especially as implicit collecting technologies can be used successfully for acquiring large amount of data. Earlier studies that have been

presented in this article did not use or did not know about Ajax technology; however, using Ajax in further works can open a new area of implicit method research. Applying Ajax-Based Data Collection, implicit algorithms can collect data at every moment, without expecting the user to click on a link or hit a submit button.

Because of the nature of this technology, it can be used in every e-Commerce system, but it can be important not only from a marketing point of view, but also from a sociological and educational one. Imagine if we use Ajax-Based Data Collection in e-Learning systems, to discover and follow students' own path to the solution, record every movement and activity during learning or testing processes. Also, we can filter out possible cheat attempts, if we identify the correlation between previously given but deleted and finally given answers.

Ajax technology has already revolutionized the web user experience and application development, this is the perfect time to use it in other fields of science, especially to harness all of its benefits for recommender systems.

References

- [1] Wei K, Huang J, Fu S., A survey of e-commerce recommender systems. *IEEE international conference on service systems and service management*, China, 2007, pp. 1–5.
- [2] J. A. Konstan, J. Riedl, Recommender systems: from algorithms to user experience *User Model. User-Adapt. Interact.*, Vol. 22, Nr. 1-2, 2012, pp. 101-123.
- [3] CL. Huang, WL. Huang, Handling sequential pattern decay: Developing a two-stage collaborative recommender system, *Electronic Commerce Research and Applications* Vol. 8, Nr. 3, 2009, pp. 117–129,
- [4] J. Chen, W. Geyer, C. Dugan, M. Muller, I. Guy., “Make New Friends, but Keep the Old” – Recommending People on Social Networking Sites, *CHI 2009*, April 3–9, 2009, Boston, MA, USA.
- [5] J. Chen, R. Nairn, E. H. Chi, Speak Little and Well: Recommending Conversations in Online Social Streams *CHI 2011*, May 7–12, 2011, Vancouver, BC, Canada.
- [6] K. Hofmann, C. Reed, H. Holz, Unobtrusive Data Collection for Web-Based Social Navigation, *Workshop on the Social Navigation and Community-Based Adaptation Technologies In Conjunction with Adaptive Hypermedia and Adaptive Web-Based Systems (AH'06)* June 20th, 2006, Dublin, Ireland
- [7] P. Melville, R. J. Mooney, R. Nagarajan, Content-Boosted Collaborative Filtering for Improved Recommendations *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, pp. 187-192, Edmonton, Canada, July 2002
- [8] J. Goecks, J. Shavlik, Learning Users' Interests by Unobtrusively Observing Their Normal Behavior *IUI '00*, New Orleans, LA, Jan 2000, 129-132.
- [9] Claypool, M., Le, P., Wased, M., & Brown, D., Implicit interest indicators. *Proc 6th Intl Conf Intelligent User Interfaces (IUI '01)*, Santa Fe, NM, Jan 2001, 33-40.
- [10] Hijikata, Y., Implicit user profiling for on demand relevance feedback. *IUI '04 Funchal*, Madeira, Portugal, Jan 2004, pp. 198-205.
- [11] P. Melville, V. Sindhvani, Recommender Systems, *Encyclopedia of Machine Learning* Springer-Verlag, Berlin Heidelberg, 2010
- [12] X. Su and T. M. Khoshgoftaar, A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, vol. 2009, Nr. 4, 2009.
- [13] M. J. Pazzani and D. Billsus., Content-Based Recommendation Systems. *The Adaptive Web, Lecture Notes In Computer Science*, Vol. 4321, 2007, pp 325–341
- [14] R. Burke., Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, Vol. 12 Nr. 4, 2002, pp.331–370,
- [15] Pazzani, M. J., A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, Vol. 13, Nr. (5-6), 1999, pp. 393-408
- [16] D. Dell’Aglia, I. Celino, D. Cerizza., Anatomy of a Semantic Web-enabled Knowledge-based Recommender System, <http://www.larkc.eu>, 2010
- [17] R. Burke, Knowledge-Based Recommender Systems. *Encyclopedia of Library and Information Science*, 69 (32), 2000.
- [18] D. W. Oard, J. Kim: Modeling Information Content Using Observable Behavior, In *Proceedings of the 64th Annual Meeting of the American Society for Information Science and Technology*, USA, 38-45., 2001
- [19] Kelly, D. and Teevan, J. Implicit feedback for inferring user preference: A bibliography. *ACM SIGIR Forum*, Vol. 37, Nr. 2, 2003, pp. 18-28.
- [20] J. Sobacki, Implementations of Web-based Recommender Systems Using Hybrid Methods, *International Journal of Computer Science & Applications* Vol.3 Issue3 , 2006, pp 52 – 64,
- [21] Zs. Nagy, Ajax 2 in 1: Interactive education and modern web technology, *Journal of Applied Multimedia*, vol.: 1, num.: 1, 2010, pp. 39-44

- [22] OpenAjax Alliance: Introducing Ajax and Open Ajax. <http://www.openajax.org/whitepapers/>
- [23] Garrett, J, J (2005) : *Ajax: A New Approach to Web Applications*: <http://www.adaptivepath.com>
- [24] E. Costello: Remote Scripting with IFRAME, <http://www.oreillynet.com/pub/a/javascript/2002/02/08/iframe.html>
- [25] Kobsa, A., Privacy-Enhanced Personalization. *In Communications of the ACM*, Vol. 50, Nr. 8, 2007, pp. 24-33
- [26] Y. Wang, A. Kobsa, Technical Solutions for Privacy-Enhanced Personalization, *Intelligent User Interfaces: Adaptation and Personalization Systems and Technologies*. Hershey, IGI Global, 2008
- [27] W3, <http://www.w3.org/TR/XMLHttpRequest>
- [28] W3C: Document Object Model (DOM), <http://www.w3.org/DOM/>