Map-based recommendation of hyperlinked document collections

Mieczysław A.Kłopotek, Sławomir T.Wierzchoń,
Krzysztof Ciesielski, Michał Dramiński, Dariusz Czerski

Institute of Computer Science, Polish Academy of Sciences,
ul. Ordona 21, 01-237 Warszawa,Poland
kciesiel,klopotek,mdramins,stw,dcz@ipipan.waw.pl

Abstract. The increasing number of documents returned by search engines for typical requests makes it necessary to look for new methods of representation of the search results.
In this paper, we discuss the possibility to exploit incremental, navigational maps based both on page content, hyperlinks connecting similar pages and ranking algorithms (such as HITS, SALSA, PHITS and PageRank) in order to build visual recommender system. Such system would have an immediate impact on business information management (e.g. CRM and marketing, consulting, education and training) and is a major step on the way to information personalization.

1 Introduction

Recommender systems became indispensable part of modern e-business, especially using the electronic medium of the Internet. It is claimed that even 20% of clients may be encouraged to purchase a good by the most successful recommender systems. Recommenders are applied in advertisement of books (Amazon), CD's (MediaUnbound, MoodLogic, CDNow, SongExplorer), films (Reel.com Movie Matcher, MovieFinder.com Mach Maker), cosmetics (Drugstore.com) and other. But also recommender systems are applied when advice is sought by firms when selecting training courses for employees, job search for unemployed etc.

Recommender systems are not only applied to increase sales, but also to expand cross-selling, attraction of new clients, extending of trust of existent clients, to overcome the barriers of mass advertisement through high personalization of offers. The intense research did not, however, overcome many important limitations, like sparseness of data [21], scalability and real time action, detail level of available data, feedback to the sales firm [22], protection against preference manipulation, visualization of recommendation acceptability reasons [15], client modeling, system evaluation, creation of distributed systems [3], etc.

In our on-going research effort, we target at recommender systems capable of overcoming the limitations of present-day systems with respect to problems of rare data in recommendation, scalability and visualization of recommendation

for the purpose of proper explanation and justification of recommendation. Results of this research will surely lead to practical guidelines for construction of commercial recommender systems, where above mentioned problems are crucial.

We have created a full-fledged search engine BEATCA for small collections of documents (up to several millions) capable of representing on-line replies to queries in graphical form on a document map. We extended WebSOM's goals by a multilingual approach, new forms of geometrical representation and we experimented also with various modifications to the clustering process itself [17, 18] . The crucial issue for understanding the 2D map by the user is the clustering of its contents and appropriate labeling of the clustered map areas.

Several important issues need to be resolved in order to present the user with an understandable map of documents. The first issue is the way of document clustering. In the domains, like e.g. legal documents, where the concepts are not sharply separated, a fuzzy-set theoretic approach to clustering appears to be a promising one. The other one is the issue of initialization of topical maps. Our experiments showed that the random initialization performed in the original WebSOM may not lead to appearance of meaningful structure of the map. Therefore, we proposed several methods for topical map initialization, based on SVD, PHITS, PLSA and Bayesian network techniques.

In this paper, before we report on our current state of research effort starting with Sect. 4, we briefly present an overview of recommender system concepts (Sect. 2) and our concept of an integrated recommender system (Sect. 3).

2 Recommender systems overview

Intelligent agent (IA) is a user's assistant or recommender system based on machine learning and data mining techniques. Construction of IA uses the following paradigm taken from the ordinary life: "people uses helpful information without any fixed plans". People do not need and cannot describe their own work in terms of coefficients and classification. People just operate and know what they are interesting in, or what they want when they see it.

Recommender system simulates some social behavior. No one has unlimited knowledge and such knowledge is not necessary on daily basis. However, in some decision problems we have to go into details of specific, narrow knowledge. Sometimes there is a possibility to use advice from experienced person (expert) in a given area. Recommender systems try to help user in such situation by using knowledge collected in specified discipline and watching decisions made by other users decisions in the similar case. These systems have been built as a help in decision process for people, but also for multi-agent systems and generally speaking for systems consisting of objects that have limited knowledge about environment. Recommender system uses knowledge about passive objects to recommend next (somehow similar) item to active objects. For example, recommender system can recommend next web page or article in Internet shop (passive object) that user (active object) is probably looking for.

Recommender systems may be classified along the following criteria: amount and type of data that come from active object, amount and type of required data about community of active objects, method of recommendation, result type of recommendation, way of delivering recommendation to the active object and the degree of personalization (adaptivity to active object characteristic). More detailed classification and examples of commercial recommender systems can be found in [22].

Methods of recommendation in early systems were based mostly on the following approaches: recommendations based on searching, categories, clustering, association rules or classifiers. Finally, evolution of recommender systems has led to two major approaches in construction of IA:

1. Content-based approach. System creates users profiles by analyzing their operations and recommends documents that are compatible with these profiles.
2. Collaborative approach - collaborative or social filtering [12]. System focuses on a group of users.

The first approach, rooted in the tradition of information processing, is applicable if the system deals with text only. The system seeks information similar to that preferred by the user. If a user is interested in some knowledge areas (represented by documents described by some keywords or phrases) then the recommender looks for documents with similar content to already articulated. The basic problem here is to capture all specific aspects of a document content (e.g. in disciplines such as music, film, computer-related issues etc.). Even restricting recommendations to text documents only, most representations are able to cover only some aspects of document content, which results in weak quality of presented recommendations.

The second approach, called also social learning, relies on exploiting reactions of other users to the same object (e.g. a course, educational path, a film, etc.). The system looks for users with similar interests, capabilities etc. and recommends them information or items they are searching for.

This approach allows for posing questions like "show me information I have never seen but it turned interesting to people like me". Personalized information is provided in an iterative process where information is presented and user is asked to rank it, what allows to determine his/her profile. This profile is next used to locate other users with similar interests, in order to identify groups with similar interests.

Instead of calculating similarity between documents, this method determines degree of membership to a group (for example based on surveys). In contrast to first approach, it does not require analysis of document content, what means that document with arbitrary content could be presented to the user, with the same probability. Each document is assigned with identifier and a degree of membership to a group.

This approach is characterized by two features: first of all the document relevance is determined in the context of the group and not of a single user. Second, evaluation of the document is subjective. Hence one can handle complex and heterogeneous evaluation schemas.

3 Integrated recommendations

Existing recommender systems, based on a paradigm of content-based filtering as well as those based on collective filtering principle, do not take into consideration possible synergic effects. Such effects emerge when:

- both methodologies are merged,
- system is able to model joint, integrated recommendation of passive and active objects (i.e. clients and products), and not only passive objects pointed by active ones,
- recommendations are based on visual system, which helps to explain and justify a recommendation.

Application of joint methodology is possible if available data contain information on recommended objects as well as relations between recommended and recommending objects. Such information is present, e.g. in WWW documents, where individual html pages have not only textual context, but also hyperlinks between them. From logs saved on a particular host one can obtain so-called click-stream of users surfing from one page to another, and some additional data such as voluntarily filled-in questionnaires. Among other examples are libraries, book stores, or any shop (including e-shops), where products can be described by a set of attributes (e.g. advertisement leaflet) and users can be identified by some ID cards (e.g. loyalty program participation cards). Similarly, for some services (e.g. concerning education or health), both pointed(passive) and pointing(active) objects are described by attributes.

By an *integrated recommendation* we mean recommendation such as "People interested in <characteristics of people> are buying also book <title>" (instead of typical recommendation in form: "People interested in <title> are buying also book <title>"). Thus, integrated recommendation requires that system has an ability to generalize features describing characteristics of active objects (i.e. users or clients).

Recommendation with a visual explanation and justification is a completely new approach, based on creation of two-dimensional, navigational map of objects. Such a map yields a possibility to present an identified area of user's interests together with surrounding context, i.e. main directions of his/her future activities.

4 BEATCA search engine

Our first step towards a new model of recommendation system was to create a new-type search engine, based on a document map interface. Our map-based approach to search engine interfacing comprises two important features from the point of view of the target recommendation system: providing an overview over the whole collection of objects, and a very detailed clustering into groups of objects and their immediate (local) contexts.

With a strongly parameterized map creation process, the user of BEATCA can accommodate map generation to his particular needs, or even generate multiple maps covering different aspects of document collection. The overall complexity of the map creation process, resulting in long run times, as well as the need to avoid "revolutionary" changes of the image of the whole document collection, require an incremental process of accommodation of new incoming documents into the collection.

Within the BEATCA project we have devoted much effort to enable such a gradual growth. In this study, we investigated vertical (new topics) and horizontal (new documents on current topics) growth of document collection and its effects on the map formation capability of the system. To ensure intrinsic incremental formation of the map, all the computation-intense stages involved in the process of map formation (crawling, indexing, GNG clustering, SOM clustering) need to be reformulated in terms of incremental growth.

In particular, Bayesian Network driven crawler is capable of collecting documents around an increasing number of distinct topics. The crawler learning process runs in a kind of horizontal growth loop while it keeps its performance with increasing number of documents collected. It may also grow vertically, as the user can add new topics for searching.

In the next section we briefly mention our efforts to create a crawler, that can collect documents from the internet devoted to a selected set of topics. The crawler learning process runs in a kind of horizontal growth loop while it improves its performance with increase of the amount of documents collected. It may also grow vertically, as the user can add new topics of for search during its run time.

4.1 Intelligent topic-sensitive crawling

The aim of intelligent crawling [1] is to crawl efficiently documents belonging to certain topics. Often it is particularly useful not to download each possible document, but only that which concerns a certain subject. In our approach we use Bayesian nets (BN) and HAL algorithm to predict relevance of documents to be downloaded.

Topic-sensitive crawler begins processing from several initial links, specified by the user. To describe a topic of our interest, we use query document. This special pseudo document contains descriptive terms with a priori given weights, which are later used to calculate priorities for crawled documents. During crawling first few hundred documents, crawler behavior depends only on initial query. Later the query is expanded by BN or HAL methods described below.

4.1.1 Bayesian net document query expansion At increasing time intervals, we build Bayesian Net by using ETC learning algorithm [16] to approximate term co-occurence in topical areas. We use them to expand query and to calculate priorities for further documents links. We expand query by adding parent and children nodes of BN terms, which are already present in query document.

New terms get weights proportional to the product of the likelihood of their co-occurrence and the weight of the original term.

4.1.2 HAL document query expansion To expand query document we also use HAL (*Hyperspace Analogue To Language*, [20]) model. It is based on psychological theory claiming that meaning of a word is a function of contexts in which it appears; and the words sharing contexts have similar meanings. From computational perspective, HAL model can be represented as a matrix H in which cell h_{ij} corresponds to similarity measure of terms i and j.

Like in the BN algorithm, final document links priorities are calculated by modified cosine measure between new expanded query document and document containing those links.

4.1.3 Evaluation To see, how effective the proposed topic sensitive crawkling is, We run two experiments, one for BN algorithm, the other for HAL algorithm [5]. In both cases, three seed links [`http://java.sun.com/j2ee/index.jsp`, `http://java.sun.com/products/ejb/`, `http://www.javaskyline.com/learning.html`] served as starting points. We used a query consisting of six weighed descriptive terms, [java(with weight of 20) documentation(30) ejb(100) application(50) server(50) J2EE(30)]. Figure 1(a) depicts results for crawler based on BN algorithm and figure 1(b) presents results for crawler based on HAL algorithm.

Quality measure is the average relevance measure, computed after every 500 new document downloads. Relevance is equal to modified cosine measure, but only for terms which are present in the initial user query ($Q = Q_0$), i.e. $relevance = cos(q_0, d)$.

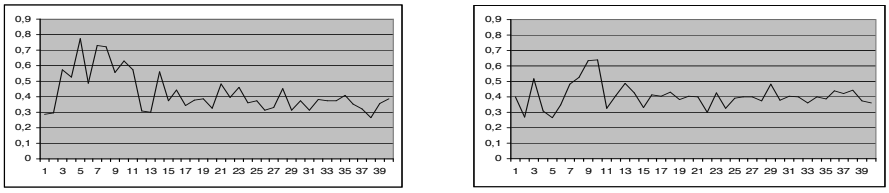

Fig. 1. Crawler evaluation (20000 documents downloaded): (a) Bayesian Net algorithm (b) HAL algorithm

Both methods appear to be satisfactory: average cosine measure amounts 0.4. The crawler does not lose a priori defined topic during the crawl. BN proved to be faster of the two methods, but it requires to stop whole process in order to rebuild BN model. HAL table can be built during the crawl, but it requires more computations.

4.2 Map creation process outline

4.2.1 WebSOM approach One of main goals of the project is to create 2D document map in which geometrical vicinity would reflect conceptual closeness of documents in a given document set. Additional navigational information (based on hyperlinks between documents) is introduced to visualize directions and strength of between-group topical connections. Our starting point was widely-known Kohonen's Self-Organizing Map principle [19], which is an unsupervised learning neural network model, consisted of regular, 2D grid of neurons.

4.2.2 Growing Neural Gas approach Similarly to WebSOM, growing neural gas (GNG) can be viewed as topology learning algorithm, i.e. its aim is to find a topological structure which closely reflects the topology of a given collection of high-dimensional data. In typical SOM the number of units and topology of the map is predefined. As observed in [10], the choice of SOM structure is difficult, and the need to define a decay schedule for various features is problematic.

GNG starts with very few units and new units are inserted successively every each few iterations. To determine where to insert new units, local error measures are gathered during the adaptation process; new unit is inserted near the unit, which has accumulated maximal error. Interestingly, GNG cells of the GNG network are joined automatically by links, hence as a result a possibly disconnected graph is obtained, and its connected components can be treated as different data clusters. The complete GNG algorithm details and its comparison to numerous other soft competitive methods can be found in [11].

4.2.3 GNG with utility factor Typical problem in web mining applications is that processed data is constantly changing - some documents disappear or become obsolete, while other enter analysis. All this requires models which are able to adapt its structure quickly in response to non-stationary distribution changes. Thus, we decided to implement and use GNG with utility factor model, presented by Fritzke in [11].

A crucial concept here is to identify the least useful nodes and remove them from GNG network, enabling further node insertions in regions where they would be more necessary. The utility factor of each node reflects its contribution to the total classification error reduction. In other words, node utility is proportional to expected error growth if the particular node would have been removed. There are many possible choices for the utility factor. In our implementation, utility update rule of a winning node has been simply defined as $U_s = U_s + error_t - error_s$, where s is the index of the winning node, and t is the index of the second-best node (the one which would become the winner if the actual winning node would be non-existent). Newly inserted node utility is arbitrarily initialized to the mean of two nodes which have accumulated most of the error: $U_r = \frac{U_u + U_v}{2}$.

After utility update phase, a node k with the smallest utility is removed if the fraction $\frac{error_j}{U_k}$ is greater then some predefined threshold; where j is the node with the greatest accumulated error. Detailed description of the GNG-U algorithm can be found in [11].

4.2.4 GNG network visualization Despite many advantages over SOM approach, GNG has one serious drawback: high-dimensional networks cannot be easily visualized. Nevertheless, instead of single documents, we can build Kohonen map on GNG nodes reference vectors, treating each vector as a centroid representing a cluster of documents. Such a map is initialized in the same way as underlying GNG network (i.e. with the same broad topics) and next is learned in the usual manner. The resulting map is a visualization of GNG network with the detail level depending on the SOM size (since a single SOM cell can gather more than one GNG node). User can access document content via corresponding GNG node, which in turn can be accessed via SOM node - interface here is similar to the hierarchical SOM map case.

4.2.5 PHITS technique Alternatively to content-only based representation one can build a map which will visualize a model of linking patterns. In such model, document is represented as sparse vector, whose i-th component equals to path length *from* [via outcoming links] or *to* [via incoming links] to i-th document in a given collection. It is usually assumed, that these computations can be restricted to the paths of maximum length **5** and above this value document similarities are insignificant. It is also possible to estimate a joint term-citation model.

PHITS algorithm [6] does the same with link information as PLSA algorithm [13] with terms contained in a document. From mathematical point of view, PHITS is identical to PLSA, with one distinction: instead of modeling the citations contained within a document (corresponding to PLSA modeling of terms in a document), PHITS models "in-links," the citations to a document. It substitutes a citation-source probability estimate for PLSA term probability estimate. On the Web and in other document collections, usually both links and terms could or should be used for document clustering. The mathematical similarity of PLSA and PHITS enables to create a joint clustering algorithm [6], taking into consideration both similarity based on document content and citation patterns.

5 Final Remarks

Modern man faces a rapid growth in the amount of written information. Therefore he needs a means of reducing the flow of information by concentrating on major topics in the document flow. In order to achieve this, he needs a suitable recommendation system.

Grouping documents based on similar contents may be helpful in this context as it provides the user with meaningful classes or clusters. Document clustering and classification techniques help significantly in organizing documents in this way. A prominent position among these techniques is taken by the WebSOM of Kohonen and co-workers [19]. However, the overwhelming majority of the existing document clustering and classification approaches rely on the assumption

that the particular structure of the currently available static document collection will not change in the future. This seems to be highly unrealistic, because both the interests of the information consumer and of the information producers change over time.

A recent study described in [14] demonstrated deficiencies of various approaches to document organization under non-stationary environment conditions of growing document quantity. The mentioned paper pointed to weaknesses among others of the original SOM approach (which itself is adaptive to some extent) and proposed a novel dynamic self-organizing neural model, so-called Dynamic Adaptive Self-Organising Hybrid (DASH) model. Other strategies like that of [9], attempt to capture the move of topics, enlarge dynamically the document map (by adding new cells, not necessarily on a rectangle map).

We take a different perspective in this paper claiming that the adaptive and incremental nature of a document-map-based search engine cannot be confined to the map creation stage alone and in fact engages all the preceding stages of the whole document analysis process.

Though one could imagine that such an accommodation could be achieved by "brute force" (learning from scratch whenever new documents arrive), there exists a fundamental technical obstacle for such a procedure: the processing time. The problem is even deeper and has a "second bottom": the clustering methods like those of SOM contain elements of randomness so that even re-clustering of the same document collection may lead to changes in resulting map.

The important contribution of our research effort so far is to demonstrate, that the whole incremental machinery not only works, but it works efficiently, both in terms of computation time, model quality and usability. . At the same time, it comes close to the speed of local search and is not directly dependent on the size of the model. This means that it is possible to target at large scale recommendation systems with a visual map-based interface.

Our investigation into influence of crawling via an intelligent crawling agent on the quality of the created document maps indicates a positive impact of this type of crawler on the overall map quality. This, together with known investigations of combined PLSA/PHITS model, seems to be an encouraging confirmation of our assumption that combining content-based and collaborative filtering may provide foundations for a more reliable recommendation.

Also apparently the new methods for creation of stable maps, that we propose, are successful to the extent that we may be able to develop visual recommendation justification in which changes in visual patterns will be attributed to real changes of user preferences and not due to artifacts of map construction algorithms.

References

1. C.C. Aggarwal, F. Al-Garawi, P.S. Yu: Intelligent crawling on the World Wide Web with arbitrary predicates. In Proc. 10th Int. World Wide Web Conference, pp. 96–105, 2001.

2. J.S. Breese, D. Heckerman, and D.C. Kadie: Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998, pp. 43-52.
3. J. Callan, et. al: Personalisation and recommender systems in digital libraries, Joint NSF-EU DELOS Working Group Report, May 2003 http://www.dli2.nsf.gov/internationalprojects/working_group_reports/personalisation.html
4. S. Cayzer, U. Aickelin: A Recommender System based on Idiotypic Artificial Immune Networks, J. of Mathematical Modelling and Algorithms, 4(2)2005, 181-198
5. K. Ciesielski et al.: Adaptive document maps. In: Proc. IIPWM'06, Springer.
6. D. Cohn, H. Chang: Learning to probabilistically identify authoritative documents, Proceedings of the 17th International Conference on Machine Learning, 2000
7. M.W. Berry: Large scale singular value decompositions, Int. Journal of Supercomputer Applications, 6(1), 1992, pp.13-49
8. R. Decker: Identifying patterns in buying behavior by means of growing neural gas network, Operations Research Conference, Heidelberg, 2003
9. M. Dittenbach, A. Rauber, D. Merkl: Discovering hierarchical structure in data using the growing hierarchical Self-Organizing Map, Neurocomputing, 48 (1-4)2002, pp. 199-216
10. B. Fritzke: A growing neural gas network learns topologies, in: G. Tesauro, D.S. Touretzky, and T.K. Leen (Eds.) Advances in Neural Information Processing Systems 7, MIT Press Cambridge, MA, 1995, pp. 625-632
11. B. Fritzke, A self-organizing network that can follow non-stationary distributions, in: Proc. of the Int. Conference on Artificial Neural Networks '97, 1997, 613-618
12. D. Goldberg, D. Nichols, B.M. Oki, D. Terry: Using collaborative filtering to weave an information tapestry, Communication of the ACM, 35:61-70, 1992.
13. T. Hoffmann: Probabilistic latent semantic analysis, in: Proceedings of the 15th Conference on Uncertainty in AI, 1999
14. C. Hung, S. Wermter: A constructive and hierarchical self-organising model in a non-stationary environment, Int. Joint Conference in Neural Networks, 2005
15. A. Jameson: More than the sum of its Mmmbers: Challenges for group recommender. Proc. of the Int. Working Conference on Advanced Visual Interfaces, Gallipoli, Italy, 2004 http://dfki.de/~jameson/pdf/avi04.jameson-long.pdf
16. M. Kłopotek: A new Bayesian tree learning method with reduced time and space complexity, Fundamenta Informaticae, 49(4) 2002, IOS Press, pp. 349-367
17. M. Kłopotek, M. Dramiński, K. Ciesielski, M. Kujawiak, S.T. Wierzchoń: Mining document maps, in Proceedings of Statistical Approaches to Web Mining Workshop (SAWM) at PKDD'04, M. Gori, M. Celi, M. Nanni eds., Pisa, 2004, pp.87-98
18. M. Kłopotek, S.T. Wierzchoń, K. Ciesielski, M. Dramiński, M. Kujawiak: Coexistence of fuzzy and crisp concepts in document maps, in: Proc. of the Int. Conference on Artificial Neural Networks (ICANN 2005), LNAI 3697, Springer-Verlag, 2005
19. T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, vol. 30, Springer, 2001
20. B. C., K. Livesay, and K. Lund. Explorations in context space: Words, sentences, discourse. Discourse Processes, 25(2-3):211–257, 1998
21. Sarwar, G. Karypis, J. Konstan, J. Riedl: Item-based Collaborative Filtering Recommendation Algorithms, WWW10, May 1-5, 2001, Hong Kong
22. J. B. Schafer, J. Konstan, J. Riedl: Electronic Commerce Recommender Applications, Journal of Data Mining and Knowledge Discovery, 5(1-2): 115–152, 2001
23. U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In ACM Conference Proceedings on Human Factors in Computing Systems, pages 210–217, Denver, CO, May 7-11 1995.