Proceedings of the IASTED International Conference
APPLIED INFORMATICS
February 18-21, 2002, Innsbruck, Austria

A Software Agent for Adaptive Visualization of Hyperspace Based on Clustering of Navigation Graphs

Dirk Kukulenz
Institute of Computer Science
University Kiel
Preusserstr. 1-9
24105 Kiel, Germany
email: dku@ks.informatik.uni-kiel.de

Josef Pauli
Institute of Computer Science
University Kiel
Preusserstr. 1-9
24105 Kiel, Germany
email: jpa@ks.informatik.uni-kiel.de

ABSTRACT

A software agent will be described that makes it possible to provide the visitor of an internet site with adaptive maps of the hyperlink structure. The maps represent estimations of important navigation decisions for the user depending on his own former partial navigation path and navigation information provided by other users. A user's navigation behaviour is characterized by the set of his navigation decisions in the hyperlink structure (navigation graphs). The problem of finding a suitable characterization of the distribution of these navigation graphs and the development of an estimation procedure for this distribution will be described. This estimation can be reached by defining different metrices or distance functions between graphs like string-edit- or largest-common-subgraph distance and by application of nearest neighbourhood clustering. The knowledge concerning the distribution is then used for a classification of a user and thereby an adaptive hyperspace view presentation using graph matching techniques. A visualization example will be presented.

KEY WORDS

Software Agent, Collaborative Filtering, Clustering, Hyperspace Visualization

1 Introduction

The fast increasing amount of information in the internet makes it more and more necessary to examine the problem of finding information in hyperspace. Basically there are two different ways to look for information in the internet that are normally used in combination. It is possible to use a search engine where a request is done by search words. A second search strategy is the navigation or browsing between data objects that is made possible by the hyperlink structure. Browsing is suitable when a concrete question is difficult to formulate or when a web site has to be scanned according to certain aspects.

The system described in this article makes it possible to present an internet user a visualization of the local hyperlink structure similar to a map in the real world in order to make browsing easier. The actual layout strategy for the structure, which is an important problem in the field of information visualization [1], [2] is not considered here. Emphasis is laid on the estimation of relevant navigation decisions between data objects and sets of such decisions. This estimation can be used to simplify and improve the visualized hyperlink structure e.g. by omitting data and navigation decisions that are estimated to be not relevant.

Due to a similarity to systems presented in the field of intelligent software agents especially collaborative filtering agents as presented in [3] and [4], we will call the described software system a *software agent*.

In [5] a procedure was presented to apply a collaborative filtering technique for the creation of index lists based on sets of requested data objects. In [3] a navigation support system is presented that learns from search words and browsing decisions of users, applying a reinforcement learning technique. In the system described here, however, it will be assumed that no search words are available. In [6] and [7] techniques for presending documents on the WWW are described that apply Markov-learning techniques and can also be used to learn the relevance of data objects for a specific client from former profiles of other users.

The system described here applies a different (unsupervised) learning technique. The agent actively registrates navigation decisions between data objects, a method that can easily be applied to an access analysis for multiple servers, too [8]. Having measured the sets of navigation decisions of clients we are going to apply clustering techniques as described in [9] using results about graph metrices as presented in the field of pattern recognition [10], [11]. The computed clusters can be used for the classification of a new profile and thus a prediction of future decisions and especially a relevance estimation is possible. The system will be tested using simulated data; the described procedure can however easily be applied to real navigation profiles.

The advantage of this technique compared to Markov models is that we don't have to think about the order of the Markov process. Such a predefinition may cause classification errors or otherwise cause an unnecessary increase of complexity. Moreover, we don't work with sets of requested data objects but with sets of navigation decisions which we suppose to contain more information. The described strategy is similar to procedures in the field of data mining [12],[13], however we will not deal with very large databases but we hope to achieve a better segmentation of the set of navigation profiles.

In section 2 the structure of the agent will be described, laying emphasis on the profile registration component. In the third section a technique for the estimation of graph distibutions will be described and the applicability for a prediction of navigation profiles. Some properties of the estimation technique will be shown and a visualization example is presented. Section 4 gives a summary and mentions further research issues.

2 Agent description and data acquisition

In this section the structure of the agent is described, laying emphasis on the technique to registrate navigation decisions of a specific client on the server's side. The visualization component of the agent was presented in [8]. The structure of the software system can be seen in figure 1. It is shown that the agent, represented by the ellipse on the left acts between the internet client and server. The ellipse on the right represents a number of internet servers, e.g. the servers of a university.

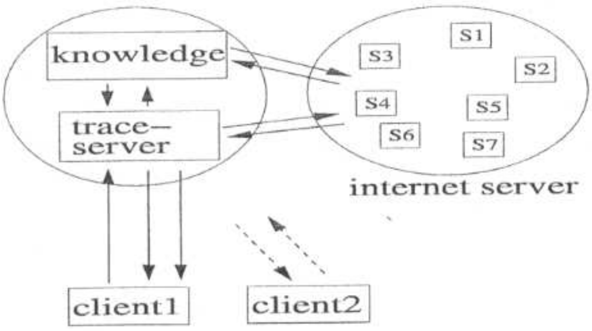

Figure 1. Structure of the navigation support tool.

The tracing technique for user navigations is similar to a method described in [3] but it is quite different from a proxy server method applied e.g. in [14]. The navigation support starts by sending a prepared web page to the user, which can be done automatically if the user isn't registrated yet. The page is modified in a way that all hyperlinks i.e. URL addresses point to the IP-address of the agent. The modified hyperlinks contain additional parameters like the original address where the hyperlink originally pointed to, the address of the page where the hyperlink is located and an identification number being assigned to a specific client. The address 'http://server/origpage' is modified to 'http://agent?server/origpage+frompage+id' where 'agent' is the web address of the agent and 'server' is the original server containing the web page, 'origpage' is the address of the page, 'frompage' is the address of the page where the hyperlink is located, 'id' is the identification number. It is obvious that in this way the agent can registrate and seperate the navigation decisions of internet users on the considered website. As it may happen that a user applies a search engine during his navigation, the agent should be combined with a local search engine that replaces hyperlinks and communicates with the agent in a similar way. The different profiles are stored in the agent component denoted as 'knowledge' in figure 1.

One disadvantage of this method compared to a proxy server technique is the effort to realize the modification step in the case of web-pages containing e.g. JavaScript or Java. However, the provided information is more accurate than the usual access-log information stored by internet servers. Due to caching techniques in the internet we may only get a subset of the actual navigation decisions of users. Therefore in the following we model user profiles by the set (and not a sequence) of navigation decisions.

3 Estimation of graph distributions

3.1 Definitions and model

As described in the preceding section, the information we get about a specific internet user on the server side is the set (or at least a subset) of his navigation decisions. These navigation decisions take place between certain data objects being available on the considered internet site, like web pages, images, scripts, etc . Let 'D' denote the set of data objects in the internet site, having an (own) URL address.

A user profile, measured by the agent, is then a graph structure:

DEF. 1 *A (profile-) graph or navigation profile is a 4-Tupel $G = (V, E, \mu, \nu)$. V is a set of knots and $E \subseteq V \times V$ is a set of edges. Function $\mu : V \to L_V \subset D$ assigns labels to the knots. Function $\nu : E \to L_E$ assigns labels to the edges.*

Let $< G >$ be the set of all graphs following the preceding definition. This set will be denoted as 'graph space' based on D. Let $\{G\} \subseteq < G >$ denote a set of graphs.

L_V is a subset of D or a set of pointers to D. The edges considered here are in the most common case hyperlinks that are present on certain web pages, Java applets or scripts. However, with the help of a search engine, the user can get from one data object to possibly any other object.

In the following sections the definitions of a subgraph, a graph and subgraph isomorphism, graph-edit operations and an error correcting subgraph isomorphism are used that are common in the field of graph theory or artificial intelligence and that are presented e.g. in [15] and [10].

3.2 Characterizations of graph distributions

It is our aim to classify a new profile graph according to a set of former profiles supplied by users. For this purpose it is helpful to know the distribution of graph profiles or at least to get an idea of the shape of this distribution. It is possible to regard $< G >$ as a discrete set and to assign a probability value to each element depending on the relative frequency. However, graphs may be similar according to certain aspects which may not be taken into account by the discrete formulation.

It is very likely that people having the same question in mind produce similar navigation profiles that are however slightly distorted because of internet caching, different starting points or different searching strategies. Vice versa, similar profiles are likely to result from similar questions or intentions of users which is the main assumption we make [16], [17]. We therefore assume that the profiles are distributed in a way that one or a number of profiles in some 'places' in the graph space have a high likelihood and the other profiles being more and more distant from one of these 'central' profiles have a decreasing likelihood with respect to a distance function that will be defined in section 3.3. This distribution can then be characterized by a function:

DEF. 2 *Charac1:* $\{G\} \longrightarrow \{1,..n\}$

Here, every profile is associated with one of the clusters.

Another method is to consider the centres of the clusters and to take into account some characteristics of the inner cluster structure:

DEF. 3 *Charac2:* $\bigcup_{i=1,..n}\{(\mu_i,\sigma_i,A_i)\}$,

where μ_i is the centre graph of cluster i, σ_i is a measure for the distribution within the cluster, e.g. the mean value of the distances of the elements in the cluster i from the centre element μ_i and A_i is the number of elements in the cluster. The centre values μ_i can easily be found from *Charac.1* by determining the element in the cluster with the smallest sum of the distances to all the other elements in the same cluster.

In the following we will use a simplification of the graph distribution characterization in definition 3 by taking only the centre points, i.e. we define *Charac2'*: $\bigcup_{i=1,..n}\{\mu_i\}$,

3.3 Graph metrices

The 'shape' of the graph distribution as being characterized by definition 2 or 3 depends strongly not only on the data elements but also on the distance measure between graphs. Several definitions of graph distances are known from the field of pattern recognition.

A simple idea to define such a distance function is to count the number of identical knots. To achieve a better segmentation of the set of graph profiles however, the structure of the connections i.e. edges in the graphs should be taken into account, too. A measure for such a structural similarity is the size of the largest common subgraph. In [11] it was shown that for two graphs G_1 und G_2 and the largest common subgraph $lcS(G_1, G_2)$ the function

DEF. 4 $d(G_1, G_2) := 1 - \frac{|lcS(G_1,G_2)|}{max(|G_1|,|G_2|)}$

has the mathematical properties of a metrics ($|.|$ denotes the number of nodes in a graph). A similar graph distance was defined in [18].

The disadvantage of this metrics is that possible similarities between different knots can't be taken into account. Such similarities between the type of knots that are considered here, i.e. data objects, have been examined for textual data in the field of information retrieval [19]. They are important for the automatic indexing of web pages for the realization of search engines. One well-known distance measure is the *tfidf*-Norm, in which text pages are converted into vectors of weights of words that can be compared with the help of the cosine between the vectors.

A distance measure for two graphs G_1 and G_2 making it possible to take such similarities into account is the following function, where Δ is a set of graph-edit operations and C is a cost function for the edit oprations as described in [10]:

DEF. 5 $d(G_1, G_2) := min_\Delta\{\, C(\Delta) \mid$ *there exists an error-correcting-subgraph-isomorphism* f_Δ *from* G_1 *to* $G_2\}$

The error-correcting-subgraph-isomorphism fulfilling the condition on the right is called optimal-error-correcting-(oec)-subgraph-isomorphism. However the distance function in definition 5 is not symmetric. In order to create a symmetric distance function it is possible to take the minimum of $d(G_1, G_2)$ and $d(G_2, G_1)$.

3.4 Estimation technique

The previously defined metrices or distance functions can now be applied to estimate the shape of a graph distribution considering the distribution characterization *Charac2'* in section 3.2. The navigation graphs can be clustered using a common clustering techniques like nearest neigbourhood clustering as described in [9] and by using one of the distance functions given in section 3.3. Further investigations concerning the shape of the inner cluster distributions according to *Charac2* in section 3.2 can then be made.

In order to measure the quality of such a distribution estimation it may be helpful to determine the distance between a real distribution that is known in advance and an estimation of this distribution.

Let $G_1, ..G_n$ be the elements in $\{G\}$, $H_1, ..H_m$ $(m \leq n)$ be the real cluster centres characterizing the graph distribution and $d(G_1, G_2)$ be the distance between two graphs according to one of the definitions in section 3.3. Let $\delta(G) := min\{d(G, H_j)|j = 1, ..m\}$ with $G \in \{G\}$.

DEF. 6 *Given an estimation of the cluster centres $\hat{H}_1, ..\hat{H}_m$, let err* $:= \sum_{i=1,..m} \delta(\hat{H}_i)$.

Obviously, *err* decreases, if the estimation result gets better i.e. the estimated cluster centres move towards the real ones, given that the starting points are sufficiently good.

Knowing the estimated distribution of navigation graphs we can describe a prediction technique to find future navigation steps of a specific user if we assume that the new profile follows the same distribution as the former ones. One way is to compare the new navigation profile to the estimated cluster centres and to find the closest centre. Given the cluster centres $\hat{H}_1, ..\hat{H}_m$ and the new profile G, in this method $d1_j := d(G, \hat{H}_j)$ has to be minimized in j where $d(G, \hat{H}_j)$ is a distance of G to the cluster centre $\hat{H}_j$ as defined in section 3.3. This centre element $\hat{H}_j$ can then be expected to have a high relevance for the user.

A further possibility is to take into account the absolute probability that a user profile belongs to a cluster. This probability can be estimated by the relative number of elements in the cluster. The minimization of $d2_j := d(G, \hat{H}_j)\frac{1}{1+A_j/A}$ in j takes this absolute probability into account, where A is the number of observed profiles, A_j is the number of patterns in cluster j. These functions will be tested in the following.

3.5 Simulation and visualization examples

It is the aim to show some of the properties of the described distribution estimation and classification with simulated navigation profiles where the distribution (i.e. *Charac2*' in section 3.2) of the original data is known in advance and can be compared to the estimation results. The simulation process starts by defining a graph space $< G >$ as defined in section 3.1. A number of graphs will then be computed randomly with equal distribution, the number of knots being identical and a fix number of edges. These graphs represent the real centre graphs. Then a sequence of graphs will be computed presenting the simulated graph data. Each graph is obtained by randomly choosing one of the real centre graphs randomly and a number for the label errors. The error value is choosen according to a discrete Gaussian $N(0, \sigma)$ distribution. The simulated graph is computed by changing a number of knot labels of the centre graph, equal to the number of label errors.

In figure 2 the dependence of the estimation quality according to definition 6 on the number of graphs in the sequence of navigation profiles is shown. The number of elements in D is 30, the number of knots in each graph is 25, with 30 edges. The graphs were computed from 2 original graphs (m=2), constituting the real distribution characterization. The number of identical simulations was 10. In figure 2 each value is the mean value of the estimation errors in the identical simulations. The graph metrics applied here for the clustering and the estimation quality measurement was the subgraph metrics in definition 4. As can be expected, the estimation error decreases, when the number of graphs increases since more information about the distribution is available for the estimation process.

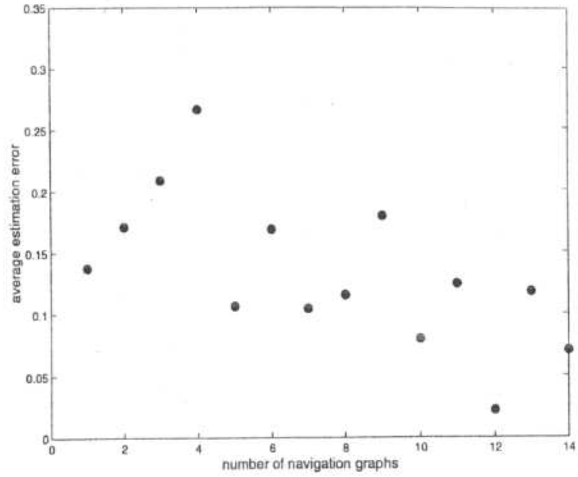

Figure 2. Dependence of the estimation error on number of graphs

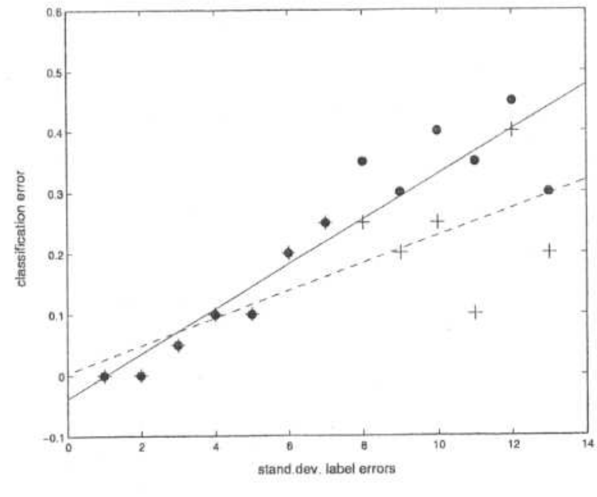

Figure 3. Classification experiment of user profiles by minimizing d1 (•) and d2 (+) in section 3.4.

In a second experiment we examined the prediction quality supposing that the distribution characterization is already known. A number of profiles was generated, following this distribution as described above. The percentage ($\times\frac{1}{100}$) of missclassifications was determined, denoted as 'classification error'.

Fig. 3 shows the classification error based upon the mimimization of $d1$ (•) and $d2$ (+) in section 3.4. In the experiment the deviation of label errors is changed. As can be seen, the prediction based upon minimization of $d2$ shows better results for higher values of the label error. This result was expected since more information about the shape

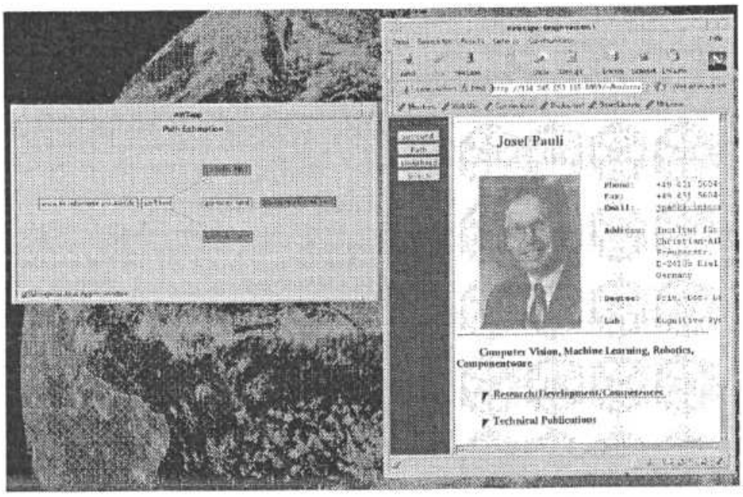
Path Estimation
Josef Pauli
Phone:
Fax:
Email:
Address:
Germany
Degree:
Lab:
Computer Vision, Machine Learning, Robotics, Componentware
Research/Development/Competences
Technical Publications

Figure 4. A visualization example: The window on the right shows the browser software presenting a webpage. On the left, a window with navigation decisions can be seen, estimated to be relevant.

of the distribution is used in the case of $d2$.

The algorithms can easily be applied to real data (however without any quality measure like in the simulations yet). In figure 4 it is shown, how the estimation results can be used to present relevant navigation decisions applying a visualization technique presented in [8]. In the right window in figure 4, the browser software, in this case Netscape, can be seen, showing the original web page. The modified hyperlinks according to section 2 don't change the presentation of the page. The left window outside the browser window shows the actual path of the user with navigation steps estimated to be relevant.

4 Conclusion and further aspects

One possibility to prevent a user from getting confused in the internet is the presentation of maps showing the hyperlink structure. This presentation can be improved, if only relevant data objects and (sequences of) navigation decisions are shown. In the article an estimation technique was presented that applies a graph clustering technique with the help of a definition of a distance between graphs. This process provides a characterization of the distribution of graphs which is difficult to describe directly. This characterization can then be applied for a relevance estimation presuming that the new navigation graph follows the same distribution.

Some properties of the algorithm were shown by applying a simulation procedure. The advantage of the use of simulated data is the knowlege about the distribution that isn't known for real data.

This method has the advantage compared to Markov modelling that a multi-step-prediction can easily be done and that not only sequences of navigation steps but also navigation graphs i.e. sets of navigation steps can be taken into account. A graph modelling of user decisions can be of advantage if e.g. caching strategies in the internet cause distorted navigation profiles or if the actual navigation decisions have to be considered, too. The article describes a software structure making it possible to track navigation decisions of users on the server's side and to use these data for the relevance estimation.

There are more refined methods to describe a distribution of graphs conceivable. A first improved method is given in definition 3, however further improvements should be developed. Different and more refined graph distances can be defined e.g. taking into account knot distances as described in definition 5. Additionally the prediction quality has to be examined closely for real data. The time requirements of the prediction algorithm are very important because this step has to be done in real-time. Further improvements of the system with respect to information visualization or with respect to learning from additional information about a user or the internet site are of interest.

References

[1] G.W. Furnas. Effective view navigation. In *Readings in Information Visualization, Using Vision to Think*, pages 589–596, 1999.

[2] I. Herman, G.Melancon, and M.S.Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transaction on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[3] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. Web watcher: A learning apprentice for the www. In *AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, pages 6–12, 1995.

[4] Henry Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 924–929. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995.

[5] Mike Perkowitz and Oren Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118(1–2):245–275, 2000.

[6] I. Zukerman, D.Albrecht, and A.Nicholson. Predicting users' requests on the www. In *UM99 – Proceedings of the Seventh International Conference on User Modeling*, 1999.

[7] R. Sarukkai. Link prediction and path analysis using markov chains. In *Intern. World Wide Web Conf.*, 2000.

[8] D. Kukulenz and J. Pauli. Navigation-dependent visualization of distributed internet structures. In *IEEE Conference on Information Visualization*, pages 518–523, 2000.

[9] B.S. Everitt. *Cluster Analysis*. Edward Arnold, 3 edition, 1993.

[10] B. Messmer and H. Bunke. *Efficient graph matching algorithms for preprocessed model graphs*. PhD thesis, Bern University, 1996.

[11] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. In *Pattern Recognition Letters*, volume 19, pages 255–259, 1998.

[12] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc.of the ACM SIGMOD Conference on Management of Data*, 1993.

[13] M. Chen, J.S. Park, and P.S. Yu. Data mining for path traversal patterns in a web environment. In *Proc. of the 16th ICDCS*, volume 16, pages 385–392, 1996.

[14] A. Wexelblat and P. Maes. Footprints: History-rich tools for information foraging. In *CHI99*, pages 270–277, 1999.

[15] Bollobas. *Graph theory*. Springer, 3 edition, 1999.

[16] E. Carmel, S. Crawford, and H. Chen. Browsing in hypertext: A cognitive study. In *Transactions on System, Man and Cybernetics*, volume 22, pages 865–883, 1992.

[17] C. Hoelscher and G. Strube. Web search behavior of internet experts and newbies. In *World Wide Web Conf*, volume 9, 2000.

[18] W.D. Wallis, P. Shoubridge, M. Kraetz, and D. Ray. Graph distances using graph union. In *Pattern Recognition*, volume 22, pages 701–704, 2001.

[19] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–979, 1991.