Web Usage Mining with Inductive Logic Programming

Amund Tveit *

June 7, 2000

Abstract

This paper suggests an experimental approach of how to apply inductive logic programming in the discovery of web usage patterns in the form of first-order rules representing user sessions. Such rules may be used to improve the quality and the performance of a web site. The experiment has been done using the Progol Inductive Logic Programming System, and the data source are log files from an Apache web server.

Keywords: Inductive Logic Programming, Data Mining

1 Introduction

The main motivation for this work is to investigate an approach to extract declarative knowledge about Web user browsing and access patterns. The discovery of such knowledge is called Web usage mining [Coo97]. Declarative knowledge is selected because the hypothesis is that web log data are highly relational.

The goal for this paper is to find a possible representation to generate first order rules for Web site traffic patterns.

Discovery of Web usage patterns is of interest because it might be possible to use this knowledge to improve the quality of the Web service itself, or maybe of equal importance in a networked environment, to improve the access to the existing service by analyzing how the service has been previously accessed, entered and exited [Per97]. User motivation and reaction is also of interest, particularly in a sales or marketing scenario for an Electronic Commerce site [Zaa98]

Mathematical logic, and in particular first order logic has been the preferred representation for declarative knowledge, so it needs knowledge discovery techniques which can generate logical formulae from data [Bry99]. Inductive Logic Programming (ILP) provides such techniques [Mug95],[Mit97], and is the chosen framework in this paper. ILP has earlier been applied with good results in several data mining problems, e.g. in classification of text [Coh97], in protein structure prediction [Mug98] and in diagnosing Ferro Silicon furnaces [Alf96]

*Email: amundt@idi.ntnu.no, Tel: +47 7359 4480, Knowledge Systems Group, Department of Computer and Information Science, Norwegian University of Science and Technology, N-7491, Trondheim, Norway

1.1 Organization of this paper

First an introduction to the Web usage mining and the motivation for web usage mining of first order rules, second some related research, third some examples and overview of data sources, fourth the relevant background theory, fifth the main approach for the discovery of Web usage patterns, and finally the discussion and conclusion.

1.2 Example of patterns

Declarative knowledge about Web usage patterns in the form of rules is interesting because humans are used to understand if-then type rules, e.g. the pattern

Swedish users applying a Nokia 7110 browser are visiting few web pages per session

can be understood by most english speaking adults, and it can be transformed into such english text from the (machine readable) prolog rule (discovered pattern) below:

$$visited(S, low) \leftarrow domain(S, sweden), browser(S, nokia7110). \quad (1)$$

If one at the same time discovers the pattern below

$$visited(S, high) \leftarrow domain(S, sweden), browser(S, ericssonR320). \quad (2)$$

one might suspect that the service doesn't support the Nokia 7110 browser properly.
Rules such as the ones above can e.g. be communicated to web designers or publishers for them to determine what and if something has to be done to improve the web site.

1.3 Data sources for Web usage mining

The main information source for the discovery of Web usage patterns can be extracted from structured log files on a Web server.

Another source of information are cookies. Cookies are small (non-executable) files that are exchanged between the web server and the browser, and stored in the users browser. Cookies are frequently used to store information about user preferences (e.g. myYahoo), the content of shopping baskets (e.g. Amazon.com, but is also being used for session tracking and in a more accurate identification of users, particularly in targeted marketing and in the discovery of Web usage patterns.

Other sources of information can be data entered by the user on the Web site, or data collected from other sources (e.g. external databases or other web sites). By adding clickthrough scripts (e.g. CGI or PHP) one can potentially measure when users are leaving the site. (Search engines such as Google has started to register clickthroughs, probably both for web usage mining and collaborative filtering reasons).

2 Background theory

2.1 Knowledge discovery and Data Mining

Knowledge discovery is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [Fay96a]

Data mining is the application of one or several data mining algorithms in a knowledge discovery process with the purpose of extracting patterns from a set of example data and possibly some background knowledge. Applying ILP in automatic synthesis of logic rules can thus be considered as a Data Mining approach.

Web mining is data mining on the World Wide Web domain, it is used in two separate areas: The first called *Web content mining*, is based on hypermedia data on the Web. The second called *Web usage mining* is allready described [Coo97]. Web usage mining has been approached with several techiques, e.g. clustering methods, statistical methods and propositional logic systems

2.2 Overview of Inductive Logic Programming

Inductive logic programming (ILP) can be seen as the intersection between logic programming and inductive machine learning.

ILP is divided into two major subfields, interactive ILP and empirical ILP. Interactive ILP is most often used on relatively small data sets and the user has possibilities to participate during learning of clauses. Empirical ILP is automatic extraction of regularities from big and possibly imperferfect data sets without user interaction [Lav93].

Overview of predicate learning in ILP [Dze96]

Given

- a set of positive ($\mathcal{E}^+$) and negative ($\mathcal{E}^+$) training examples ground facts of an unknown predicate p,
- a concept description language $\mathcal{L}$, with syntactic restrictions on the definition of p,
- background knowledge $\mathcal{B}$, defining predicates q_i which may be used in the definition of p.

Find

- a definition $\mathcal{H}$ for p, expressed in $\mathcal{L}$, such that $\mathcal{H}$ is complete, i.e., $\forall e \varepsilon \mathcal{E}^+ : \mathcal{B} \wedge \mathcal{H} \vdash e$ consistent, i.e. $\forall e \varepsilon \mathcal{E}^- : \mathcal{B} \wedge \mathcal{H} \nvdash e$, with respect to $\mathcal{E}$ and $\mathcal{B}$.

Generalization with Inverse resolution

Inverse resolution is used to produce general clauses from specific examples by inverting the resolution rule of deductive inference [Lav93]. For generalization inverse substitution is used, which means replacing constants with variables and at the same time ensuring that the original example(s) can be restored by ordinary substitution. (See figure 1)

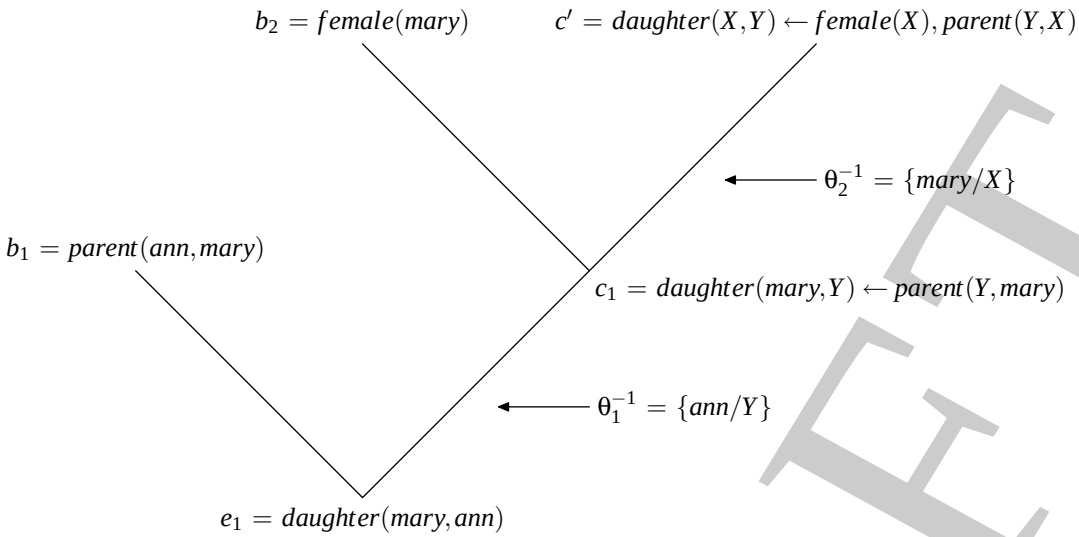

Figure 1: Inverse resolution

Generality ordering of clauses - Θ-subsumption

The Θ-Subsumption lattice induces a generality ordering on clauses, see definition below.

Definition 1 A clause $c_1\theta$-*subsumes a clause* c_2,denoted by $c_1 \preceq c_2$ iff $\exists\theta\, c_1\theta \subseteq c_2$. c_1 is a generalization of c_2 *under* θ-subsumption.

Relative Least General Generalization (rlgg)

Least general generalization (lgg) of two clauses is such that by applying two substitutions on the same variable in the lgg, two clauses will be regenerated, e.g.

$$lgg(parent(ann,mary), parent(ann,tom)) = parent(ann,X). \tag{3}$$

A rlgg is the most specific generalization of two clauses relative to the given background information, e.g.

$$C1 = daughter(mary,ann) \leftarrow female(mary), parent(ann,mary). \tag{4}$$

and

$$C2 = daughter(eve,tom) \leftarrow female(eve), parent(tom,eve). \tag{5}$$

then the least general generalization of the two clauses is

$$lgg(C1,C2) = daughter(X,Y) \leftarrow female(X), parent(Y,X). \tag{6}$$

Relative least general generalization is the least general clause more general than both C1 and C2 regarding to the background knowledge B.

2.3 ILP Software

Due to the size and structure of the web log data used, an empirical ILP system favorable over an interactive system. Some of the systems possible to select were Progol, FOIL and Golem [Bry97]. Due to good experience with Progol [Alf96], it became the selected system for this approach.

Overview of Progol [Mug95]

Progol searches a bounded sub-lattice for each example e relative to the background knowledge $\mathcal{B}$ and mode declaration $\mathcal{M}$. The sub-lattice has a most general element ($\top$) which is the empty clause, $\square$, and a least general element $\bot_i$ which is the most specific element in $\mathcal{L}_\rangle(\mathcal{M})$ such that

$$\mathcal{B} \wedge \bot_i \wedge \overline{e} \vdash_h \square \tag{7}$$

Definition 2 Mode declaration *A mode declaration has either the form* $modeh(n, atom)$ *or* $modeb(n, atom)$ *where* n, *the recall is either an integer,* $n > 1$, *or* $'*'$ *and atom is a ground atom.*

(modeh is used to constrain the syntactics of the head part of the clause, and modeb is the same for the body of the clause)

Definition 3 Most-specific clause $\bot_i$ *Let* h, i *be natural numbers B be a set of horn clauses,* $e = a \leftarrow b_1, \ldots, b_n$ *be a definite clause, M be a set of mode declarations containing exactly one modeh m such that* $a(m) \preceq a$ *and* $\bot$ *be the most − specific (potentially infinite) definite clause such that* $B \wedge \bot \wedge \overline{e} \vdash_h \square$. $\bot_i$ *is the most − specific clause in* $\mathcal{L}_\rangle(\mathcal{M})$ *such that* $\bot_i \preceq \bot$.

3 Main approach

The data for this experiment was 31 days of extended web logs from the web site agentus.com (February 10.-March 13., 2000). The web logs including the preprocessing scripts can be downloaded from here

3.1 What is interesting to discover knowledge about?

The chosen representation for left sides of the horn clause rules, for this experiment the user session is selected, rules generated by this left side can be used to check overall popularity of the site as well as more detailed analysis in the high or low range of visits based on combinations of body predicate values.

Sessions

Definition 4 Grouping unit - user session *A user session is a set of individual requests from the same IP address, andwhere the time between subsequent requests in the set must be less than the threshold* τ_{max}

In this approach $tau_{max} = 3600$ *seconds* (1h). (This definition is based [Nas99]).

3.2 The preprocessing steps were done using Perl scripts

step1a Concatenation of the 31 24h-period log files

step1b Translation of relative HTTP requests into full URL request

step1c Add user hostnames (looks up ip addresses)

step2a Maps log file into first order logic representation and group them in sessions

step3 Add progol headers to first order logic output (manually)

step4 Run progol on created input file (start manually)

step5 Calculate goodness of create rules (coverage and error) (manually)

3.3 Structure of extended web log file

The test data had extended logging on, the added feature values caused by extended logging are the refererring URL and the client type which has information about which link the user selected the request for the Web object from, and which type of browser that is used. Below is an example of a log entry refered from a search engine query for the string ”domene”[1] in the Norwegian language.

host ident authuser date request status bytes refererinfo clientinfo
212.125.164.252 - - [12/Mar/2000:11:16:54 -0500] ”GET / HTTP/1.1”
200 4115
”http://search.sol.no/kvasir/search.cgi?query=domene&language=no”
”Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)”

3.4 Example of log entry in first order logic

The above extended log file format example encoded into first order logic could be:

domain(session0,network). url(session0,0). browser(session0,explorer). os(session0,win98). osprice(session0,commercial). reftype(session0,searchengine).

(Information about time, HTTP request type and type of search engine query were not processed in this experiment)

[1] Domene in Norwegian means domain name

3.5 The Progol ILP system

Progol uses essentially a Prolog format, with some specification in the beginning of the file to tell the system what predicates and types that are allowed in the head and the body of the induced rules.

Progol Mode Declaration

The above listing shows some of the predicate specifications given to Progol. *modeh* means that the predicate is a part of the head of the induced clauses, and modeb means that it is in the literal body. The number specifies the number of times the relation can appear in one clause. The + in front of a string specifies that it is a substitutable variable, whilst the # says that it is a constant of particular type (se listing below).

Progol Type Declaration

The above listing shows the type declaration section, and specifies what variables and constant that are in use in the data set.

User session in Progol format

The above listing shows one session with 3 visited URLs.

4 Results

4.1 Acceptance criterion for rules

The acceptance criterion for induced rules is inspired by Bayes [Mor97]

$$\frac{pos}{pos+neg} \geq \frac{concl}{concl+negconcl} \tag{8}$$

where *pos* is the number of tuples for which the premises and the conclusion are true, *neg* is the number of tuples for which the premise is true but the conclusion is not true, *concl* is the number of tuples for which the conclusion is true, regardless of any premise, and *negconl* is the number of tuples for which the conclusion is not true. The first expression corresponds to the conditional probability of the conclusion given the conditions described by the premise. The second expression corresponds to the a priori probability of the conclusion [Fay96b]

Rules for a high number of visits per session

Rules for a medium number of visits per session

Rules for a low number of visits per session

5 Related works

5.1 Web Mining

Web Content Mining

Craven, et. al [Cra98] applies ontologies describing relations and classes of documents in an a inductive learning approach towards learning definitions of page classes and definitions of relations between pages.

Web Usage Mining

Parson [Par98] applies inductive logic programming methods in learning rules for users web-page preferences. This approach can be seen as a combined Web content and usage mining since it both involves analysis of content as well as user behavior. Nasraoui, et. al [Nas99] applies fuzzy clustering of user sessions.

6 Conclusion

This paper presents an approach in how to apply Inductive Logic Programming (ILP) in Web usage mining. The structured log data available on most commonly used Web servers are suitable to be converted into first order logic, there are however issues regarding scalability of ILP systems for large scale web logs. Scalability issues may be handled by combining methods, i.e. clustering and information retrieval methods [Bae99] on most of the data but on selected or random samples one can use the more expressive first-order to discover relational dependencies. Performance of ILP systems are more dependent on the number of features, so high performing preprocessing methods, i.e. rough sets [Øh99] to reduce the number of features might be worth investigating.

6.1 Further work

How can one apply the results of the generated rules in increasing the quality of the web site? Can this be automated i.e. with an integrated expert system in the Web server?

Can such rules be used in the reranking from search engines (in major web sites with several similar pages, it is often determined by the search engine's own ranking methods which of those pages are shown to the user in the result

of the query), the hypothesis is that one might locally on a web site know more about what pages matches the incoming query that is registered in the referer field.

Can induced rules be used in the automatic detection “subtle errors”, e.g. to detect that some parts of a web site not user friendly for some combinations of operating systems and browsers[2]

Since the data used as the foundation in Web mining and adaptive web sites are so valuable, could one get an intermediate market of automatic agent-based trading of (short-term) user profiles? I.e. if the web site detects a potential customer, one could possibly get a better apriori for mood detection if an agent instantianously bought that customers last half ours web logs on different web sites.

All these advanced data analysis may cause abuse of results, legal, ethical and privacy aspects should be considered[Lan00]

Investigate other ILP systems, in [DJM+98] Progol was outperformed by in run time performance by TILDE and ICL, so these will be investigated later.

Acknowledgements

Thanks to my supervisor associate professor Mikhail Matskin for giving valuable feedback. This work has been supported by the Norwegian Research Council through the project ”Electronic Commercial Agents” (ElComAg)

References

[Alf96] Alfheim, E. and Tveit A. Inductive Logic Programming applied in Knowledge Discovery in Databases. Semester project at the Norwegian University of Science and Technology, 1996. 1, 2.3

[Bae99] Baeza-Yates R. and Ribeiro-Neto B. *Modern Information Retrieval*. Addison-Wesley, 1999. 6

[Bry97] Bryant, C.H. Data Mining via ILP: The Application of Progol to a Database of Enantioseparations. In Džeroski, S. and Lavrač, N., editor, *ILP97*, volume 1297 of *LNAI*, pages 85–92. SV, 1997. 2.3

[Bry99] Bryant C.H., Muggleton S., Page C.D. and Sternberg M.J.E. Combining Active Learning with Inductive Logic Programming to close the loop in Machine Learning. In Colton S., editor, *Proceedings of AISB’99 Symposium on AI and Scientific Creativity*, pages 59–64. The Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB), 1999. 1

[Coh97] Cohen W. W. Learning to Classify English Text with ILP Methods. In De Raedt L., editor, *Advances in Inductive Logic Programming*, 1997. 1

[Coo97] Cooley R., Srivastava J. and Mobasher B. Web Mining: Information and Pattern Discovery on the World Wide Web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’97)*, 1997. 1, 2.1

[Cra98] Craven M., Slattery S. and Nigam K. First-order learning for Web Mining. In *ECML-98*, 1998. 5.1

[DJM+98] S. Džeroski, N. Jacobs, M. Molina, C. Moure, S. Muggleton, and W.” Van Laer. ”detecting traffic problems with ilp”. In D.” ”Page, editor, *”ILP98”*, volume ”1446” of *”LNAI”*, pages ”281–290”. ”SV”, ”1998”. 2

[2]This problem has occured in the plethora of WAP emulators and phones

[Dze96] Dzeroski S. Inductive Logic Programming and Knowledge Discovery in Databases. In Fayyad U.M., Piatetsky-Shapiro G., Smyth P. and Uthurusamy R., editor, *Advances in Knowledge Discovery and Data Mining*, pages 117–152. MIT Press, 1996. 2.2

[Fay96a] Fayyad U.M., Piatetsky-Shapiro G. and Smyth P. From Data Mining to Knowledge Discovery. In Fayyad U.M., Piatetsky-Shapiro G., Smyth P. and Uthurusamy R., editor, *Advances in Knowledge Discovery and Data Mining*, pages 1–34. MIT Press, 1996. 2.1

[Fay96b] Fayyad U.M., Piatetsky-Shapiro G., Smyth P. and Uthurusamy R., editor. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996. 4.1

[Lan00] Langford, D. *Internet Ethics*. MacMillan Press Ltd, 2000. 2

[Lav93] Lavrač N. and Džeroski S. *Inductive Logic Programming - Techniques and Applications*. Prentice Hall, 1993. 2.2, 2.2

[Mit97] Mitchell T. *Machine Learning*. McGraw-Hill, 1997. 1

[Mor97] Morik, K. An Inductive Logic Programming Approach. In *Lecture Notes in Computer Science*. Springer Verlag, 1997. 4.1

[Mug95] Muggleton, S. Inverse entailment and Progol. *New Generation Computing*, pages 245–286, 1995. 1, 2.3

[Mug98] S. Muggleton. Inductive logic programming: issues, results and the LLL challenge. In H. Prade, editor, *Proceedings of ECAI98*, page 697. John Wiley, 1998. Abstract of keynote talk. 1

[Nas99] Nasraoui O., Krshnapuram R. and Joshi A. Mining Web Access Logs Using a Fuzzy Relational Clustering Algorithm Based on a Robust Estimator. In *Proceedings of the 8th International World Wide Web Conference*, 1999. 3.1, 5.1

[Øh99] Øhrn A. and Ohno-Machado L. Using Boolean reasoning to anonymize databases. *Artificial Intelligence in Medicine*, pages 235–254, 1999. 6

[Par98] Parson, R. and Muggleton S. An experiment with browsers that learn. *Machine Intelligence*, 1998. 5.1

[Per97] Perkowitz M. and Etzioni O. Adaptive web sites: an AI challenge. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, 1997. 1

[Zaa98] Zaane O. R., Xin M., and Han J. Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs. In *In Proc. Advances in Digital Libraries Conf*, (ADĹ98), Santa Babara, CA, April 1998. 1