# Data Mining for User Modeling
# On-line Proceedings of Workshop held at the International Conference on User Modeling UM2007

**Corfu, Greece, 25 June 2007**

UM 2007
11th international conference on user modeling
Corfu, Greece, 25-29 June, 2007

## Summary

These are the on-line proceedings of the Workshop on Data Mining for User Modeling held at the International Conference on User Modeling (UM2007), on Corfu, Greece, on 25 June 2007. This full-day workshop covered a variety of topics in data mining as it relates to user modeling issues in ubiquitous computing and education, and was composed of three sessions.

- The morning session focused on Educational Data Mining
- At mid-day there was a shared session on data mining for UM for education in ubiquitous contexts. In particular it comprised an invited talk by Gord McCalla.
- The afternoon session focused on Ubiquitous Knowledge Discovery for User Modeling

The workshop brought together researchers and practitioners from a variety of backgrounds, including: user modeling, ubiquitous computing, student modeling, personalization, Web mining, machine learning, intelligent tutoring systems, and assessment.

Due to space limitations, not all papers appear in the printed proceedings. All papers went through the same review process, but some authors volunteered to have their papers appear these on-line proceedings only.

## Workshop Organizers

### Knowledge Discovery for Ubiquitous User Modeling

Bettina Berendt, Institute of Information Systems, Humboldt-Universität zu Berlin

Alexander Kröner, German Research Center for Artificial Intelligence, Saarbrücken

Ernestina Menasalvas, Facultad de Informatica, Universidad Politecnica de Madrid

Stephan Weibelzahl, School of Informatics, National College of Ireland, Dublin

### Educational Data Mining

Ryan S.J.d. Baker, University of Nottingham

Joseph E. Beck, Carnegie Mellon University

## Table of Contents

### Knowledge Discovery for Ubiquitous User Modeling

### Educational Data Mining

## Program Committee

A special thanks to all members of our program committee for the effort taken and the constructive feedback provided.

Ricardo Baeza-Yates, Director of Yahoo! Research Barcelona, Spain and Yahoo! Research Latin America at Santiago, Chile

Jörg Baus, German Research Center for Artificial Intelligence, Saarland University, Germany

Shlomo Berkovsky, University of Haifa, Israel

Christophe Choquet, University of Maine, France

Michel Desmarais, Ecole polytechnique Montreal

Marko Grobelnik, Jozef Stefan Institute, Ljubljana, Slovenia

Dominik Heckman, German Research Center for Artificial Intelligence, Germany

Pilar Herrero, Universidad Politécnica de Madrid, Spain

Anthony Jameson, German Research Center for Artificial Intelligence, Germany

Judy Kay, University of Sydney

Christian Kray, Informatics Research Institute. University of Newcastle, UK

Bruce McLaren,  DFKI

Tanja Mitrovic, University of Canterbury

Dunja Mladenic, Jozef Stefan Institute, Ljubljana, Slovenia

Bamshad Mobasher, DePaul University Chicago, Chicago / IL, USA

Junichiro Mori, University of Tokio, Japan

Katharina Morik, University of  Dortmund,  Germany

Helen Pain, University of Edinburgh

Kaska Porayska-Pomsta, University of London

Thorsten Prante, Fraunhofer IPSI, Germany

Valerie Shute, ETS

Myra Spiliopoulou, University of Magdeburg, Germany

Silvia Viola, Universita Politecnica delle Marche, Ancona, Italy

Titus Winters, DirecTV

Kalina Yacef, University of Sydney

Panayiotis Zaphiris,City University London, UK

# Estimation of User Characteristics using Rule-based Analysis of User Logs

Michal Barla and Mária Bieliková

Institute of Informatics and Software Engineering,
Faculty of Informatics and Information Technologies,
Ilkovičova 3, Bratislava, Slovakia,
{barla,bielik}@fiit.stuba.sk

**Abstract.** Personalization is becoming more important if we want to preserve the effectiveness of work with information, providing larger and larger amount of the content. Systems are becoming adaptive by taking into account characteristics of their users. In this paper we describe our work in the field of automatized user characteristics acquisition based on capturing user behavior and its successive rule-based analysis. We stress on re-usability aspect by introducing rules used for the analysis of logs. Resulting user characteristics are stored in an ontology-based user model.

## 1 Introduction

Many teachers around the world are using information technologies to support the learning process. A lot of tools and frameworks exist, which allow for creation and publication of study materials for students online. There exist specifications like LOM or IMS-LD to support re-usability and interoperability of learning objects. However, learning objects are really re-usable and of great value for students only if they are integrated in such a way that their presentation reflects the needs and skills of the students – individual users, i.e., it is adaptive.

E-learning is an ideal domain for adaptation since every student might prefer different style of learning (e.g., top-down, bottom-up), might have different background and experience in a topic of e-course. If an educational system is aware of these user characteristics (represented explicitly in a user model) it can noticeably improve user's experience with the system and ease the learning process.

In this paper, we present a rule-based approach to analysis of logs of user activity (user modeling based on observing user behavior). We focus on aspect of re-usability and interoperability of the solution. We explicitly defined logs of user activity and devised a generic method of its processing according to a given set of rules. The method produces instances of user characteristics in an ontology-based user model. As a result, it is easy to incorporate a user modeling feature into existing systems and thus enable personalization.

The paper is structured as follows. In section 2 we describe current trends and problems in the field of user characteristics acquisition. Next in section 3 we introduce our user model. Section 4 contains description of proposed rule-based adaptation knowledge representation. In section 5 we describe a process of user characteristics discovery using proposed rules. We evaluate our work and describe future work in section 6. Finally, we give conclusions.

## 2 Related Works

On the top-level, user modeling consists of two stages: data collection and data processing (analysis) [1]. It is important to recognize that the first stage has a substantial impact on possibilities of the second stage.

If we consider automatized approaches to data collection, it is popular to use logs produced by a web server as a basis for the analysis. *Web Usage Mining* [2] is a special branch of data mining techniques applied on the web server logs (clustering, classification, association rule and sequential patterns mining). These techniques are based on a social aspects, where the actual user session is mapped to some patterns of a group of users and as a result they can not be used directly to acquire characteristics of an individual. Still, techniques of *Web Usage Mining* can be used effectively to support students in learning process. A good example can be found in [3].

In the case where the web server log is produced by some specialized tools, it is crucial to transform the log into usable form [4]. Even if it is done, the log lacks semantics of majority of user–system interactions (records are based on low-level HTTP protocol). The user model is then often realized as a simple statistical model expressing whether (and how many times) a user visited some page. Such a model is very system-dependent and is of no use to other systems. What we need is to have a model filled by characteristics rather than statistics.

Because of the mentioned problems, many researchers have proposed separate logging subsystem (often on a client side of the system) which replaces web server log or is used together with it [5]. However, in a majority of current approaches we are missing explicitness of the produced log. Semantics of acquired logs is used implicitly in the part of log analysis which results in tightly coupled modules with limited re-usability. In [6] authors refer to a *Log Ontology* but do not provide more details of it.

A re-usable and interoperable user modeling solutions already exist. For instance the *Duine* toolkit [7] allows any information system to incorporate recommendation services. Disadvantage is that the user model produced by *Duine* is closed, stored in relational database and thus not enough shareable. We found similar problem also in *BGP-MS* system [8].

## 3 User Model

Our user model consists of two parts: logs of user actions and ontology-based part used for actual adaptation. Because collected logs represent huge amount of data, we are taking advantage of maturity of existing relational databases for the storage. Ontological representation of user characteristics allows easy interconnection of several models, sharing and re-usability of constructed user model.

### 3.1 Logs of User Actions

Because we consider logs produced by a web server as not sufficient for the estimation of characteristics of an individual user, we designed and developed a logging sub-system [9] which is responsible for creation of detailed logs of a user activity. Basic requirement is to have *self-contained* records, so we would not need any other additional data to be able to process and interpret them. The semantics of the action should be expressed in the log itself. Therefore we log *event* (as a result of the user action) together with all its *attributes* as well as with a description of current *display state* (description of *items* and their *attributes* which were displayed when the action was performed).

Our next requirement is to have a flexible enough representation of the logs which allows for uniform storing of records of user interactions from several types of user interfaces. Simultaneously, we required a representation which can deal flexible with changes of the adaptive application and its presentation layers.

As a result, we designed a generic data model, whose flexibility was achieved by using a two layer model:

- meta-layer, which prescribes associations between types of entities. We defined types of known events, types of their attributes;
- operating layer, which contains specific run-time values.

### 3.2 Ontology-based model of user characteristics

We use ontology as a mean for representation of user characteristics. It is divided into domain independent and domain dependent parts [10]. The domain independent part defines characteristics like age or sex as well as structure of a characteristic. We combine several ontologies where the domain dependent part is always connected to the appropriate domain model. In this paper we consider representing the ontology by RDF/OWL formalisms.

In our model used for e-learning domain, we currently use two types of characteristics (*CoursePropertyPreference* which informs about a relation of the user to the specific domain properties and *CourseSpecificUserCharacteristic* which informs also about *values* of properties), derived from a common super-class (*gu:UserCharacteristic*), which gives common attributes to all characteristics (see Figure 1). Each characteristic has a time stamp and a source. For the purpose of adaptation we define a user goal, and the characteristics are somehow relevant to the user in achieving this goal. Because our method produces estimations of user characteristics, we store a level of confidence for each characteristic. Confidence informs about quality of the estimation [1].

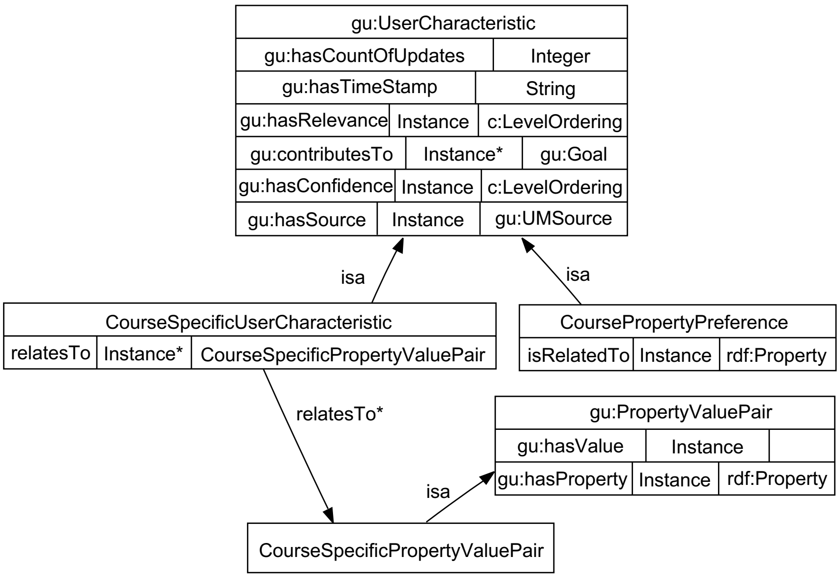


**Fig. 1.** Representation of a user characteristic in used user model.

## 4 Knowledge on User Characteristics Acquisition

Knowledge on user characteristics acquisition is represented by rules. Each rule consists of two parts: a pattern and (at least one) consequence (see Figure 2). Knowledge representation formalism is crucial for devised method of user characteristics acquisition. The rules must be able to store various types of possibilities which can occur during user-system interaction.

### 4.1 Pattern

Pre-defined patterns detected in the user activity log form the base of data analysis. A pattern is on the top-level defined as a sequence of event types and other sub-sequences (see Figure 2). A pattern is detected when an occurrence of the top-level sequence is found. Finding occurrence of the sequence means that we are able to map prescribed events to specific events found in the log of the user activity.

**Sequence.** A sequence can be of two different types:

- *AllRequired* - basic sequence type which is detected in the log of user activity if we detect occurrence of all its events and subsequences (equivalent to logical operation AND);

[1] User models constructed using the same approach can be found at `nazou.fiit.stuba.sk` and `mapekus.fiit.stuba.sk`

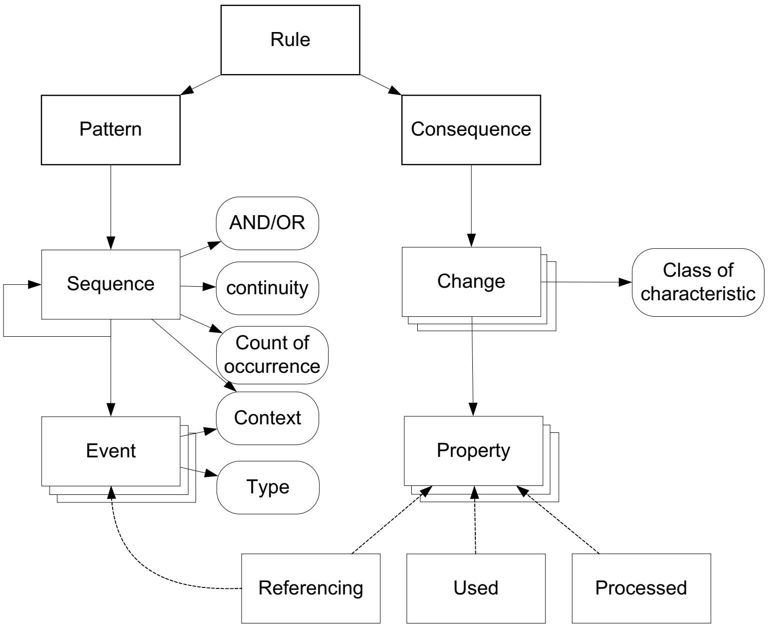


**Fig. 2.** Structure of rules used for estimation of characteristics.

- *OneRequired* - to detect this kind of a sequence, it is sufficient to detect occurrence of one of its events or subsequences (equivalent to logical operation OR).

Further, we divide sequences into *continuous* and *discrete*. A continuous sequence demands that all of its events must succeed directly one after another. Events of a discrete sequence can be separated by any count of other events and sequences. A sequence can thus span through multiple user sessions.

We define following attributes of a sequence:

- *Count-of-occurrence* - prescribes the required count of the sequence repetition in a pattern. The execution engine will continue to process the next sequence only if this count was achieved. This attribute can have a special value (negative number) to define a sequence as optional.
- *Context* - optional attribute which defines the restrictions on events being mapped to the current sequence. For example, a context restriction can define types of displayed items attributes which must stay unchanged for all events mapped to the sequence.

**Event.** An event represents an elementary part of a pattern. During the pattern detection, we map the events from log of user activity to events prescribed by the pattern. Each event has its type which corresponds to the known event type from meta-level of our user actions model. Each event can have a weight. Weight can be determined by considering various factors. We can use information such as time to next event to compute the weight or use a predefined one for a specific type of event.

Similarly to the sequence, an event can also have contextual restrictions. The context of an event defines restrictions solely on attributes of the event while the context of a sequence deals with display state.

Each event context condition can be of the following types:

- *SameAsPrevious* a value of some defined event attribute must be the same as in previous event of a sequence;
- *DifferentThanPrevious* a value of some defined event attribute must be different from the one in previous event of a sequence;

– *MinValueOfWeight* - this contextual condition requires a weight of an event to be higher than some defined value. For instance, if an event "show detail" is immediately followed by a "show overview" event, it will be assigned a low weight not fulfilling defined contextual restriction (user did not have time to see the page with detail). Therefore, we will not map this event into the sequence.

Figure 3 illustrates an example of the pattern representing *"result browsing"*. We can consider for example a repository of available e-learning courses on various topics and a user which is interested in some topic and looks for relevant study materials. When the user selects some restrictions on the information space, a set of results which fulfills selected restrictions is returned and the user can browse them to find out more details.

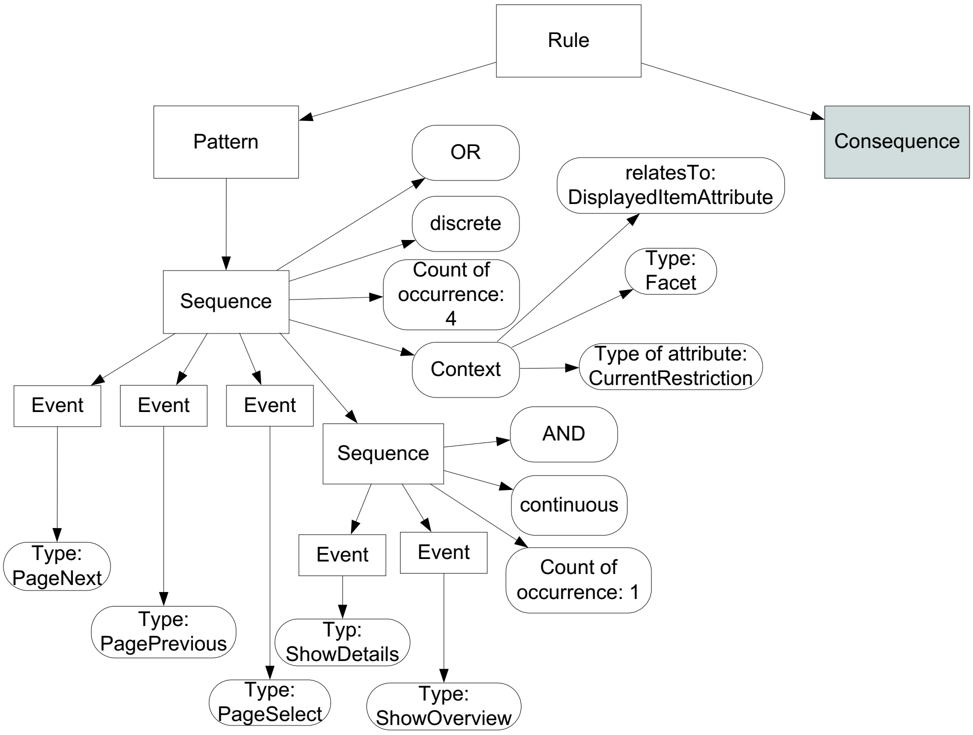


**Fig. 3.** Example of a pattern part of the rule *"results browsing"*.

A pattern is on the top level formed by a discrete sequence of type OneRequired. The sequence has to be found four times in the log of user action for the pattern to become detected. The sequence has a contextual restriction which refers to an attribute of displayed item. All mapped events have to be connected to such a display state, which have for all displayed items of type "facet" constant value of actually chosen restriction. In other words, the user is not changing currently chosen restrictions of the information space and is only browsing in the list of results. Events can be of type *PageNext*, *PagePrevious*, *PageSelect*, *ShowDetails* and *ShowOverview*. Former three types of events represent navigation through individual pages of results while latter two events, joined in a continuous subsequence, represent display of details and navigation back to the list of results.

### 4.2 Consequence

A consequence determines what and how should be changed in the user model in the case when the instance of a pattern is detected. The consequence consists of unlimited count of changes of user characteristics (see Figure 2). Each change has an attribute *class*, which determines the type of the user characteristic being changed (a class where the instance of a characteristic belongs to) and several *property* attributes prescribing changes of the object and literal properties of an instance being changed. Each property is defined by its URI as defined in a T-box of the used ontology.

The change can have three different types of properties:

- *Used property* - a rule defines directly the value which should be used for a property.
- *Processed property* - contains an instruction how to compute value of given property. It is used for numerical data-type properties such as confidence or relevance. Basic information is whether the existing value will be increased or decreased. Processed property defines an increment/decrement for one step and boundaries of interval, where the actual rule can still change the value. Processed property also defines a strategy how to change the characteristic (e.g., progressively or uniform).
- *Referencing property* - refers to an event from the pattern part of the rule. A value of property is actually a value of an attribute of the referenced event.

On the Figure 4 is displayed an example of a consequence part of the rule "result browsing". The consequence changes instances of the *CourseSpecificCharacteristic*.

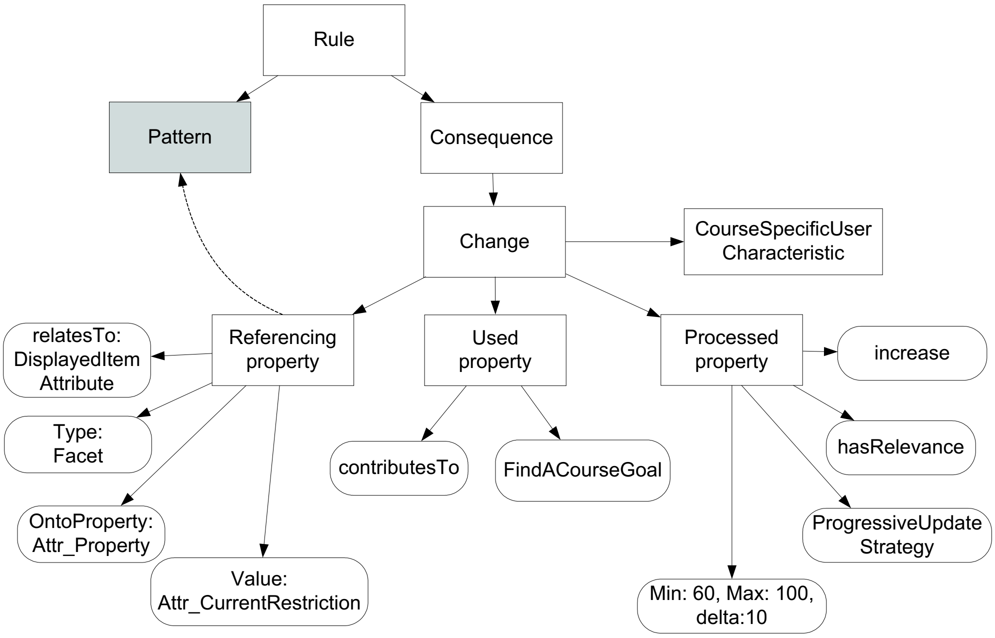

Rule
Pattern
Consequence
Change
CourseSpecificUser Characteristic
relatesTo: DisplayedItem Attribute
Referencing property
Used property
Processed property
increase
Type: Facet
contributesTo
FindACourseGoal
hasRelevance
OntoProperty: Attr_Property
ProgressiveUpdate Strategy
Value: Attr_CurrentRestriction
Min: 60, Max: 100, delta:10

**Fig. 4.** Example of a consequence part of the rule *"results browsing"*.

## 5 User Characteristics Acquisition Process

We have proposed a method for user log analysis based on adaptation knowledge represented by above described rule-based mechanism. The analysis process is depicted on Figure 5. It consists of following steps: pre-processing of the entry, pattern detection, and the user model update. If a pattern of implicit feedback was detected, we include feedback evaluation (acquirement of concept rating) and a comparison of rated concepts into the process.

### 5.1 Data Pre-processing

Thanks to the well defined semantics of data already in the data collection stage and used representation of collected data we do not need such a complex data-preprocessing as we

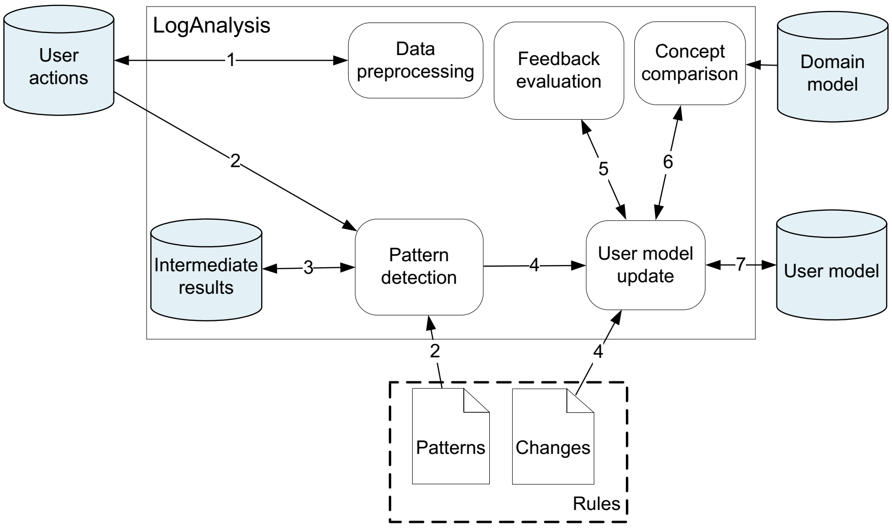


**Fig. 5.** Overview of a data analysis process.

can see in other solutions which work with logs produced by a web server [4]. This stage consists of mainly filtering consecutive events of the same type and context, which is often a result of user's repeated click on the same item (e.g., because of slow response times of the system). We also assign weights as described in 4.1 to individual events at this stage.

### 5.2 Pattern Detection

Pattern detection is a key task in the process of log analysis. Detection works similarly as it is in standard production systems, it maps an event prescribed by the rule to a specific event from the log of user actions. Events are mapped to the instances of the rule for a specific user. Each rule instance holds references to the instances of its sequences to have an evidence of reached count-of-occurrence for each sequence.

The basic idea of the algorithm is explained in the following pseudo-code:

```
DetectPattern(Event):
     find rule candidates for Event;
     for each rule in rule candidates
        find applicable rule instances(rule, event);
        for each rule instance in applicable rule instances
                     apply event on rule instance;

findRuleCandidates(Event):
     for each rule in known rules
         if type of event matches the first event of pattern part of the rule

            add rule to candidate rules;
         else if exists such rule instance of rule belonging to the current user
                     that type of expected event match type of upcoming event
             add rule to candidate rules;
     return candidateRules;

findApplicableRuleInstances(Rule, Event):
   for each ruleInstance of rule belonging to current user
       checkContextOfCurrentSequence(Event);
       checkContinuity(Event);
       checkContextOfEvent(Event);
```

```
      if all checks passed
         add ruleInstance to applicableRuleInstances;
   return applicableRuleInstances;

apply(Event, ruleInstance):
   map Event and expectedEvent of RuleInstance;
   update state of ruleInstance; //nextExpectedEvent, count-of-occurrences
   if Pattern was detected performConsequence part of the ruleInstance;
```

### 5.3 User Model Update

Update of a user model is driven by changes specified in the consequence part of the rule. It performs these steps for each change:

```
UMupdate():
   retrieveInstanceOfUserCharacteristic(); //which is being changed
   for each property in processed properties;
      update value according to given strategy;
   update timestamp;
   update count-of-updates;

retrieveInstanceOfUserCharacteristic():
   check value of all referencing properties;
   check value of all used properties;
   if rule does not allow for change of ‘‘foreign characteristic
      check value of source of characteristic;
   if no instance fulfills these criteria
      create a new instance;
      set all referencing and used properties;
      set source;
   return found or created instance;
```

### 5.4 Feedback evaluation and Concept Comparison

In case that a detected pattern represents an implicit feedback, we compute an evaluation of the concept, which is related to the feedback. This rating is an estimation of the user rating as if the user would rate the concept explicitly by choosing a level from a given scale.

There are several strategies to evaluate implicit feedback according to type of implicit feedback [11] and implicit interest indicators [12]. Transformation of an implicit feedback into numerical value of the user rating separates further processing of the feedback from its source. This allows replacing the implicit feedback by an explicit one with no impact on its processing.

Evaluation of the feedback gives us the user ratings of concepts. Our goal is to estimate user characteristics from these ratings. We aim at finding out which concept attributes and values were the reason of low (or high) ratings. This can be achieved by comparing concepts with different and similar ratings. The basic idea is that if the difference of two concepts in one attribute caused very different ratings we can infer importance of this attribute and its value. The comparison is a complex process, it needs to employ multiple strategies and approaches [13]. Another approach which use feedback evaluation to infer user characteristics is presented in [14].

## 6 Evaluation and Conclusions

In this paper we presented an approach to user characteristics acquisition. Our approach is based on rule-based analysis of logs of user actions. We consider the ease of incorporation of our user modeling subsystem into existing web-based systems architectures as a substantial advantage. Only requirement is to produce a log of user actions and to prepare a set of rules for log analysis which can be easily done as shown in [15]. We presented an idea of the rule mechanism which allows for creation of simple but also (if needed) more complicated pairs of patterns and consequences. Advantage is that we can use the same mechanism for navigation patterns as well as for patterns of implicit feedback. Output of our method is the ontology-based user model filled by estimation of user characteristics. Thanks to chosen representation these characteristics can be shared among several systems and refined to better reflect reality.

To evaluate the proposed method of user log analysis we created a software tool *LogAnalyzer* which performs rule-based analysis of collected data – logs of user actions. Similarly as with design of the method we were focused on re-usability of the tool by separating it from the rest of the system by well-defined interfaces. It is implemented in Java SE 5.0 which means that it is platform independent.

*LogAnalyzer* uses three types of data sources:

- *logs of user actions* stored in relational database. The tool is separated from the actual implementation of RDBMS by an O/R mapper Hibernate;
- *rules* stored in a file using XML based language;
- *user model* stored as triples in RDF repository. The tool is separated from actual implementation of RDF repository by generic enough interface.

We integrated the tool in portal solutions of two different domains – job offers in a project NAZOU [16] (`nazou.fiit.stuba.sk`) and scientific publications in a project MAPEKUS [17] (`mapekus.fiit.stuba.sk`). Characteristics retrieved by the *LogAnalyzer* tool were used for presentation adaptation. In the first stage of the evaluation process we focused on finding an execution model of the user characteristic acquisition process. We let a test user to use the web application and measured execution time of analysis after each event. On the Figure 6 are depicted average values of execution time from multiple runs of the test with a set of four or six rules. As can be seen, there is a substantial difference in time when only instances of rules for individual users were updated and when also the user model stored in RDF repository was updated (a pattern prescribed by the rule was detected). Because of very time consuming call of RDF repository, we decided for asynchronous model of tool execution instead of a pure online mode.

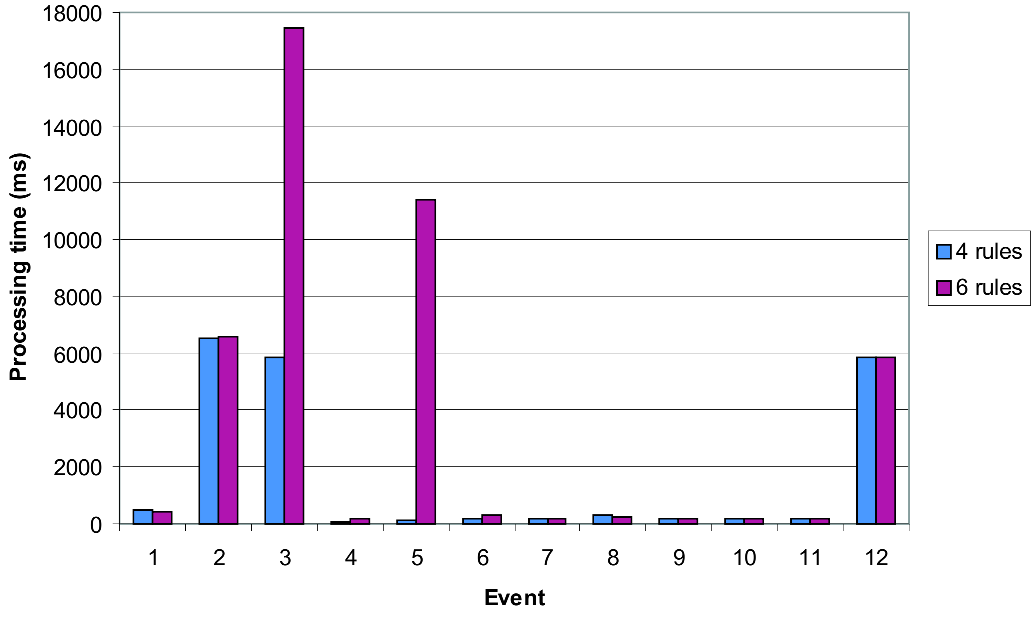


**Fig. 6.** Execution time of LogAnalyzer tool.

In the future work we plan to interconnect our user model with other top-level ontology-based user models [18]. Moreover we work on definition additional adaptation rules based on general heuristics related to navigation and specific heuristics related to an application domain and the evaluation of their impact for user characteristics acquisition.

**Acknowledgment.** This work was partially supported by the Slovak Research and Development Agency under the contract No. APVT-20-007104 and the State programme of research and development under the contract No. 1025/04.

## References

1. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. User Model. User-Adapt. Interact. **6**(2-3) (1996) 87–129
2. Pierrakos, D., et al.: Web Usage Mining as a Tool for Personalization: A Survey. User Modeling and User-Adapted Interaction **13**(4) (2003) 311–372
3. Krištofič, A., Bieliková, M.: Improving adaptation in web-based educational hypermedia by means of knowledge discovery. In Reich, S., Tzagarakis, M., eds.: Hypertext 2005, Salzburg, Austria (2005) 184–192
4. Chen, Z., Fu, A., Tong, F.: Optimal Algorithms for Finding User Access Sessions from Very Large Web Logs. World Wide Web **6**(3) (2003) 259–279
5. Kay, J., Lum, A.: Creating User Models from Web Logs. In: Intelligent User Interfaces Workshop: Behavior-Based User Interface Customization, Funchal, Madeira, Portugal (2004) 17–20
6. Razmerita, L., Angehrn, A., Maedche, A.: Ontology-based user modeling for knowledge management systems. In Brusilovsky, P., Corbett, A., Rosis, F., eds.: User Modeling 2003. LNCS 2702, Johnstown, PA, USA, Springer (2003) 213–217
7. Setten, M.: Supporting People In Finding Information: Hybrid Recommender Systems and Goal-Based Structuring. PhD thesis, Telematica Instituut (2005)
8. Kobsa, A., Pohl, W.: The User Modeling Shell System BGP-MS. User Model. User-Adapt. Interact. **4**(2) (1995) 59–106
9. Barla, M., Tvarožek, M.: Automatic Acquisition of Comprehensive Semantic User Activity Log. In Návrat, P. et al., ed.: Tools For Acquisition, Organisation and Presenting of Information and Knowledge, Bystrá dolina, Slovakia, STU (2006) 169–174
10. Andrejko, A., Barla, M., Bieliková, M.: Ontology-based User Modeling for Web-based Information Systems. In: Information Systems Development (ISD 2006), Budapest, Hungary, Springer (2006)
11. Oard, D., Kim, J.: Implicit Feedback for Recommender Systems. In: AAAI Workshop on Recommender Systems, July 1998., Madison, Wisconsin, USA (1998)
12. Claypool, M., et al.: Implicit Interest Indicators. In: Intelligent User Interfaces, Santa Fe, New Mexico, USA (2001) 33–40
13. Andrejko, A., Barla, M., Tvarožek, M.: Comparing Ontological Concepts to Evaulate Similarity. In Návrat, P. et al., ed.: Tools For Acquisition, Organisation and Presenting of Information and Knowledge, Bystrá dolina, Slovakia, STU (2006) 71–78
14. Gurský, P., et al.: UPRE: User Preference Based Search System. In: Web Intelligence 2006 (WI'06), ACM (2006) 841–844
15. Tvarožek, M., Barla, M., Bieliková, M.: Personalized Presentation in Web-Based Information Systems. In J, van Leeuwen, et al., ed.: SOFSEM 2007. LNCS 4362, Harrachov, Czech Republic, Springer (2007) 796–807
16. Návrat, P. et al.: Acquiring, Organising and Presenting Information and Knowledge in an Environment of Heterogenous Information Sources. In Návrat, P. et al., ed.: Tools For Acquisition, Organisation and Presenting of Information and Knowledge, Bystrá dolina, Slovakia, STU (2006) 1–12
17. Bieliková, M., Návrat, P.: Modeling and acquisition, processing and employing knowledge about user activities in the Internet hyperspace. In Mikulecký, P., Dvorský, J., Krátký, M., eds.: Znalosti 2007, VSB, Ostrava, Czech Republic (2007) 368–371 (in Slovak).
18. Heckmann, D., et al.: GUMO – The General User Model Ontology. In Ardissono, L., Brna, P., Mitrovic, A., eds.: User Modeling 2005. LNCS 3538, Edinburgh, Scotland, UK, Springer (2005) 428–432

# Context, (e)Learning, and Knowledge Discovery for Web User Modelling: Common Research Themes and Challenges

Bettina Berendt

Institute of Information Systems, Humboldt University Berlin,
http://www.wiwi.hu-berlin.de/~berendt

**Abstract.** "Context" has been a popular topic in recent work on interactive systems, in user modelling, knowledge discovery, and ubiquitous computing. It is commonly acknowledged that understanding context is vital for effectively interpreting and supporting users. Several contributions to this workshop explore the use of context for better understanding and/or supporting learning with electronic, networked media. The purpose of this paper is to start a fruitful discussion by showing commonalities and differences between various notions of context, and to outline challenges for future research.[1]

## 1 Introduction

*User – material/environment – interaction – context.* "Context" has been a popular topic in recent work on interactive systems. It is acknowledged that understanding context is important for correctly interpreting user input. This is evident in simple examples like polysemous words as search terms – if the context is known, the right sense can be chosen (e.g., whether "Jaguar" was typed in the context of a search for cars or for animals). In general, context is neither a (stable) property of the user nor one of the material interacted with, but a property associated with the interaction.[2]

*Defining context.* In everyday usage,[3] a piece of information is considered context when it meets at least one of the following two, somewhat orthogonal, criteria: when it is about certain content (features of somebody's cognitive setup as above, environmental conditions like the time of day, the weather, etc.), or when it is information that is transferred non-explicitly (context is "what's between the lines"). In the study of interactive computational systems, the latter is often translated into "data that are collected non-reactively".

Definitions focused on context being about specific content include "any information that can be used to characterise the situation of entities" [20], often with a differentiation between (a) environment (location, time, weather, other properties of the physical environment or computational infrastructure, ...), e.g. [35], and (b) persistent or transient properties of the user (trait and state variables) such as the social environment, preferences, or task, e.g., [57, 35].

That context is non-explicitly transferred information has been emphasized by Lieberman and Selker [35] in a review of work on software agents, sensors, embedded devices, and mathematical and formal studies. They suggest that all non-explicit input to a system can be considered as context and the output of the system itself can be divided into explicit output and changes to the context.

[1] A version of this paper that includes the results of workshop discussions will be available at http://www.wiwi.hu-berlin.de/~berendt/DM.UM07/.

[2] The terms "association", "action", and "activity" are used in the UML sense [45].

[3] Cf. the definition in the Oxford English Dictionary: "parts that precede or follow a passage or word and fix its meaning ...; ambient conditions"

*The relevance of context for UM, UbiComp, KD/DM, and K-DUUM.* [4] Thus, context modelling is recognized as an important part of *user modelling*. In *ubiquitous computing*, it is hoped that more, and more distributed, sensors imply improvements both in the measurement and in the content aspects of context assessment: more things can be measured, and a larger part of the environment can be surveyed. A central idea is to capture contextual information anywhere and anytime, which leads directly to the idea of ubiquitous user modelling, re-use of user models, cross-context user modelling etc., cf. for example [27] and the UbiDeUM workshop at this conference (`http://www.ubideum.org`). *Knowledge discovery / data mining* is interested in measuring context because such measurement produces more features that describe an instance, features which may improve predictive and possibly also diagnostic models. Thus, context is a key concept for ubiquitous knowledge discovery for user modelling.

In the following, we focus on the Web as a space of (at least potentially) ubiquitous availability and usage; in this sense this paper is also about user modelling for ubiquitous knowledge discovery (see the papers and presentations at the UKDU'06 workshop: `http://vasarely.wiwi.hu-berlin.de/UKDU06`).

*Context and eLearning and the relevance for EDM.* [5] Several contributions to this workshop explore the use of context for better understanding and/or supporting learning with electronic, networked media. (The term "eLearning" is used in a broad sense, i.e. not only to denote not only teacher-directed activities in eLearning platforms, but also learning that takes place with other Web resources and learning that takes place out of traditional educational settings.)

Context is currently also heavily researched in neighbouring areas that share many of the issues of context discussed here, but also each have their own specifics. One example is Cognitive Information Retrieval, cf. [29] and the Information Interaction in Context Symposium (`http://www.db.dk/IIiX`). A description of this research and its relationship to our fields of study would constitute another paper; here, this integration is therefore only mentioned as a promising future research direction.

*Motivation and purpose of this paper.* This paper was motivated by the observation that context is a topic currently much discussed both in the (Ubiquitous) Knowledge Discovery / User Modelling and in the Educational Data Mining communities. Thus, this topic appeared to be a good choice for a paper aiming to be an integration point for the present workshop that addresses both communities. The purpose of this paper is to start a fruitful discussion by showing commonalities and differences between various notions of context, and to outline challenges for future research. The purpose is *not* to give an exhaustive state-of-the-art; rather, the focus is on the work of participants of the DM.UM'07 workshop and its precursor, UKDU'06, as examples of wider ranges of literature.

## 2 Context in Web usage mining and eLearning

In this section, I will comment on the formal role of context representations in models ("how" to represent), go on to investigate in which models / model parts they may play this role ("where"), and then give an overview of "what" they are about. An example will illustrate the use of context data for analysis.

### 2.1 Context as data and as metadata

In the following, different approaches to representing context will be outlined. All representations are obviously data. However, context is defined relationally: It is always a context

[4] This paper appears in the "Data Mining for User Modelling (DM.UM)" workshop that consists of "Knowledge Discovery for Ubiquitous User Modelling (K-DUUM) and "Educational Data Mining (EDM)".

[5] see previous footnote

*of* something, generally of a piece of data that is the "proper" core of the situation and/or analysis (a user query or generally user action, a text passage, an object being accessed or manipulated, ...). Thus, context is often modelled as metadata.

eLearning is an application area with a long tradition of metadata. In particular, *learning objects* (LO, e.g., a text, an online course, ...) are annotated with standardized metadata to support search and re-use. Standards like LOM (Learning Object Metadata) collect relevant facets comprising content descriptions, media characteristics, and descriptions of intended educational settings.

LOM describes facets of a learning object's intended context of usage. For example, "educational.context" can be 'school', 'university', or 'training'; and the "educational.typical learning time" helps to identify whether this LO can be used in a given setting of total time. In order to evaluate whether intended and actual usage coincide or not, and in order to obtain a more fine-grained picture of actual usage, it is of course interesting to measure aspects of actual usage.

### 2.2 Context and model parts

Context is a feature of the interaction between user and material. Thus, context representations can form and/or enrich (a) user models, (b) material/environment models, or (c) interaction models.

As pointed out above, UM research generally chooses (a) or (b), cf. for example [57] vs. [30]. Educational metadata research builds on its tradition of (b), enriching the material-centered LOM by context metadata. Finally, research focusing on behavioural observations (inspired, among others, by marketing research and experimental psychology) concentrates on (c).

In the following, I will start by viewing the different approaches to be examined from the (c) perspective (Section 2.3). Subsequently, I will discuss further aspects of context modelling that are also treated in these approaches. Background knowledge (Section 2.4) is often about the materials (this is the focus of the example in that section) or about users (this is mentioned in the interpretation of the example). In this sense, background knowledge often focuses on the (a) or (b) perspective, and on (relatively) persistent properties. Activity structure (Section 2.5) is primarily a feature of the interaction and in this sense transient.[6]

Note, however, that the distinction between "persistent" and "transient" becomes more complex as further investigation layers are considered. Specifically, as the example in Section 2.6 will show, the use of background knowledge becomes a transient feature of the interaction between a new pair of material and user: the end-user interaction data and the analyst, respectively. Therefore, the analysis process implies that different aspects of context representations may "change their role".[7] For example, one usually does not look at background knowledge (or "the user context", "the materials context") for its own sake. Thus, even though these model parts may be said to *constitute* context, they are generally primarily considered because they are the context *of* the interaction being studied, i.e. these model parts are used to extract (previously implicit) context from the activity structure and interaction parameters. Examples include the inference of long-term user properties from location data [41] or from clickstream data [5, 28].

### 2.3 Context: parameters of the (inter)action

This view of context regards a user action that is an interaction with a material (such as clicking a hyperlink, giving an answer in a multiple-choice question, or downloading a document) as an atomic unit of analysis. This action is associated with certain parameters/metadata such as [42, 17]:

[6] The importance of transient properties of the learning setting is stressed by Baker [4], who shows that certain learner actions can be much better predicted from these than from more long-term user properties.

[7] thus the wording "context representations ... form and/or enrich ..." at the beginning of this section

**Date and time** including access time and dwell time
**Action type** such as download, insertion, viewing
**Query terms**
**IP address**
**Operating system, browser** and further technical characteristics of hardware and software
**The application or tool used** including its name, URI, type such as LOR or LMS
**The learner's perspective on the LO** including feedback on the content or knowledge of the content
**The social context of use** including links to the learner model instances attached to LOs previously encountered by the learner.

In our analysis of Web usage behaviour [13], we identified the action of requesting a Web page (by clicking, entering a query, etc.) with the *request for a service*, and we showed that the analysis of such service requests yields different results than the more common analysis of *delivered content* (an investigation of the Web page delivered), in particular results that give more insight on user expectations and intentions. Formalizing these ideas, in [11] we termed such actions *atomic application events* whose metadata are given by session ID, URL query parameters etc.

Service types have also been classified in domain-specific action models, such as Digital-Library usage. For example, the DAFFODIL project [32] distinguishes the following event types: search, navigate, inspect, display, browse, store, annotate, author, help, and communicate. Web-usage analysis projects inspired by marketing models of buying behaviour proceed analogously, e.g., [40, 51, 54].

Najjar, Wolpers and Duval [42] propose to record and analyse *contextualized attention metadata*.[8] These metadata span a wide range of content, service, and environment. *Action* resembles service/event type, *content* has the same meaning as in the models above, *datetime* identifies the temporal aspects (when? how long? in what sequence?) that are also analyzed in Web usage analysis (e.g., [58]). Extensions become possible through the recording of *session*. It corresponds to the "user environment" in Web usage mining: browser type, connection speed, etc. In Web usage mining, these parameters are generally only used for data preprocessing. Another interesting extension is the *application*, which emphasizes that the same material and the same service requests / actions can happen in vastly different tool contexts.

Tanimoto [53] emphasizes that may be difficult to conclude, from a mere clicking event, that there was indeed attention paid to (specific) content of the requested page. He proposes design ideas for pages and hyperlinks to better capture attention. For example, to turn the reading of a page from an eye-moving activity to a combined eyemoving and mousemoving/ clicking activity, two modifications often suffice: a change of the widget that renders the page, and a change of the information structure to make it hierarchical. This encourages the active exploration of the material, but it needs to be used with caution to avoid the loss of context, a slow-down in reading, unwieldy materials, or even repetitive-strain injury.

Brooks and McCalla [17, 36] frame their analysis in terms of the *ecological model*, which "sees metadata as the process of reasoning over observed interactions of users with a learning object for a particular purpose" [17, p. 50]. The usage-context metadata discussed in this work are similar to the ones discussed in the previous paragraph.

In addition, the authors also mention the possibility that different users may add different tags or feedback to the material, based on their usage of it. This is a promising complement to the above notions, which focus on non-reactive methods of gathering context information.[9] The hopes currently attached to Web2.0 applications are that – in contrast to most previous experiences that require explicit user annotations – there are ways to make people enjoy annotating and at the same time collect useful metadata from these annotations. Social

[8] See also the 2006 and 2007 workshops on this topic, `http://sa1.sice.umkc.edu/-cikm2006/workshop.htm`, `http://ariadne.cs.kuleuven.be/cama2007`.

[9] Not all methods are entirely non-reactive: For example, typing a query term requires action from the user. However, this action is an integral part of the user's "real" application behaviour, its procurement of data for context measurement is incidental.

media are recognized as having great promise for learning, e.g. [56]. User tags promise to be a source of context in the sense of (inter)action parameters as discussed in the present section, and also a source of context in the sense of background knowledge. User tags are therefore discussed in more detail in the following section.

### 2.4 Context: background knowledge

Background knowledge is recognized as essential for data mining. It is often needed to discover knowledge at all (e.g., [13]), it is needed to identify interesting and therefore relevant patterns in the multitude of relationships that emerge statistically in the data (e.g., [50]), and it is needed to do more reasoning on patterns (e.g., [44]). In Web settings, the advantage is that the Semantic Web idea provides one of the most advanced proposals for representing such background knowledge. In addition, the Semantic Web has great potential for a distributed and "democratic" way of gathering and combining background knowledge. We have proposed the term *Semantic Web Mining* as a general name for combinations of Semantic Web and Web Mining [52]. Recently, very similar ideas have been discussed under the buzzword "Web 3.0".

To model behaviour in Semantic Web Mining, our framework contains content and service ontologies, which are referenced in material-centered metadata, as ways of interpreting and reasoning on user interactions. A formalization and references to further work are given in [11].

Brooks and McCalla [17] use Semantic Web techniques for representing what an action means. They formulate cognitive-behavioural models in RDF; for example, this allows for an interpretation of what the correct or incorrect solution to a multiple-choice question means. They combine a domain ontology (of the topic domain to be learned) and an educational-objectives ontology. The latter enables them to represent the cognitive proces and the level of knowledge attained. The Semantic Web architecture allows a flexible association of data on usage / interactions with such background models of leaner behaviour and competencies. Thus, this approach offers a way of combining (a) the very behaviour-centered models of Web usage mining, (b) cognitive models, and (c) the flexibility of the Semantic Web.

Material-centered metadata offer another way of integrating background knowledge, for example when keywords are chosen from an ontology, which can then be used for reasoning, making recommendations, etc., e.g., [1].

User tagging promises to be another way of capturing background knowledge. First, it presents a way of capturing multiple perspectives (and *user*, not designer, perspectives), second, it presents a way of gathering different users' choices not only from the same vocabulary/ontology (e.g., [26]), but also from different vocabularies/ontologies. (User tagging can refer to the material, but also to the context, as argued above. Regardless of whether it refers to material, context, or even the user him/herself, user tags are usually tightly bound up with background knowledge.) Collaborative tagging for educational contexts has been proposed for example by Bateman , Brooks, and McCalla [6].

There are proposals for formalizing user tags as metadata and ontologies and using them for machine reasoning, e.g., [24, 49]; however, while this may work in thematically closely circumscribed domains like music, the general semantic status of user tags is very heterogeneous and unclear. In [9], for example, we report evidence of individual differences concerning the semantic status of tags and conclude that for a considerable number of taggers, tags are not metadata, but "just more content".

### 2.5 Context: Activity structure

Last but not least, actions provide context for other actions. In Web usage mining, this occurs in simple forms such as the interpretation of the referrer of a URL request. This metadatum can provide important information about a visitor's intention or expectation (e.g., whether they followed a prescribed link from a course page, or whether they found a material by actively searching with a very detailed search phrase). More commonly, it occurs in the form of analyzing an action as contextualized by the set of requests this action is part of (many recommender algorithms based on the idea of collaborative filtering are based on this assumption) or by a sequence of requests it is part of (see the literature surveys in [38, 11]) or by a higher-order (usually graph) structure of requests, e.g., [12]. Items that were viewed or items that were rated can constitute context [2].

A key question is how much of the temporally surrounding action is relevant. In Web usage mining, a server session (visit) [55] is often taken as the relevant unit (the reason being a combination of measurement constraints and semantic assumptions), although it has been shown that integrating visits to different sites, and more than one visit to the same site, may increase model quality [46]. Within session-defined sequences, patterns are often modelled as first-order Markov structures (effectively claiming that all relevant sequential context lies in the previous click) or as arbitrarily complex sequential patterns. For example, WUM (`http://www.hypknowsys.de`) provides a query language for frequent sequences that allows the explicit specification of contiguity and non-contiguity constraints, and the resulting "navigation patterns" are tree-shaped. Borges and Levene [16] investigated Markov chains of different orders to find out how many previous requests constitute the context needed for predicting the next click. Anand [2] has proposed a cognitively inspired model of short-term memory content (as opposed to long-term memory content) providing the relevant context for recommendations.

There are also ways of bringing background knowledge about activity structure to bear. In [11] we termed such actions *composite application events*: sequences or other structures on atomic application events. Examples include "typical buying behaviours" in eCommerce sites [40, 51, 54]. For reasons of space, this will not be investigated further here.

### 2.6 An example

The implications of these different notions of context for the interpretation of users interacting with (learning or other) materials can best be seen when all notions of context are combined. The following is a simple example taken from the analysis of search behaviour in a medical Web site [7]. It integrates the three aspects of context:

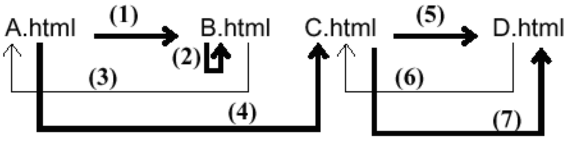


**Fig. 1.** From sequence to graph. The numbers indicate the order of transitions.

**Activity structure** A user visit is modelled as a multigraph. In principle, a visit is represented as a sequence of subsequently visited materials, linked by directed edges representing the transitions. The order of materials/page visits is the order recorded in the Web server log file. However, the repeated request for "the same" material is regarded as an edge pointing to an "earlier" node. Figure 1 shows the construction principle of a graph based on Web pages visited in the order [A.html, B.html, B.html, A.html, C.html, D.html, C.html, D.html] [8].

Different graphs are formed using background knowledge because the graph of visits to URLs is coarsened: Each URL is mapped to a concept (in the ontology) that it represents.

**Parameters of the (inter)action** Base URL and user-specified query parameters are used to interpret each click as the request for a certain content (e.g. information on a specific disease) and for a certain service (e.g., a catalogue-search functionality, a description of an individual disease, etc.). Further distinctions are made for example according to the requested material's media type (a textual description of the disease, pictures of disease symptoms, etc.).

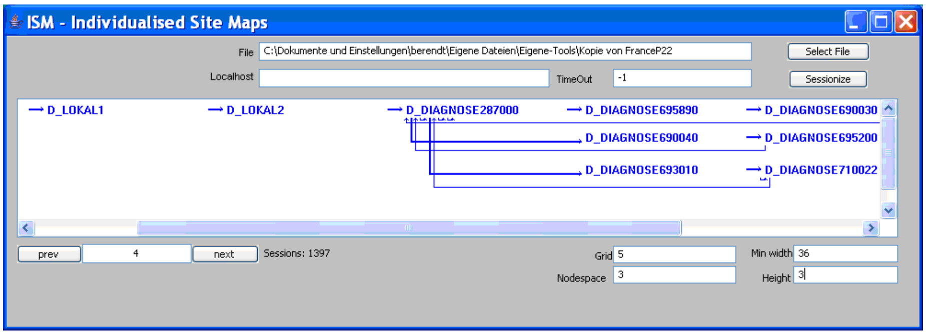
**Background knowledge** The content of the materials is modelled with reference to a domain ontology (here: an internationally standardized classification of diseases) and to a service ontology (here: interaction types in search/information sites). In addition, there can be a media ontology (textual materials, pictures, videos, ...).

Alternative models are used to display the different structures that emerge when one focuses on content, service or media type. In Fig. 2, the focus is on service (search options, here different types of localization search (Lokal...), or information on specific items, here: different diseases / diagnoses Diagnose...); in addition, there is a differentiation by content (identified by diagnosis number). In Fig. 3, the focus is on media type: text (Text), images (Bild), or mixed-media information on differential diagnoses (DD). The graph formed by the displayed session is therefore coarsened in different ways depending on the utilized background-ontology aspect.

This type of analysis allows one to answer questions such as: Which search options are popular (and are there differences between users)? Which content areas were heavily frequented, and how did people navigate between them: did they go back to the search options, or did they use the inter-content links (here: differential diagnosis links)? Did certain content areas become "hubs"for navigation and thus served to organize the domain and/or the presentation of the domain?

On the other hand, media didactics may be more interested in questions like: To what extent did a user / users employ pictorial material or textual material for learning? Were differences found between users with high verbal and users with high visuo-spatial competencies? Did certain textual or pictorial elements become "hubs"? Such questions have been explored in many studies of navigation behaviour (an example is [43]).

For example, the user in Fig.s 2 and 3 used diagnosis 287000 as a content hub, and (s)he accessed primarily pictorial materials, in particular as a first information type from search results pages. Textual materials were only requested once and as a description of a differential diagnosis.


ISM - Individualised Site Maps

File C:\Dokumente und Einstellungen\berendt\Eigene Dateien\Eigene-Tools\Kopie von FranceP22

Localhost

TimeOut -1

4

Sessions: 1397

Grid 5

Min width 36

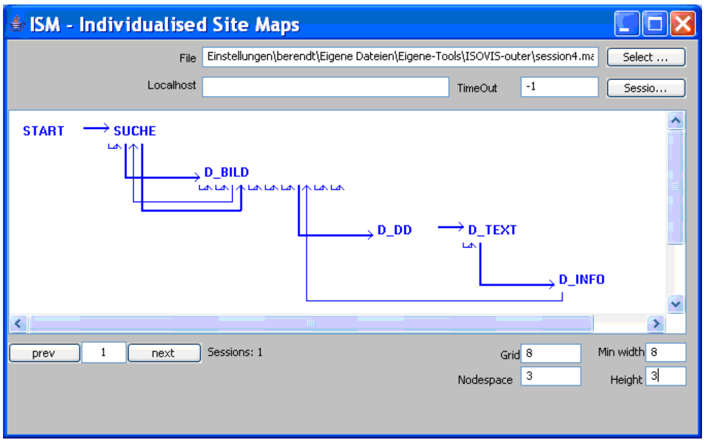
Nodespace 3

Height 3

**Fig. 2.** Interaction graph with a focus on requested service. (The START node that is visible in Fig. 3 was "scrolled out". It is just to the left of the first node.)

This model of behaviour is descriptive and focuses on individual sessions. The graph model has been used in studies of learning behaviour [8, 7]; the background-knowledge and annotation models have been used in studies of the use of educational Web sites [13]. A data-mining extension of it is described in [12].

In [12] and a companion study that involved questionnaires [33], we found that users with low domain expertise (here: patients) and low language competency (here: people using the site in a second language) preferred the localization search, while other users preferred the alphabetical search. In addition, navigation graphs showed that localization search was often followed by the use of differential-diagnosis links, such that patients were able to (possibly incidentally) learn about the domain structure.



File Einstellungen\berendt\Eigene Dateien\Eigene-Tools\ISOVIS-outer\session4.ma

Localhost

TimeOut -1

1

Sessions: 1

Grid 8

Min width 8

Nodespace 3

Height 3

**Fig. 3.** The same interaction graph with a focus on media type

## 3 Outlook: Challenges

In summary, recent years have produced a lot of exciting research on context in eLearning, and further integration of and collaboration between these research efforts promise new insight. Much remains to be done which is not addressed in this work yet. I would like to discuss five major challenges. These are described by the broad research area they concern (the computational study of semantics, data mining, pedagogy, and privacy research) and by a more specific identification of the research topic.

**Semantics: Ontology alignment, standards, other solutions?** Semantic interoperability is a key issue for the processing and analysis of metadata. Is it better to try to establish "global" standards, to have "local" standards that are then matched, or to aim at solutions that do not aim at being formal semantics, but operate in a looser, "Web 2.0", style? Who decides on standards and/or alignment procedures, and what effects does this have?

**Semantics and mining: Using mining for finding semantics** Where do the semantics come from? In principle, semantics can be determined by people, and resources can be annotated manually, for example in Web pages that clearly ask the user to enter each metadatum separately. However, this is unpopular, error-prone, and – on a large scale – simply infeasible. Therefore, the claim has been made that "Web forms must die" [23], and that machine intelligence should be employed to supplant or even replace manual annotation. This raises the question: To what extent can semantics be learned (semi-)automatically? A large number of proposals for ontology learning and instance learning exist (see the survey in [52]), and the idea is being explored for learning educational metadata [19]. Yet, evaluations of the general usefulness of these methods remain scarce. In addition, mining faces the problem of what the relevant input data and structures are, as discussed in the next point.

**Mining: Pattern types** The above considerations have shown that it is an open issue not only what data describe the relevant aspects of context, but also what data structures do that.

For example, are (or: for what purposes are) sets, sequences, or graphs the right structural description of activities? Mobasher et al. [39] examined a part of this question for recommender systems applications. Using predictive quality as an evaluation measure, they showed how the appropriateness of sets vs. sequences depends on the material's hyperlink structure. To extend such ideas to context modelling, one needs to define evaluation criteria and then perform appropriate tests. Another issue concerns the complexity of representation. It is likely that the simple instance-feature matrices commonly used in data mining may not be expressive enough to capture rich models of users and context. Relational data mining [15] is one research direction that should be explored not only in Semantic Web Mining [52], but also in context modelling.

**Mining: Evaluation measures** Evaluation is a key concept for determining quality, and therefore it has a number of different meanings in data mining [10].

First, evaluation measures are needed to assess the quality of a data mining analysis (for example, how well certain context parameters predict learning outcomes). This is typically done using traditional measures like accuracy, ROC, intra- and inter-cluster variance, association-rule support and confidence, etc. The question arises to what extent these domain-independent measures of interestingness are valid indicators of application interestingness (cf. the discussion on interestingness measures, e.g., [50, 37]).

Second, evaluation measures are needed to operationalize the quality of interactive systems that are being evaluated using data mining. A number of evaluation measures have been proposed to evaluate the success of a commercial Web site in turning visitors into customers ("Web metrics" / "Web analytics", e.g., [34]). Indicators exist for measuring usability [21]. For learning, common measures concern the "learning success", but it has been pointed out by educational scientists just how manifold and ill-defined this concept is. For example, Kerres [31, p. 112] points out that learning success comprises the "objective" learning success (e.g., points in a multiple-choice test) at different temporal intervals and in different real-application proximity; the subjectively perceived quality of the learning offers; the emotional reaction and learning motivation; learning behaviour; the degree of satisfaction with the learning behaviour and/or its results; the factual usage, acceptance and thus chances-for-survival in the organisational context.

A third group of evaluation criteria concern the whole process of data collection and mining. This is least well explored in data mining [10], and it goes beyond data mining to incorporate broader issues of software design and project management. Yet, it is one of the most important elements of data mining applications (in learning and elsewhere) that yield process and results which are truly meaningful for the participants.

**Pedagogy and system design: Context data for constructing context metacognition** This paper has been predicated on the assumption that context can in fact be represented, i.e. that there are (meta)data that represent (at least to some extent that makes a system more useful) context.

This assumption has been questioned by Dourish in [22] by referring to practices of formalizing content (particularly in work from the UbiComp community). He argues against the modelling assumptions that context for a certain application is delineable in advance, that it is stable, and that context and activity are two separate aspects of modelling and system design.

However, equating these assumptions with "the representational stance" per se appears to be inadequate. First, Dourish himself later in the paper proposes computational approaches to bringing context into HCI. Second, he identifies as key aspects of context that it is "a feature of interaction", that it is emerging, dynamic, evolving and adapting. I hope to have shown in this paper that many approaches to context see it as a feature of interaction, and that knowledge discovery / machine learning approaches are well-suited to creating systems in which data and knowledge are dynamic, evolving and adapting.

Rather than denounce information processing per se, Dourish appears to emphasize a distinction that has been much discussed in other HCI-dominated areas such as Knowledge Management. In Knowledge Management, the question is phrased as follows: Is knowledge something that can be extracted or externalized and then represented "on a hard disk", or is it that which is constructed in "in people's heads". Proponents of the second position still use representations in their systems, but their key point is that (a) it is a mistake to think that these representations capture all of the relevant knowledge ad that therefore (b) more emphasis should be put on the computer as a *medium* or tool that enables people to communicate, organize their thoughts, etc., and less on the computer as a *machine* that has – by implication – almost equal knowledge-processing power as a person.

Reading Dourish in this way, one can regard his paper as a call to design systems that present context-related data (for example, the system should present its own context, such as available processing resources, to the user) to support the user in constructing the (situation-dependent, emerging, evolving, ...) context "in her head". This can further be read as a call for having computers – through context-related processing and presentation activities – support users' metacognition. Dourish calls this "reflective architectures"; in [7], I have presented a system based on the context captured in Web-usage data and mining results to support metacognition in learning. As is well-known, reflection (or metacognition) is a key activity for successful learning that deserves a prominent role in eLearning.

Further explorations of this constructive nature of context, the use of lessons learned in Knowledge Management, and the building of systems that help people reason about their context and the context of their interaction, are important issues for future research.

**Pedagogy: Participant-centeredness** Research on and system building for eLearning are often characterized by a strange disparity. On the one hand, constructivist beliefs on learning are emphasized; theory and evidence show that without the *active and constructive involvement* of the learner, without the learning process being *meaningful* for the learner, without the consideration of *context* ("situatedness"), no learning can take place. On the other hand, core activities of research and system design effectively take place without learners – specific activities are expected of learners within the framework of the system or setting arranged by the researcher, but the learners are neither (co-)designers of the software nor of the research questions.

Unfortunately, the same problem re-appears at the next level: that of the teachers. I want to argue that both with respect to learners and to teachers, often the basic constructivist principles are violated, and that this may lie behind a frequently observed lack of enthusiasm of teachers for taking part in eLearning studies.[10]

It would be preposterous to attempt, in the space of this article outlook, a thorough analysis (let alone a solution) of the problems of educational systems. In the following, therefore, I can but summarize personal conversations I had with highly motivated but frustrated teachers, and report them as necessarily subjective, and intentionally polemic, observations. (An additional caveat is that specifics of the German situation may have influenced these impressions.)

Teachers (at least school teachers[11]) see themselves as experts on teaching and learning, severely and increasingly hampered in the exercise of their professional skill and ethos by societal, political, financial, and bureaucratic conditions. This is *their* relevant context. They also experience disrespect for their expert judgment, and they experience the (many) evaluations they are subjected to as superficial, summative, and not helpful. Thus, the perception of "real" problems in the learning arrangements and stymied self-efficacy combine into an experience of meaningless settings of professional action and development.

Alas, any "natural-science" oriented empirical research by design tends to weaken the active involvement of the (learner or teacher) participants in the setup of the study. In principle, the exploratory nature of data mining may actually help to conduct more open and more formative/helpful evaluation environments [10]. Also, the use of recorded data/metadata to support metacognition (see above) may be conducive to more active and meaningful roles for participants.

Last but not least, it can be hoped that with the increasing emphasis on context modelling, one can progress to modelling that context that teachers and learners regard as their relevant one.

**Privacy: Against surveillance in learning** Privacy is seen as increasingly endangered in electronic environments, the more so now that more and more data are being recorded and analyzed. And of course, context modelling as outlined above is inherently about

[10] an observation that often baffles researchers: After all, isn't learning the job (and, by implication, the hobby) of a teacher?

[11] The situation is, in some respects, different for university teachers. A big problem here is that efforts to improve teaching may be perceived as meaningless if they do not help a career but impede it, in the sense that time spent on teaching is time not spent on research, funding acquisition, etc.

recording and analyzing more data. Regardless of where one stands in the current debate on privacy in general, it is a particular challenge for learning. As Schulmeister [48] remarked already in 2001: The anonymity of electronic environments has many advantages, especially for failure-oriented learners, and the recording of behavioral data may make fearless interaction impossible.

While in many electronic-transaction domains, legal/societal arguments can be constructed that condone surveillance, it is difficult to do this in the learning domain.[12] Therefore, learning research should embrace current developments in privacy-protecting techniques and technology. This comprises techniques for privacy-preserving data mining (for example, users taking part in collaborative-filtering-based recommendation systems without disclosing private information, cf. [18]), it should take into account that users evaluate different data obfuscation methods differently [14], and it should also go beyond that and take into account that privacy is more than data protection [25, 47].

### Acknowledgements

I thank five anonymous reviewers for their helpful comments.

## References

1. S. S. Anand, P. Kearney, and M. Shapcott. Generating semantically enriched user profiles for web personalization. *ACM Transactions on Internet Technology*, 7(4), 2007 (to appear).
2. Sarab Anand. Putting the user in context. In *Proc. of the ECML/PKDD 2006 Workshop on Ubiquitous Knowledge Discovery for users (UKDU'06). Berlin*, 2006.
3. Ryan Baker, Joseph Beck, Bettina Berendt, Ernestina Menasalvas, Alexander Kröner, and Stephan Weibelzahl, editors. *Proceedings of the Workshop on Data Mining for User Modelling at UM 2007*, 2007. `http://vasarely.wiwi.hu-berlin.de/DM.UM07/Proceedings/DM.UM07-proceedings.pdf`.
4. Ryan S.J.d. Baker. Is gaming the system state-or-trait? educational data mining through the multi-contextual application of a validated behavioral model. In *[3]*, 2007.
5. Michal Barla and Maria Bielikova. Estimation of user characteristics using rule-based analysis of user logs. In *[3]*, 2007.
6. Scott Bateman, Christopher Brooks, and Gord McCalla. Collaborative tagging approaches for ontological metadata in adaptive e-learning systems. In *Proceedings of the International Workshop on Applications of Semantic Web technologies for E-Learning AH'06*, 2006. `http://www.win.tue.nl/SW-EL/2006/camera-ready/02-bateman_brooks_mccalla_SWEL2006_final.pdf`.
7. B. Berendt. Lernwege und Metakognition. In Brigitte Berendt, H.-P. Voss, and J. Wildt, editors, *Neues Handbuch Hochschullehre*, pages 1–34. Raabe Fachverlag für Wissenschaftsinformation, 2006.
8. B. Berendt and E. Brenstein. Visualizing individual differences in web navigation: Stratdyn, a tool for analyzing navigation patterns. *Behavior Research Methods, Instruments, & Computers*, 33:243–257, 2001.
9. B. Berendt and Ch. Hanser. Tags are not metadata, but "just more content" – to some people. In *Proc. of the International Conference on Weblogs and Social Media (ICWSM)*, 2007. `http://www.icwsm.org/papers/paper12.html`.
10. B. Berendt, M. Spiliopoulou, and E. Menasalvas. Evaluation in web mining, 2004. Tutorial at the 15th European Conference on Machine Learning / 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD04), Pisa, Italy, 20 September 2004. `http://ecmlpkdd.isti.cnr.it/tutorials.html#work11`.
11. B. Berendt, G. Stumme, and A. Hotho. Usage mining for and on the semantic web. In H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha, editors, *Data Mining: Next Generation Challenges and Future Directions*, pages 461–480. AAAI/MIT Press, 2004.

[12] There are exceptions such as the current debate on plagiarism detection software: while teachers see a new method of fraud detection, learners object to being put under "general suspicion". However, the legal issues in this controversy focus less on privacy and more on copyright; besides, the focus of data analysis is more on term papers etc. as an externalized product of learning than on the learning process and its context per se.

12. Bettina Berendt. Using and learning semantics in frequent subgraph mining. In Olfa Nasraoui, Osmar R. Zaïane, Myra Spiliopoulou, Bamshad Mobasher, Brij M. Masand, and Philip S. Yu, editors, *WEBKDD*, volume 4198 of *Lecture Notes in Computer Science*, pages 18–38. Springer, 2005.
13. Bettina Berendt and Myra Spiliopoulou. Analysis of navigation behaviour in web sites integrating multiple information systems. *VLDB J.*, 9(1):56–75, 2000.
14. Shlomo Berkovsky, Nikita Borisov, Yaniv Eytani, Tsvi Kuflik, and Francesco Ricci. Examining users' attitude towards privacy preserving collaborative filtering. In *[3]*, 2007.
15. Hendrik Blockeel, David Jensen, and Stefan Kramer, editors. *Machine Learning: Special issue on multi-relational data mining and statistical relational learning.* 2006. 62 (1–2).
16. José Borges and Mark Levene. Generating dynamic higher-order markov models in web usage mining. In Alípio Jorge, Luís Torgo, Pavel Brazdil, Rui Camacho, and João Gama, editors, *PKDD*, volume 3721 of *Lecture Notes in Computer Science*, pages 34–45. Springer, 2005.
17. C. Brooks and G. McCalla. Towards flexible learning object metadata. *International Journal of Continuing Engineering Education and Lifelong Learning*, 16(1/2), 2006.
18. John F. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR*, pages 238–245. ACM, 2002.
19. Kris Cardinaels, Michael Meire, and Erik Duval. Automating metadata generation: the simple indexing interface. In Allan Ellis and Tatsuya Hagino, editors, *WWW*, pages 548–556. ACM, 2005.
20. Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.
21. Alan Dix, Janet Finlay, Gregory Abowd, and Russell Beale. *Human Computer Interaction.* Prentice Hall Europe, 1998. Cited after http://www.tau-web.de/hci/space/i7.html and http://www.tau-web.de/hci/space/x12.html.
22. Paul Dourish. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1):19–30, 2004.
23. E. Duval. Learning objects on the semantic web, 2004. Keynote speech at I2LOR 2004.
24. Oliver Flasch, Andreas Kaspari, Katharina Morik, and Michael Wurst. Aspect-based tagging for collaborative media organisation. In *Proc. of the ECML/PKDD 2006 Workshop on Ubiquitous Knowledge Discovery for users (UKDU'06). Berlin*, 2006.
25. S.F. Gürses, B. Berendt, and Th. Santen. Multilateral security requirements analysis for preserving privacy in ubiquitous environments. In *Proceedings of the Workshop on Ubiquitous Knowledge Discovery for Users at ECML/PKDD 2006*, pages 51–64, Berlin, September 2006. `http://vasarely.wiwi.hu-berlin.de/UKDU06/Proceedings/UKDU06-proceedings.pdf`.
26. Peter Haase, Jeen Broekstra, Marc Ehrig, Maarten Menken, Peter Mika, Mariusz Olko, Michal Plechawski, Pawel Pyszlak, Björn Schnizler, Ronny Siebes, Steffen Staab, and Christoph Tempich. Bibster - a semantics-based bibliographic peer-to-peer system. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 122–136. Springer, 2004.
27. Dominik Heckmann. Situation modeling and smart context retrieval with semantic web technology and conflict resolution. In Thomas Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *MRC*, volume 3946 of *Lecture Notes in Computer Science*, pages 34–47. Springer, 2005.
28. Roland Hübscher, Sadhana Puntambekar, and Aiquin H. Nye. Domain specific interactive data mining. In *[3]*, 2007.
29. P. Ingwersen and K. Järvelin. *The Turn. Integration of Information Seeking and Retrieval in Context.* Springer, 2005.
30. Anthony Jameson. Modelling both the context and the user. *Personal Ubiquitous Comput.*, 5(1):29–33, 2001.
31. M. Kerres. *Multimediale und telemediale Lernumgebungen: Konzeption und Entwicklung.* 2nd edition, 2001.
32. Claus-Peter Klas, Hanne Albrechtsen, Norbert Fuhr, Preben Hansen, Sarantos Kapidakis, László Kovács, Sascha Kriewel, András Micsik, Christos Papatheodorou, Giannis Tsakonas, and Elin Jacob. A logging scheme for comparative digital library evaluation. In Julio Gonzalo, Costantino Thanos, M. Felisa Verdejo, and Rafael C. Carrasco, editors, *ECDL*, volume 4172 of *Lecture Notes in Computer Science*, pages 267–278. Springer, 2006.
33. A. Kralisch and B. Berendt. Language-sensitive search behaviour and the role of domain knowledge. *New Review of Multimedia and Hypermedia: Special Issue on Minority Language, Multimedia and the Web*, 11(2):221–246, 2005.
34. J. Lee, M. Podlaseck, E. Schonberg, and R. Hoch. Visualization and analysis of clickstream data of online stores for understanding web merchandising. *Data Mining and Knowledge Discovery*, 5(1/2):59–84, 2001.

35. Henry Lieberman and Ted Selker. Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal*, 39(3&4):617–, 2000.
36. G. McCalla. The ecological approach: Using patterns in learner behaviour to inform pedagogical goals, 2007. Invited Talk at [3]. `http://vasarely.wiwi.hu-berlin.de/DM.UM07/mcCalla-abstract.html`.
37. Ken McGarry. A survey of interestingness measures for knowledge discovery. *Knowledge Engineering Rev.*, 20(1):39–61, 2005.
38. B. Mobasher. Data mining for web personalization. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*. Springer, 2006.
39. Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. Using sequential and non-sequential patterns in predictive web usage mining tasks. In *ICDM*, pages 669–672. IEEE Computer Society, 2002.
40. W. Moe. Buying, searching, or browsing: Differentiating between online shoppers using in-store navigational clickstream. *Journal of Consumer Psychology*, 13(1&2), 2002.
41. Junichiro Mori, Dominik Heckmann, Yutaka Matsuo, and Anthony Jameson. Learning ubiquitous user models based on users' location history. In *[3]*, 2007.
42. Jehad Najjar, Martin Wolpers, and Erik Duval. Attention metadata: Collection and management. In *Proc. of WWW 2006*, 2006.
43. J. Oberlander, R. Cox, P. Monaghan, K. Stenning, and R. Tobin. Individual differences in proof structures following multimodal logic teaching. In *Proceedings COGSCI96*, pages 201–206, 1996. `ftp://ftp.cogsci.ed.ac.uk/pub/graphics/cogsci96b.ps`.
44. Daniel Oberle, Bettina Berendt, Andreas Hotho, and Jorge Gonzalez. Conceptual user tracking. In Ernestina Menasalvas Ruiz, Javier Segovia, and Piotr S. Szczepaniak, editors, *AWIC*, volume 2663 of *Lecture Notes in Computer Science*, pages 155–164. Springer, 2003.
45. Object Management Group (OMG). Unified modeling language: Superstructure version 2.0, 2005. `http://www.omg.org/cgi-bin/doc?formal/05-07-04`.
46. Balaji Padmanabhan, Zhiqiang Zheng, and Steven Orla Kimbrough. Personalization from incomplete data: what you don't know can hurt. In *KDD*, pages 154–163, 2001.
47. S. Preibusch, B. Hoser, S. Gürses, and B. Berendt. Ubiquitous social networks – opportunities and challenges for privacy-aware user modelling. In *[3]*, 2007.
48. R. Schulmeister. *Virtuelle Universitäten – Virtuelles Lernen*. Oldenbourg Verlag, 2001.
49. Eric Schwarzkopf, Dominik Heckmann, Dietmar Dengler, and Alexander Krïier. Learning the structure of tag spaces for user modeling. In *[3]*, 2007.
50. A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, 1996.
51. M. Spiliopoulou, C. Pohle, and M. Teltzrow. Modelling and mining web site usage strategies. In *Proceedings of the Multi-Konferenz Wirtschaftsinformatik*, Sep 2002.
52. Gerd Stumme, Andreas Hotho, and Bettina Berendt. Semantic web mining: State of the art and future directions. *J. Web Sem.*, 4(2):124–143, 2006.
53. Steven L. Tanimoto. Improving the prospects for educational data mining. In *[3]*, 2007.
54. M. Teltzrow and B. Berendt. Web-usage-based success metrics for multi-channel businesses. In *Proc. of the WebKDD Workshop on Web Mining and Web Usage Analysis*, pages 17–27, 2003.
55. World Wide Web Committee Web usage characterization activity. W3C working draft: Web characterization terminology & definitions sheet, 1999. `http://www.w3.org/1999/05/WCA-terms/`.
56. R. Vuorikari. Can personal digital knowledge artefacts' management and social networks enhance learning? In *Proc. of ICL 2005*, 2005. `http://elgg.net/riina/files/-1/1444/social_networks_learning_vuorikari_22_9_2005.pdf`.
57. W. Wahlster and A. Kobsa. User models in dialog systems. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 4–34. Springer, Berlin, Heidelberg, 1989.
58. Qiang Yang, Hui Wang, and Wei Zhang. Web-log mining for quantitative temporal-event prediction. *IEEE Computational Intelligence Bulletin*, 1(1), 2002.

# Examining Users' Attitude towards Privacy Preserving Collaborative Filtering

Shlomo Berkovsky[1], Nikita Borisov[2], Yaniv Eytani[2], Tsvi Kuflik[1], Francesco Ricci[3]

[1] University of Haifa, Israel

[2] University of Illinois at Urbana Champaign, USA

[3] Free University of Bozen-Bolzano, Italy

[1]slavax@cs.haifa.ac.il, [1]tsvikak@is.haifa.ac.il, [2]{nikita, yeytani2}@uiuc.edu, [3]fricci@unibz.it

**Abstract.** Privacy hazard to Web-based information services represents an important obstacle to the growth and diffusion of the personalized services. Data obfuscation methods were proposed for enhancing the users' privacy in recommender systems based on collaborative filtering. Data obfuscation can provide statistically measurable privacy gains. However, these are measured using metrics that may not be necessarily intuitively understandable by end user, such as conditional entropy. In fact, it could happen that the users are unaware, misunderstand how their privacy is being preserved or do not feel comfortable with such methods. Thus, these may not reflect in the users' actual personal sense of privacy. In this work we provide an exploratory study to examine correlation between different data obfuscation methods and their effect on the subjective sense of privacy of users. We analyze users' opinion about the impact of data obfuscation on different types of users' rating values and generally on their sense of privacy.

## 1 Introduction

Web users leave identifiable tracks while surfing the Web, and there is a growing awareness of and concern about the misuse of such information [18, 22]. Many eavesdroppers on the Web violate user privacy for their own commercial benefits, and as a result, users concerned about their privacy refrain from using Web applications, just to prevent possible exposure [7]. Personalized information delivery in general, and products recommendation in particular play a major role in the development of the Web [19]. Privacy hazards for personalization systems are exacerbated by the fact that effective personalization requires large amounts of personal data. For example, consider a collaborative filtering (CF) system, a commonly used technology in the E-Commerce recommender systems [19]. In order to generate a recommendation, CF initially creates a neighborhood of users with the highest similarity to the user whose preferences are to be predicted, and it then predicts a rating for a target product (a recommendation) by averaging the ratings given by these similar users to the target item [5]. It has been shown that the accuracy of the recommendations thus generated is correlated with the number of similar users and the degree and reliability of their similarity [11] [10]. The more detailed are the user profiles and the larger their cumulative number, the more reliable will be the recommendations. Hence, there is a trade-off between the accuracy of the provided personalization and the privacy of user data.

According to a recent survey [6], most users will not agree to openly sharing their private information. However, people are not equally protective of every attribute in their data records [20, 6]. A user may not divulge the values of certain attributes at all, may not mind giving true values for others, or may be willing to share private information by giving modified values of certain attributes. Hence, in order to provide a stable dynamic infrastructure while preserving the users' privacy, a previous study [2] suggested obfuscating the user's profiles [9] by substituting part of the real values in the profiles with fake values. This setting allows users to store their personal profile locally and leaves them in control as to what personal information they would like to reveal, and when. Thus, a user requesting, for instance, similar user profiles for generating a CF recommendation, would receive only modified user profiles. From these profiles the requesting user can learn only limited information about the true ratings of individual users. Experiments conducted with various datasets demonstrate that a relatively large part of the user profile can be obfuscated, and only a small subset of users is required to generate a recommendation with acceptable average loss in accuracy of the CF [4].

The described setting relies on the assumption that users will feel that such method does improve their actual sense of privacy and that in turn will result in their willingness to provide more personal information for the recommendation process. Further, prior CF works have highlighted that various types of CF ratings have different importance in CF. Accuracy is most crucial when predicting extreme, i.e., very high or very low, ratings on the items. This is explained by the observation that achieving high accuracy when recommending the best and worst items is most important, while poor performance on average items is acceptable [13]. Users are interested in certain predictions on items they might like or avoidance of items that might dislike, but not in precise predictions on items of which they have an average evaluation [12]. The different role played by ratings with extreme or average values is also relevant for privacy-preserving recommender systems. In fact, some ratings in the user profile are more important than the other ratings, i.e., the amount of private information encapsulated in certain ratings is higher than in other ratings.

We consider privacy enhanced personalization as a set of methods that has the characteristic of not deteriorating the prediction accuracy while at the same time use less personal data. Thus, these will leave whoever may look at the personal rating unsure about their true values. Privacy gains measure such increase in the uncertainty about original ratings found in the data. These can be by estimating by the possibility to reconstruct the distribution of the original data [2] [1]. It was previously shown [9] that data obfuscation methods provide privacy gain during the collaborative filtering process. However, such privacy metrics provide only an ordinal measure allowing comparing, on average, different methods. Further, these metrics are usually statistically oriented and thus end users may not understand how privacy is being preserved or not feel comfortable, in general, with such methods. This implies that such measurable privacy gains might not correlate with a human perception of privacy and may not reflect in the users' personal sense of privacy. Thus, it is important to examine the correlation between measurable privacy gain and its effect on the sense of privacy of users found in the system.

In addition, previous work [6] dealt mainly with the attitudes of users towards different types of items while not differentiating various ratings' values. However, we believe that not all ratings' values within one class of item (e.g., movies etc.) bear the same level of importance. This is because of their relative importance in the collaborative process as motivated before. This is also due to the fact that users intuitively express a more clear preference about an item. Thus, it is important to analyze users' opinion about the impact of privacy preserving methods to different ratings; values of ratings and the users' personal sense of privacy. In this work, we conjecture that users may want to protect ratings having extreme values (referred as extreme ratings) more carefully from being exposed. In order to examine these issues we have conducted an exploratory survey to evaluate users' opinions. Here we present our preliminary results. The main contributions of this work are:

- Assess whether users view extreme rating as being more privacy sensitive ratings (e.g., would less like to publicly share these).
- Examine to what extent users will agree to expose personal data in general, and in particular regarding to rating of different types (e.g., extreme ratings).
- Examine whether users consider different gains of their personal sense of privacy from using different types of data obfuscation policies.
- Examine whether users attitude towards sharing their personal data changes as a result of applying the obfuscation methods.

## 2 Data Obfuscation Policies

To provide personalization, while preserving users' privacy, [2] suggests adding uncertainty to the data by obfuscating parts of the user profiles. This reduces the amount of users' information exposed to the recommendation system, and therefore to possible malicious users getting access to the private data stored by the server. Before transferring personal data to the system, a user is supposed to first modify her user model (products' ratings) using various perturbation techniques. Several data perturbation methods were proposed for privacy preservation of a sensitive data: encryption [14], access-control policies [15], data anonymization [16] and others. In this work, we use the term data obfuscation [17] as a generalization of all approaches that involve perturbing the data for data privacy preservation. In this context, a perturbation technique refers to the artificial modification of some of the user ratings with fake values. The rationale of this approach is that the system, and also any malicious attacker, cannot determine with certainty the exact contents of the user profiles. Although this method changes the user's original data, experiments show that it is possible to obfuscate/perturb relatively large portions of a user's profile, and still generate accurate recommendations over the modified data. The work in [4] developed and evaluates three general policies for obfuscating the ratings in the user profiles:

- *Uniform Random* obfuscation – real ratings in the user profile are substituted by random values chosen uniformly in the range of possible ratings in the dataset.
- *Curved Random* obfuscation – real ratings in the user profile are substituted by random values chosen using a bell-curve distribution with properties similar to the statistical properties of the data in the dataset (e.g., average and standard deviation of the ratings).
- *Default* obfuscation(*x*) – real ratings values in the profile are substituted by a predefined constant value *x*; Where x is highly positive, highly negative, or has a neutral value (median of the range).

Different obfuscation methods provide different mix of privacy gains (e.g., make it harder to reconstruct the original data) and loss of accuracy. For example, the *Default* obfuscation policy uses either extreme rating values or values that are close to the average rating of the dataset. Using extreme values in the obfuscation policy, has a strong negative effect on recommendation accuracy, as it substitutes the true value, which is typically close to the average, with one that is very different from the average. Moreover, these extreme ratings will clearly show some precise polarized user preference. The *Curved Random* policy reflects the actual distribution of the data and is supposed to provide the best accuracy, while preserving user privacy, since it is going to reveal a user with average preferences. Previous experiments [4] show that the obfuscated recommendation results are quite similar for different datasets with different levels of density. For instance, the effect of the random policy is an increase of the MAE (average accuracy of the predictions [21]), compared with the value obtained with no obfuscation. With high percentage of ratings perturbed with the random approach, a MAE value close to that of non-personalized recommendations is obtained. As noted before, metrics that quantify privacy gains for a given obfuscation method may not necessarily correspond with the users' sense of privacy. Hence we aim to examine how these correlate.

## 3 Users' extreme ratings

Prior CF works already highlighted that the importance of various types of CF ratings is different. For example, in [13] the authors argue that CF accuracy is most crucial when predicting extreme, i.e., very high or very low, ratings on the items. Intuitively, this can be explained by the observation that achieving high accuracy of the predictions on the best and worst items is most important, while poor performance on average items is acceptable. Similarly, [12] focused on evaluating CF predictions on extreme ratings, i.e., ratings which are *0.5* above or *0.5* below the average rating in the dataset (the numbers refer to a scale between *0* and *5*). This is based on a similar assumption that most of the time the users are interested in certain predictions on items they might like or denial of items that might dislike, but not in uncertain predictions on items of which they are unsure. This observation is true also in privacy-preserving issues. Some ratings in the user profile are more important than the other ratings, i.e., the amount of private information encapsulated in certain ratings is higher than in other ratings. With respect to this issue, two criteria for the importance of ratings should be distinguished: *(1) Content*: This criterion refers to the very nature of the rated items. Certain items can be considered as sensitive if the users are concerned about disclosing their opinions, i.e., their ratings, on them. For example, such sensitive items are typically related to political, sexual, religious, and health domains; *(2) Rating*: This criterion refers to the values of the ratings given by the user on the items. Clearly, extreme ratings (i.e., strongly positive and negative evaluations) allow faster and more reliable identification of user's real preferences. Hence, disclosure and mining of private and sensitive information about the user is alleviated by presence of extreme ratings in user's profile.

In this work, we both build on the hypothesis of [13] and [12] regarding the importance of the extreme ratings during the personalization process and further correlate it with the users sense of privacy. This means, we conjectured on the importance of a ratings using the rating-based criteria and treat in a special way the ratings, whose values are extremely positive or extremely negative, rather than the ratings given on sensitive items. Hence, we aim to analyze users' opinion about the impact of privacy preserving methods to different types and values of ratings to their sense of privacy. We further would like to verify whether applying the proposed obfuscation policies will increase users' willingness to share such rating during the personalization process.

## 4 Examining users' personal sense of privacy

As mentioned before, measurable privacy gains may not necessarily reflect in the users' personal sense of privacy. In order to examine these issues we are currently conducting a survey to evaluate users' opinions. We defined sensitive items as follows: "*A sensitive rating is a rating you do not want to make public. For example, your ratings related to the political, sexual, religious, and health domains may be considered as sensitive*".

We obtained some preliminary results from 117 users. The rating values where supposed to be on a 1-5 scale where 1 represents disliking an item and 5 represents a highly likable item. Question replies where on a scale of 1-7 where 1 indicates strongly disagreeing and 7 represents strong agreement. Table 1 provides the average rate of agreement/disagreement for each question. Figures 1 and 2 show the distributions for the replies to each of the questions. In the figures the distribution is dived into three categories: 1-2 as disagree, 3-5 as neutral/undecided and 6-7 as agree. The survey contained 15 questions. We selected a subset of 11 questions to examine 4 issues:

First we have examined how different values of products' ratings are considered of different importance by the user within a single type of items (e.g., movies etc.). The question aims to check whether ratings with values that are extremely positive or extremely negative are conceived as more sensitive by users. This in turn implies that future algorithms should treat such ratings values differently by privacy-enhancing techniques to enhance users' personal sense of privacy.

***Hypothesis: users consider extreme rating as being privacy sensitive ratings.***

**Q1:** "All my ratings are equally sensitive for me, regardless of the value (1, 2, 3, 4, 5)."

**Q2:** "My ratings with extremely positive (equal to 5) and extremely negative (equal to 1) values are more sensitive for me than the other ratings (2, 3, 4)."

We observed that answering to Q1 (Figure 1-left), *47.79%* of users disagree that all the values of their ratings are sensitive in the same way. Furthermore, in Q2, about *42.98%* of users strongly agree that ratings with extremely positive or extremely negative values are more sensitive than ratings with moderate values. Our results indicate that users do consider their extreme ratings as more sensitive. Thus, future privacy-enhancing algorithms should treat such ratings values differently to practically enhance users' personal sense of privacy.

The second set of questions examines whether users are willing to expose their ratings to improve predictions for other users. Q4 examines to what extent users are willing to expose their average products' ratings. Q5 is similar to Q4 but examines the issue of exposing extreme ratings.

***Hypothesis: users agree to expose personal data in general, but differentiate between different types of ratings.***

**Q4:** "I agree to make my average (equal to 3) ratings public, if this can improve the accuracy of the suggestions provided by the system."

**Q5:** "I agree to make my extremely positive (equal to 5) and extremely negative (equal to 1) ratings public, if this can improve the accuracy of the suggestions provided by the system."

The results in Figure 1-left show that users are polarized towards exposing their average ratings for the purpose of improving the accuracy of the predictions. In particular, *34.78%* of the users disagree for this, and *30.44%* of them agree. Hence, this contradicts the first part of our hypothesis that the users generally agree to expose their moderate ratings. Conversely, most of the users disagree to expose their extreme ratings: only *22.61%* of users agree to expose them, while

*53.91%* disagree for this. Also the average answers shown in Table 1 validate these conclusions: the average level of agreement for exposure of moderate ratings is *4.148* and for exposure of extreme ratings is *3.191*. Intuitively, these conclusions imply that users consider extreme rating as more sensitive, i.e., as more private information, and agree for a smaller exposure of extreme ratings, validating the second part of our hypothesis.

The third set of questions examines how the users evaluate the different obfuscation policies. We compare the extreme, neutral, random and overall extreme policies which are different variants of the policies described in section 2 (similar to ones defined in [4]). When describing the experimental setting we stated: "*We have designed 5 policies that can preserve your ratings' privacy. These policies aim at substituting some of your ratings with fake ratings*". Where the policies are described as follows:

- *"Positive* – substitutes the actual rating with 5, the highest possible positive rating."
- *"Negative* – substitutes the actual rating with 1, the lowest possible negative rating."
- *"Neutral* – substitutes the actual rating with 3, which is the median between the maximum and minimum possible ratings."
- *"Random* – substitutes the actual rating with a random value in the range of possible ratings (1 to 5)."
- *"Overall* – substitutes the actual rating with a random value distributed similarly to the overall distribution of all the ratings stored by the system."

***Hypothesis: users view different personal sense of privacy gain from of different types of obfuscation policies.***

The policies are respectively represented by questions Q6-Q10. The questions where formulated in the same way: for example, **Q6:** "I believe that the **positive policy** is a good approach for preserving my privacy."

The results show that the users' evaluations on the policies are opposite. The average levels of agreement for *positive* and *negative* obfuscation policies are, respectively, *2.657* and *2.577*. Furthermore, most of the users (*56.48%* for *positive* and *58.56%* for *negative*) disagree that these policies are good privacy-preserving mechanisms. The evaluations of the other three obfuscation policies are slightly better. The average level of agreement for the *neutral* policy is *3.404*, for the *random* policy it is *3.730*, and for the *overall* policy it is *4.009*. Similarly, the percentage of users that these policies are good privacy-preserving mechanisms is lower. For the *neutral* policy it is *36.70%*, for the *random* policy it is *36.94%*, and for the *overall* it is *33.64%.*

We hypothesize that these evaluations of the policies can be described by the effect of the general evaluation of the policies and not by privacy-related evaluation only. As the *positive* and *negative* policies substitute the real ratings with highly dissimilar fake values, they hamper the accuracy of the predictions. Hence, their general evaluations are inferior to the general evaluations of the other three policies, and the bias of the general evaluations can be seen also at privacy-related evaluations.

The forth set of questions aim to measure whether the users opinion has changed in their attitude to exposing ratings when these have been perturbed with some of the above mentioned policies. Q13 examines willingness of users to expose average ratings and Q14 similarly examines the issue regarding extreme ratings.

***Hypothesis: "Users' attitude towards sharing their personal data changes as a result of applying the obfuscation policies."***

**Q13:** "I agree to make public my average (equal to 3) ratings, where part of them is substituted, if this can improve the accuracy of the suggestions provided by the system."

**Q14:** "I agree to make public my extremely positive (equal to 5) and extremely negative (equal to 1) ratings, where part of them is substituted, if this can improve the accuracy of the suggestions provided by the system."

The results clearly validate our hypothesis and show that the users increased their willingness to expose their ratings (of both types) as a result of applying the data obfuscation. The average answer regarding the moderate ratings increased from *4.148* in Q4 to *4.764* in Q13. A similar conclusion is true also for the extreme ratings as the average answer increased from *3.191* in Q5 to *3.694* in Q14. Furthermore, also the distribution of the answers validates our hypothesis. Prior to applying the data obfuscation, *34.78%* of the users agreed to expose their moderate ratings and *22.61%* agreed to expose their extreme ratings. Conversely, after applying it these numbers increased to *49.09%* and *27.78%* respectively.

**Table 1.** Average answers to the questions

| Question | Q1 | Q2 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q13 | Q14 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Average | 3.21 | 4.35 | 4.15 | 3.19 | 2.66 | 2.58 | 3.4 | 3.73 | 4.01 | 4.76 | 3.69 |

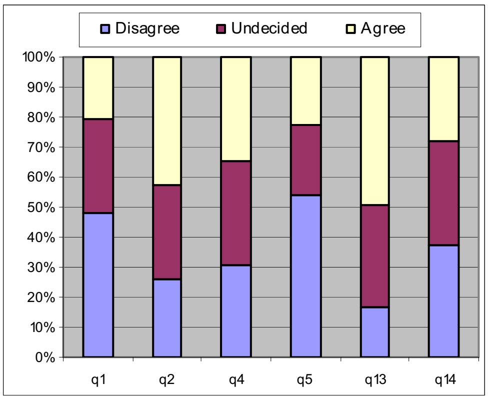


**Fig. 1.** Distribution of answers to Q1, Q2, Q4, Q5, Q13 and Q14

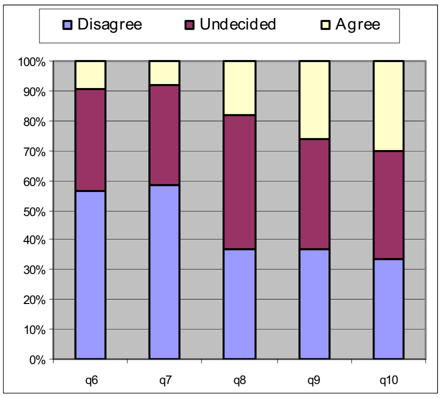


**Fig. 2.** Distribution of answers to Q6, Q7, Q8, Q9 and Q10

## 5. Discussion, conclusions and future work

We consider privacy enhanced personalization as a set of methods that has the characteristic of not deteriorating the prediction accuracy while at the same time to use less personal data. Thus, they leave unsure whoever may look at the personal rating about their true values. Privacy gains measure such uncertainty about the original ratings found in the data. It was previously shown that data obfuscation methods provide measurable privacy gains during the collaborative filtering process. However, such privacy metrics provide only an ordinal measure allowing comparing, on average, different methods. Further, these metrics are usually statistically oriented and thus end users may not understand how privacy is being preserved or generally not feel comfortable with such methods. Hence, such might not correlate with a human perception of privacy and thus may not reflect in the users' personal sense of privacy.

In addition, pervious works discuss the fact that not all ratings within one class of item (e.g., movies etc.) bears the same level of importance. Hence, it is important to analyze users' opinion about the impact of privacy preserving methods to different types of ratings to their sense of privacy. In order to examine these issues we have conducted an exploratory survey to evaluate users' opinions. This work examines the users' attitudes towards the obfuscation methods in collaborative filtering based personalization and how users would consider extreme ratings within a single type of items. Our preliminary results show that users consider extreme ratings as more sensitive and are more reluctant to expose them in the CF process. In addition users' have different attitudes towards the obfuscation methods, but in general all of them encourage users to expose their personal data. Moreover the proposed obfuscation methods seem to higher the willingness of the users to make their ratings available to the system, hence confirm the practical usability of the proposed methods.

Introducing users to the notion of privacy preserving methods when performing the survey lead them to higher willingness to share their personal data. However our current results do not allow us differentiating among the factors that lead to this inclination. Hence, future work should try to examine which are factors plays an important role for motivating users to share more of their personal data. Further, recent efforts in privacy enhanced collaborative filtering have been focusing applying it over P2P and other decentralized settings. Applying CF in such distributed setting bases on an assumption that users will feel that such methods does improve their actual sense of privacy and this in turn will result in their willingness to provide more personal information for the recommendation process. In this work we examined the former part of this assumption. We plan to examine the assumption that leaving users in control of their own profile increase their willingness to provide more information in future work. Other topic we aim to asses are how users intuitively perceives metrics for measuring average content similarity (i.e., conditional entropy) and metrics that measure probable link-ability (i.e., anonymity sets).

**Acknowledgements** We thank Rajesh Kumar, Jodie P. Boyer, Ariel Gorfinkel, Dan Goldwasser and Sadek Jbara for their help during the preparation of the survey.

## References

[1] D. Agrawal, C.C. Aggarwal, "On the Design and Quantification of Privacy Preserving Data Mining Algorithms, Symposium on Principles of Database Systems, 2001.

[2] R Agrawal, R. Srikant: "Privacy-Preserving Data Mining", ACM SIGMOD Int'l Conf. on Management of Data, Dallas, 2000.

[3] S. Berkovsky, Y. Eytani, T. Kuflik, F. Ricci. "Privacy-Enhanced Collaborative Filtering". In Workshop on Privacy-Enhanced Personalization (PEP), Edinburgh, UK, 2005.

[4] S. Berkovsky, Y. Eytani, T. Kuflik, F. Ricci. "Hierarchical Neighborhood Topology for Privacy Enhanced Collaborative Filtering". In Workshop on Privacy-Enhanced Personalization (PEP), Montreal, Canada, 2006.

[5] J. Breese, D. Heckerman, C. Kadie. "Empirical analysis of predictive algorithms for collaborative filtering." In Uncertainty in Artificial Intelligence, Madison, WI, 1998.

[6] L.F.Cranor, J.Reagle, M.S.Ackerman, "Beyond Concern: Understanding Net Users' Attitudes about Online Privacy", Technical report, AT&T Labs-Research, April 1999.

[7] P.Harris, "It is Time for Rules in Wonderland", Businessweek 20, 2000.

[8] Z. Huang, W. Du, B. Chen. "Deriving Private Information from Randomized Data." In ACM SIGMOD Conference, , 2005, Baltimore, Maryland, USA.

[9] H. Polat, W. Du. "SVD-based Collaborative Filtering with Privacy." In ACM Symposium on Applied Computing, Santa Fe, New Mexico,. 2005.

[10] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce*, pages 158-167, 2000.

[11] J.A. Herlocker, J. A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering. In SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA, pages 230-237.

[12] D. M. Pennock, E. Horvitz, S. Lawrence, C. L. Giles, "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach", in proceedings of the International Conference on Uncertainty in Artificial Intelligence, Stanford, 2000.

[13] U. Shardanand, P. Maes, "Social Information Filtering: Algorithms for Automating 'Word of Mouth'", in proceedings of the International Conference on Human Factors in Computing Systems, Denver, 1995.

[14] R. Agrawal, J. Kiernan, R. Srikant, Y. Xu, "Order Preserving Encryption for Numeric Data", in proceedings of the Special Interest Group on Management of Data, Paris, 2004.

[15] R. Sandhu, E. Coyne, H. Feinstein, C. Youman, "Role-Based Access Control Models", in IEEE Computers, vol.29(2), pp.38-47, 1996.

[16] W. Klosgen, "Anonimization Techniques for Knowledge Discovery in Databases", in proceedings of the International Conference on Knowledge and Discovery in Data Mining, Montreal, 1995.

[17] D. Bakken, R. Parameswaran, D. Blough, "Data Obfuscation: Anonymity and Desensitization of Usable Data Sets", in IEEE Security and Privacy, vol.2(6), pp. 34-41, 2004.

[18] S. Brier. "How to Keep your Privacy: Battle Lines Get Clearer." In The New York Times, 13-Jan-97.

[19] J.B. Schafer, J.A. Konstan, J. Riedl, "E-Commerce Recommendation Applications", Journal of Data Mining and Knowledge Discovery, vol. 5 (1/2), pp. 115-152, 2001.

[20] A.F. Westin. "Freebies and Privacy: What Net Users Think", Technical Report, Opinion Research Corporation, 1999.

[21] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, "Evaluating Collaborative Filtering Recommender Systems", in ACM Transactions on Information Systems, vol.22(1), pp.5-53, 2004.

[22] S. Preibusch, B. Hoser, S. Gürses, Bettina Berendt, "Ubiquitous social networks' opportunities and challenges for privacy-aware user modelling", in proceedings of the Workshop on Knowledge Discovery for Ubiquitoues User Modeling, 2007.

# Using Log Data to Detect Weak Hyperlink Descriptions

Vera Hollink, Maarten van Someren, and Bob J. Wielinga

Faculty of Science, University of Amsterdam
Kruislaan 419, 1098 VA Amsterdam, The Netherlands
{vhollink,maarten,wielinga}@science.uva.nl

**Abstract.** Users who visit a web page for the first time select links on the basis of the link anchors and the descriptions of the links that are provided on the page. If these descriptions do not give an accurate impression of the underlying pages, users can not make the right choice and select links that do not lead to their target information. In this paper we present a novel algorithm to automatically detect links with weak descriptions on the basis of usage information stored in the sites' log files. The algorithm distinguishes several types of weak descriptions and provides recommendations for how descriptions of each type can be improved.

## 1 Introduction

Users who browse unfamiliar web sites in search of information choose links on the basis of the link anchors and the text surrounding the links. This only succeeds if these descriptions accurately match the contents of the pages behind the links. If the descriptions are misleading, users often click links that do not lead to target information, which results in unnecessary long navigation times. High quality link descriptions are especially important for users of mobile devices for whom navigation is already slower. Moreover, on these devices choosing a correct link is more difficult as the small screens display less context for the links.

Finding link descriptions that are clear to all users and in all contexts can be an extremely difficult task. Link descriptions are usually designed with a particular user population in mind, but it often happens that other user groups visit the site as well. For these user groups the descriptions can be confusing. Moreover, the perspective of a visitor of a site is in part determined by the context in which the visit is made. For instance, users who visit a site in a work context have different information needs than users who visit the site from home. To create link descriptions that are optimally tailored to a users' needs, properties of the context need to be taken into account.

In this work we present a novel algorithm to automatically identify misleading link descriptions on the basis of the usage of the pages. Links with accurate and unambiguous descriptions are characterized by a usage pattern in which exactly those users who are looking for the pages behind the link, follow the link. Other usage patterns indicate that to some users the descriptions are unclear. The algorithm evaluates the usage patterns of links and classifies their descriptions as strong (clear) or weak (misleading). For weak links it determines the cause of the problem and explains in what way the descriptions need to be changed.

The methods in this work focus primarily on hierarchical link structures. These structures include, for instance, hierarchical menus, web directories and the WAP structures that are considered in [1]. For these link structures it is difficult to find good descriptions, as the descriptions of the intermediate nodes in the hierarchy must not only cover the contents of single pages, but the total contents of a set of pages.

## 2 Related work

Several methods have been developed that analyze log files and link structures and on the basis of the analysis recommend to add or remove certain links (e.g. [2, 3]). These methods evaluate the presence or absence of links, but not the link descriptions. Systems that add new links autonomously need to provide descriptions for the links. These systems usually use hand-made rules to find descriptions, such as using the page title or url (e.g. [4]).

Nakayama *et al.* [3] present an algorithm to detect page pairs that are similar in content, but that are not frequently visited in the same session. They suggest to add a link between these pages if the pages are not yet linked. If a link is already present, they conclude that the page layout must be adapted to improve the visibility of the link. They propose several methods to improve the layout, including changing the link anchor or the text preceding the link. A limitation of their work is that they can only detect weak links between pages that are very similar in content. The usage-based character of our method allows us to find weak links between pages that are related in terms of user relevance, but that have different contents.

Srikant and Yang [5] propose a method to discover the location in a web site where users expect to find certain target pages. They assume that users follow links to the location where they believe a target is located and backtrack when they find out that no link to the target is present at the expected location. Their method computes for each target page the positions where users frequently backtrack and recommends to add links to the target at these positions. This approach is similar to our approach in that both methods aim to determine incorrect navigation paths. However, Srikant and Yang search for the end points of these paths (the backtrack points). They solve the problems at the end points by adding links to the target pages. In contrast, we determine the source of the problem, the point where users deviate from the optimal path. The problem is solved at these points by improving the link descriptions that gave users incorrect expectations about the contents of the underlying pages.

## 3 Method

In this section we formalize the notion of a weak link description and present the algorithm to discover weak descriptions. In addition, we distinguish two types of weak descriptions and discuss how weak descriptions can be improved.

### 3.1 Preprocessing

Usage patterns are extracted from the data collected in server logs. Before the patterns are extracted, the log data is preprocessed. The sessions of individual users are restored with the method described in [6]. All requests coming from the same IP address and the same browser are attributed to one user. When a user is inactive for more than 30 minutes, a new session is started. After the sessions are restored, the pages in the sessions are classified into auxiliary and target pages. A page is a target page for a user if it provides a (partial) answer to his information needs. Auxiliary pages do not contain information that is interesting for the user, but only facilitate browsing. We use the classification method described in [6], which is based on the time that a user spent reading a page. All pages with a reading time longer than a reference length are marked as targets. The other pages are auxiliary pages. As reference length we use the median reading time of the terminal nodes of the hierarchy.

### 3.2 Detection of weak link descriptions

The algorithm for detecting weak link descriptions is given in Fig. 1. Below we describe the algorithm in detail. The following definitions are used:

**Menu** A whole hierarchical menu structure.
**Menu fragment** One node in a menu together with its direct children.
**Link description** The anchor text of a link and the text surrounding the link that gives information about the contents of the link.
**Link content** All content (text, images, etc.) that is located under the link in the hierarchy.
**(In)correct link** A link whose content contains (n)one of the user's target pages.

To evaluate the link descriptions in a menu, the menu is first divided in menu fragments. The descriptions are evaluated fragment by fragment (Fig. 1 (i)). For each fragment we count how many times a user with a target under each child link in the fragment opened each other child link in the fragment. These data are stored in a matrix. We call this matrix

**Algorithm 3.1:** EVALUATE_MENU(*Menu*)

```
evaluations ← ∅
for each fragment in Menu                                              (i)
  do { Create confusion matrix C                                       (ii)
       for each row in C with correct link l
         do Add EVALUATE_LINK(row, l) to evaluations
return (evaluations)

procedure EVALUATE_LINK(row, correct_link)
 if BINOMIAL_TEST(correct_link, row) = too low                         (iii)
   then { confused ← ∅
          for each incorrect_link k in row
            do { if BINOMIAL_TEST(k, incorrect_links) = too high       (iv)
                   then Add k to confused
          if confused ≠ ∅
            then return (evaluation(correct_link, confused_with(confused)))
            else return (evaluation(correct_link, unclear))
   else return (evaluation(correct_link, strong))
```

**Fig. 1.** The link description evaluation algorithm in pseudocode.

a confusion matrix as it shows how often users clicked correct links and how often they confused links with other links (Fig. 1 (ii)). Fig. 2 shows an example of a menu fragment with four child items and a confusion matrix for this fragment.

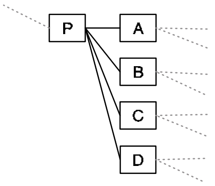
The link descriptions in a menu fragment are evaluated using the corresponding row in the confusion matrix. If a description is accurate, the frequency of the clicks on the correct link is high and the other frequencies are low. In other words, people are able to select the link that leads to a target. Large numbers of clicks on incorrect links indicate a weak link description.

We distinguish two categories of weak descriptions: unclear descriptions and confused descriptions. The first category includes descriptions that are unclear in itself. These descriptions do not match the contents of the link, so that users can not accurately predict whether their target information is located under the link. As a result, many users first try one or more incorrect links before selecting the correct link. In this case, in the matrix not only the frequency of the correct link but all frequencies in the same row are high. An example of a link with an unclear description is item D in Fig. 2: in the fourth row of the confusion matrix all frequencies are relatively high.

The second category of weak links includes links whose contents are in part covered by the description of another link in the same fragment. Users do not know which of the two links they should open and often open the incorrect link. In the matrix both the frequency of the correct link and the frequency of the other link are high, while all other frequencies in the row are low. In the example in Fig. 2, link B is often confused with A.

To determine whether a frequency is too low or too high, we use a statistical test. For each row in the confusion matrix we compare the number of clicks on the correct link to the number of clicks on incorrect links by means of a binomial test (Fig. 1 (iii)). If the proportion of clicks on the correct link is with 99% chance lower than the expected probability $\pi$, the description of the link is marked as weak. By default, the value of $\pi$ is set at the median of the observed probabilities of all correct links in a menu. The value of $\pi$ is always greater than 0, because not all deviations from the optimal path are the result of weak descriptions. In some cases, users may have abandoned their navigation paths, because they changed their information needs during browsing. The category of a weak link is determined by comparing the frequencies of the clicks on the incorrect links (iv). If the binomial tests show that an incorrect link has a significantly higher frequency than average, the correct link is confused with the incorrect link. If none of the incorrect links have a significantly high frequency, the description of the correct link is unclear.

We illustrate the link description evaluation procedure using the example in Fig. 2. First, the description of link A is evaluated. The first row of the confusion matrix shows that in total there were 109 (100+6+2+1) occasions in which users with a target under A made a



|  |  | Clicked link |  |  |  |
|---|---|---|---|---|---|
|  |  | A | B | C | D |
|  | A | **100** | 6 | 2 | 1 |
| Target | B | 40 | **86** | 0 | 1 |
| under | C | 0 | 6 | **90** | 2 |
|  | D | 9 | 10 | 12 | **61** |

**Fig. 2.** Example menu fragment with four child items (left) and a confusion matrix with example frequencies (right). Clicks on correct links are shown in bold.

navigation step within the menu fragment. 100 of these users selected link A. The binomial test shows that this is not significantly lower than expected. Therefore, we conclude that A has a strong description. In 86 of the 127 occasions where the target was under B, B was chosen. The test indicates that this is too low and we conclude that the description of B is weak. Next, we compare the frequencies of the clicks on incorrect links: 40, 0 and 1. The value of 40 of link A appears to be significantly higher that the other two values, which indicates that surprisingly many people with a target under B click link A. Therefore, we say that link B is confused with A. Testing the third row indicates no problems with the description of link C. Link D is chosen on only 61 out of 92 occasions, which is significantly low. If we compare the frequencies of the clicks on the incorrect links, we find that none of them is too high. Consequently, the description of link D is classified as unclear.

Until now we explained how link descriptions can be evaluated for a user population as a whole. As stated in the introduction, in some domains it is important to create different descriptions for different groups of users or for users in different contexts. This can be accomplished by first clustering the log data in a number of user clusters with similar navigation patterns or similar contextual properties. Several methods have been developed for this purpose (e.g. [7]). Once the data is clustered, a link description analysis is performed for each cluster using only the sessions of the users from the cluster. In this way, it is possible to find link descriptions that are clear to one group of users but unclear to others.

### 3.3 Improving weak link descriptions

Once we determined which anchors are insufficient, a web master needs to improve the problematic links. The system provides several guidelines to solve the various problems.

A description can be unclear because some of the contents of the link do not fall in the category that is suggested by the description. This problem can be solved by giving a broader description to the link. Another solution is to maintain the description and group the items that are not covered under a new menu item. A second cause of unclear links is the use of terms in the description that are not known to the users or that have a vague meaning. In this case the description needs to be reformulated in terms used by the user community. One source of such terms are the query terms submitted to a search engine as these terms are typed in by the users themselves.

If a link A is often confused with link B, part of the contents of A are covered by the description of B. This can be solved by assigning a narrower description to B. Alternatively the content items under A that are most frequently confused can be moved or copied to B. If the confusion can not be attributed to some specific content items, A and B can be merged into one large item.

## 4 Evaluation plan and preliminary results

The link evaluation algorithm was applied to the log files and menus of three Dutch web sites. The menus consisted of 85 to 287 links, 9-21% of which were marked by the algorithm as having weak descriptions. On all sites both unclear and confused descriptions were found.

Inspection of the link descriptions that were classified as 'unclear' showed three main causes for weak descriptions. Some descriptions appeared to be too general. For instance, on a site for elders about the prevention of falling accidents one of the menu items is called ‘hints for a safer home’. This description is too general as it covers almost the entire topic of the site. Another cause of weak links is the use of jargon words that are not commonly understood, such as ‘grab pole’ and ‘threshold ramp’. Finally, few descriptions consisted of ambiguous terms where the less common meaning was intended. Links that were confused with other links often had closely related descriptions, such as ‘care providers’ and ‘care institutions’. As these items are very similar, a good solution would be to merge them.

These results are promising as they show that the algorithm is able to identify a number of important shortcomings of link descriptions. However, they do not show the effectiveness of the suggested improvements. To evaluate the practical value of the method, we are planning to conduct a user study. The link evaluation algorithm will be applied to the menu of a web site. On the basis of the suggested improvements the web master of the site adapts the link descriptions and/or the structure of the menu. A number of participants will be asked to search for certain information via the site’s menu. Half of the participants use the original menu and the other half use the improved version. During the search assignments we measure the number of times an incorrect link is followed. Comparison of the results of the two menus allows us to measure the effects of the link evaluation algorithm in a realistic setting.

## 5 Discussion

From usage data we can compute how descriptive links are in their current context. However, we can not evaluate link descriptions per se as descriptiveness is highly context dependent. For instance, a description ‘seal’ is perfectly descriptive among ‘stamp’ and ‘envelope’, but when a link named ’walrus’ is added it becomes unclear. Consequently, when changes are made to the contents or structure of a menu, a new link description analysis needs to be performed for the modified branches.

We are currently extending the classification scheme for weak link descriptions. The extentions allow us to recognize more subtile problems and to give more focused recommendations for improvements. Later, the link evaluation algorithm can be combined with algorithms to estimate navigation time, so that we can predict the effects of modifying the link descriptions and the menu structure on navigation time. These estimations can help a site master to make a choice between various possible solutions for a weak description. Another direction of future research are methods to find alternative link descriptions automatically, for example, by borrowing techniques from keyword extraction or text summarization. In addition, content analysis methods may be used to analyze why descriptions are weak.

The current work addresses only virtual navigation patterns in web contexts, but the presented algorithm can also be applied to physical navigation patterns. If we can track the movements of visitors in, for instance, a physical store, we can evaluate their walking patterns and determine the points where people frequently take wrong turns. This allows us to make recommendations for the improvement of the organization of the store.

## References

1. Smyth, B., Cotter, P.: Intelligent navigation for mobile internet portals. In: IJCAI’03 Workshop on AI Moves to IA: Workshop on Artificial Intelligence, Information Access, and Mobile Computing, Acapulco, Mexico (2003)
2. Wang, Y., Wang, D., Ip, W.: Optimal design of link structure for e-supermarket website. IEEE Transactions: Systems, Man and Cybernetics - Part A **36**(2) (2006) 338–355
3. Nakayama, T., Kato, H., Yamane, Y.: Discovering the gap between web site designers’ expectations and users’ behavior. Computer Networks **33**(1–6) (200) 811–822
4. Schilit, B., Trevor, J., Hilbert, D.M., Koh, T.K.: Web interaction using very small internet devices. Computer **35**(10) (2002) 37–45
5. Srikant, R., Yang, Y.: Mining web logs to improve website organization. In: Proceedings of the 10th International Conference on World Wide Web. (2001) 430–437
6. Cooley, R., Mobasher, B., Srivastava, J.: Data preparation for mining world wide web browsing patterns. Journal of Knowledge and Information Systems **1**(1) (1999) 5–32
7. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Discovery and evaluation of aggregate usage profiles for web personalization. Data Mining and Knowledge Discovery **6** (2002) 61–82

# Learning Ubiquitous User Models based on Users' Location History

Junichiro Mori[1,3], Dominik Heckmann[1] Yutaka Matsuo[2], and Anthony Jameson[1]

[1] German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany,
dominik.heckmann,wahlster@dfki.de,
[2] National Institute of Advanced Industrial Science and Technology, Tokyo, JAPAN,
y.matsuo@aist.go.jp,
[3] The University of Tokyo, Tokyo, JAPAN,
jmori@mi.ci.i.u-tokyo.ac.jp,

**Abstract.** Recent development of location technologies enables us to obtain the location history of users. This paper proposes a new method to infer users' long-term properties from their respective location histories. Counting the instances of sensor detection for every user, we can obtain a *sensor-user matrix*. After generating features from the matrix, a machine learning approach is taken to automatically classify users into different categories for each user property. Inspired by information retrieval research, the problem to infer user properties is reduced to a text categorization problem. We compare weightings of several features and also propose sensor weighting. Our algorithms are evaluated using experimental location data in an office environment. Our algorithm will bootstrap creating ubiquitous user models to enable context-aware information services.

## 1 Introduction

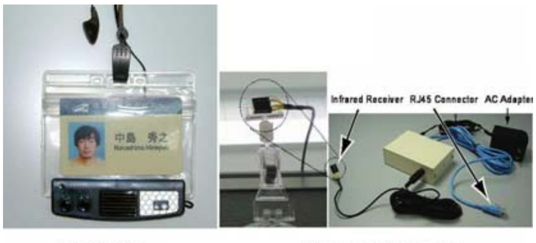
Context-aware computing are gaining increasing interest in the AI and ubiquitous computing communities. To date, numerous approaches have been taken to recognize and model a user's external context, for example one's location, physical environment, and social environment, to provide context-dependent information. Though "context" is a slippery notion [1], it is promising if we can recognize and adapt to aspects of users such as their activities, general interests, and current information needs [2]. Such user models are useful for personalized information services in ubiquitous computing.

Recently, location information has become widely available both in commercial systems and research systems. Devices such as Wi-Fi, Bluetooth, low-cost radio-frequency tags and associated RFID readers, and ultrasound devices all provide location-based information support in various situations and environments. One early and famous project was Active Badge [3]. Since that work, numerous studies of users' activity recognition and location-aware applications have been developed using location and other sensory information in the context of ubiquitous and mobile systems [4–9]. In these studies, user models are sometimes implicitly assumed. For example, being in a laboratory might imply working behavior for laboratory members. While for different types of users such as guests for a campus tour, being in a laboratory implies sightseeing behavior. Therefore, user modeling and behavior detection are mutually complementary: if we have a more precise user model, we can guess more precisely the user behavior, and vice versa. Automatically obtaining a user model will bootstrap activity recognition in a ubiquitous environment to enable context-aware information services.

Toward user modeling for ubiquitous computing, several studies have been done in recent years. Heckmann proposes the concept of *ubiquitous user modeling* [10]. He proposes a general user model and context ontology GUMO and a user model and context markup language *UserML* that lay the foundation for inter-operability using Semantic Web technology. Carmichael et al. proposes a user-modeling representation to model people, places, and things for ubiquitous computing, which supports different spatial and temporal granularity [11].

This paper describes an algorithm to infer a user's long-term properties such as gender, age, profession, and interests from location information. The system automatically learns patterns between users' locations and user properties. Consequently, the system can infer properties: it can automatically produce a user model of a new user coming to the environment. We show that some properties are likely to be inferred and others are difficult to infer. The algorithm is useful in various ubiquitous computing environments to provide user modeling for personalized information services.

We address users' long-term properties, especially among many user-modeling dimensions. Kobsa lists frequently found services of user-modeling, some of which utilize users' long-term characteristics such as knowledge, preference, and abilities [12]. Jameson discusses how different types of information



(a) ID-CoBIT (b) sensor (Tag detector)

**Fig. 1.** ID-CoBIT and sensors

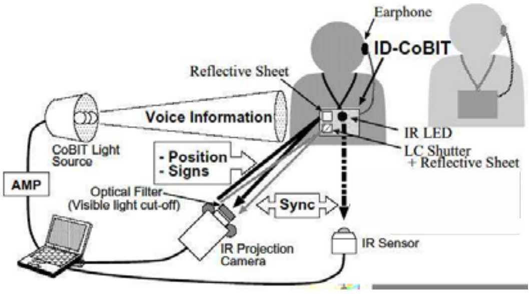


**Fig. 2.** ID-CoBIT system.

about a user, ranging from current context information to the user's long-term properties, can contribute simultaneously to user adaptive mechanisms [13]. In the ontology GUMO, long-term user model dimensions are categorized as demographics such as age group and gender, personality and characteristics, profession and proficiency, or interests such as music or sports. Some are basic and therefore domain independent, whereas others are domain dependent.

Our algorithm is so simple that it is applicable to many types of location data. Inspired by research on information retrieval, we regard the problem of inferring users' properties as text categorization problems. Support vector machine (SVM) is applied to the problem with various feature weighting methods compared in the paper. Our study is evaluated on empirical data obtained through one-week experiments at our research institute. We collected location data of more than 200 users (staffs and guests). The *ID-CoBIT* system consisting of namecard-type infrared transmitters and sensors is installed in the environment to recognize users' location.

The paper is organized as follows. The next section introduces the ID-CoBIT system and describes location data. The proposed algorithm for user modeling from location information is explained in Section 3. Analyses of the results are made in Section 4. We also propose measurements of sensor importance in Section 5. In Section 6 we describe how the proposed algorithm bootstraps creating ubiquitous user models. Related works and discussion are described in Section 7. Finally, we conclude the paper.

## 2 Location information

In our research, location information is obtained using the *CoBIT* system. This section briefly overviews the ID-CoBIT system and describes experiments to collect location data.

### 2.1 ID-CoBIT

The Compact Battery-less Information Terminal (CoBIT ) is a compact information terminal that can operate without batteries because it utilizes energy from the information carrier [14]. The ID-CoBIT is a terminal integrating CoBIT with an infrared (IR) ID tag and a liquid crystal (LC) shutter. Figure 1(a) depicts an ID-CoBIT, which is useful as a namecard holder. The ID detector for ID-CoBIT is a single detector type IR sensor, as shown in Fig. 1(b). The sending cycle of a tag is about 3 s. The effective distance is 3–5 m. Detailed specifications are available in [15].

The ID-CoBIT system provides location-based information support in the environment such as exhibitions, museums, and academic conferences [14]. Users can download information depending on their location and orientation, mainly via voice information. The entire architecture is shown in Fig. 2. Although the ID-CoBIT system has multiple communication channels, in this study we use only the IR LED on an ID-CoBIT and IR sensors in the environment. We specifically address obtaining locations of users without disturbing usual daily behavior.

### 2.2 Experiments

The ID-CoBIT system is installed in an office environment of our research institute. Location data were collected at AIST Tokyo Waterfront Research Center, from February 16, 2004 (Mon) through February 20 (Fri). In all, 94 sensors are installed at the first floor and the fourth floor. On the first floor, we have

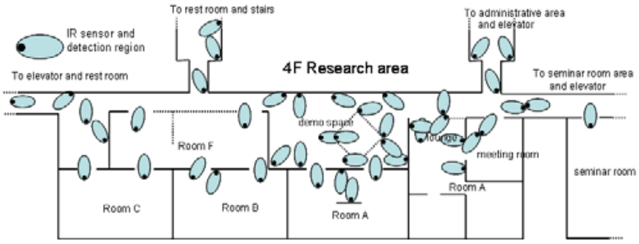

To rest room and stairs
To administrative area and elevator
IR sensor and detection region
4F Research area
To elevator and rest room
To seminar room area and elevator
demo space
Room F
lounge
meeting room
seminar room
Room A
Room C
Room B
Room A

**Fig. 3.** Sensor allocation map for a part of the fourth floor.

**Table 1.** User properties.

| user property | range |
|---|---|
| AGE | `under24, 24-29, 30-34, 35-39, over40` |
| POSITION | `sc*, full-time researcher, part-time researcher : technical-staff, temporary-staff` |
| TEAM | `research-group-A, -B, -C, -D, secretaries, administrators` |
| WORK-FREQUENCY** | `high, middle, low` |
| COFFEE*** | `high, middle, low` |
| SMOKING | `yes, no` |
| ROOM+ | `A, B, C, D, E, F` |
| COMMUTING++ | `station-A, station-B` |

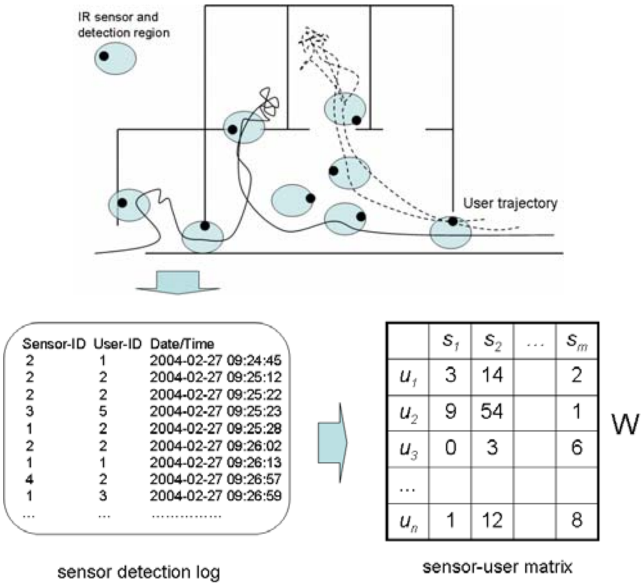
* SC stands for steering committee. ** Because of the free time system of this work environment, working time and commuting frequency depend on the person. *** How often one drinks coffee. + Working room at one's desk. ++ Two train stations on two lines are accessible from this Institute.

entrances, reception areas, lobbies, and lounges. The fourth floor is our main office, consisting of a research area and an administrative area. The sensor allocation map on the fourth floor is shown partly in Fig. 3, which is about 1000 square meters, or about a third of the entire covered area. Every working staff member on these floors was delivered an ID-CoBIT, which they continued to wear during the period.

We also delivered ID-CoBITs to and obtained location information of 170 guests who visited the institute temporarily during the period. After the experiments, we analyzed the location data. The detection instances of all sensors were 24317 times: 20273 times of staff, 4044 times of guests. The number was almost constant each day. On average, a staff member was detected 431.3 times; a guest was detected 23.8 times. Because the location information and user properties of staffs are quantitatively and qualitatively better than those of guests, we use the staff information in this paper.

For obtaining users' long-term properties, we manually surveyed user attributes for all staff such as age, work frequency, room, and whether they smoke or not. The user properties used in our study are shown in Table 1. We chose demographic properties such as AGE and POSITION, domain-dependent properties such as TEAM and WORK-FREQUENCY, and user-specific properties such as COFFEE and SMOKING.

We elaborate these properties considering usefulness in our domain and also versatility in other domains: First, AGE and POSITION are important properties especially in Japan; in the Japanese culture, age and position make large differences in communication such as using respect language and behavior. As it is often inappropriate and impolite to ask a user about the age and position directly, it is useful for the system to infer such properties. TEAM and WORK-FREQUENCY can be seen as users' interests in the research domain. Because team organization is flexible in our institute, they reflect well the reseracher's interest. COFFEE and SMOKING are useful for guests. If the system can recognize that a guest likes coffee or smoking, it can suggest appropriate restaurants or cafes in break time. Because we are often asked "do you like coffee or tea?" or "do you smoke or not?" (in Japan), it indicates the usefulness of the properties in our daily lives. Lastly, ROOM and COMMUTING are for navigation. Knowing the properties, the system can infer in which room a researcher might be (even if he does not wear the CoBIT), or recognize whether he/she goes home or not.



```
Sensor-ID  User-ID  Date/Time
2          1        2004-02-27 09:24:45
2          2        2004-02-27 09:25:12
2          2        2004-02-27 09:25:22
3          5        2004-02-27 09:25:23
1          2        2004-02-27 09:25:28
2          2        2004-02-27 09:26:02
1          1        2004-02-27 09:26:13
4          2        2004-02-27 09:26:57
1          3        2004-02-27 09:26:59
...        ...      ...............
```

|  | $s_1$ | $s_2$ | ... | $s_m$ |
|---|---|---|---|---|
| $u_1$ | 3 | 14 |  | 2 |
| $u_2$ | 9 | 54 |  | 1 |
| $u_3$ | 0 | 3 |  | 6 |
| ... |  |  |  |  |
| $u_n$ | 1 | 12 |  | 8 |

**Fig. 4.** Illustration of sensor detection and the sensor-user matrix.

## 3 Inference of User Properties

In this section, we propose our algorithm to infer user properties based on their respective location histories. We first describe how to reduce the prediction problem of user properties into a text categorization problem. Then, the feature design for machine learning is explained.

### 3.1 Reduction to a Text Categorization Problem

When a sensor detects a user, the *SensorID* and *UserID* are obtained each time a sensor detects a user. Counting the number of detections, we can build a matrix that represents how many times each sensor detects each user. We call it a *sensor-user matrix*. Denoting the number of users as $n$ and the number of sensors as $m$, the sensor-user matrix is an $n \times m$ matrix $W$. We denote $W_{ij}$ as the element of $W$, i.e., the number of detections of user $u_j$ by sensor $s_i$. The illustration of sensor detection to a sensor-user matrix is shown in Fig. 4.

Next, we consider user properties. For example, a user property of whether the user drinks coffee or not (the COFFEE property) can be represented as $\{$*yes*, *no*$\}$ or $\{1,0\}$. Assuming that three users have the values *yes*, *no*, and *yes* for the property, we have the following table as a training set.

|  | $s_1$ | $s_2$ | $s_3$ | $s_4$ | COFFEE |
|---|---|---|---|---|---|
| $u_1$ | 1 | 2 | 2 | 4 | 1 |
| $u_2$ | 1 | 0 | 2 | 0 | 0 |
| $u_3$ | 3 | 2 | 0 | 0 | 1 |

Then, when a new user $u_4$ comes and the detection frequencies are observed, a prediction problem arises. Is the COFFEE property of the user 1 or 0?

|  | $s_1$ | $s_2$ | $s_3$ | $s_4$ | COFFEE |
|---|---|---|---|---|---|
| $u_4$ | 2 | 2 | 0 | 0 | ? |

From the training set, classification can be learned using machine-learning techniques. If we take nearest neighbor method, the most similar one to $u_4$ seems to be $u_3$ (though it depends on the similarity measure). Therefore, the method outputs 1.

This approach is justified using the following example: Let us consider a situation in which sensor $s_2$ is installed in front of a coffee server. Then, frequent detection by $s_2$ means that the user comes frequently to the coffee server, which might imply that the user likes coffee. We cannot know in advance which sensors are important for classification; they might be those in front of a coffee server, the ones in front of a vending machine, or those that are completely unexpected. In any case, classification is learned through sensor detection data and the performance is evaluated by $k$-cross validation or the leave-one-out method, where each part of the training data is used repeatedly as both initial training data and test data.

jThe obtained problem closely resembles a text categorization problem. A document is often represented by a word vector (or a *bag of words*) in which each word in the document is weighted by some word weighting; all structure and linear ordering of words in the document are ignored. The term-document matrix (or a document-by-word matrix [16]) resembles our sensor-user matrix $W$ in that we have $n$ documents and $m$ words in which each user corresponds to a document and each sensor corresponds to a word. In a text categorization task, categories are annotated to each document, which can be considered as user properties in our problem. The classification is learned and used to infer the category based on the word vectors. Therefore, the user-modeling problem from location information is reduced to a (multi-label) text categorization problem under the proper assumptions and simplifications.

Text categorization is typically attained using several classification techniques. We employ support vector machine (SVM) as a learner, which creates a hyperplane that separates the data into two classes with the maximum-margin [17]. The SVMs offer two important advantages for text categorization: term selection is often unnecessary because SVMs tend to be fairly robust to overfitting. In addition, there is a theoretically motivated, "default" choice of parameter setting [18]. These benefits are also provided by our user-modeling problem.

### 3.2 Feature Design

In the context of text categorization, *tf-idf* is frequently used as feature weighting, which encodes the intuition that (i) the more often a word occurs in a document, the more it is representative of its content, and (ii) the more documents in the word occurs in, the less discriminating it is. In our studies, it is rephrased as follows: (i) the more often a sensor detects a user, the more it is representative of the user's characteristics, and (ii) the more users a sensor detects, the less discriminating it is.

The *tf-idf* weighting function tailored to our case is defined as $tfidf(s_i, u_j) = freq(s_i, u_j) \times idf(s_i)$ where $freq(s_i, u_j)$ is the number of detections of users $u_j$ by sensor $s_i$. The $idf(s_j)$ is defined as $idf(s_i) = \log(n/uf(s_i))$ where $uf(s_i)$ is the number of users that sensor $s_i$ detects (corresponding to document frequency). A sensor that detects many users has high $uf(s_i)$ value, and therefore a low $idf(s_i)$ value. In an extreme case, a sensor detecting all $n$ users has a zero $idf$ value as $log(n/n) = 0$.

Aside from *tf-idf* weighting, several ways of feature weighting are possible. We compare typical weighting methods that are often used and compared in information retrieval. The following is a list of feature weighting methods that we use:

- Frequency (number of detections): $w_{ij} = freq(s_i, u_j)$
- Binary: $w_{ij} = \begin{cases} 1 \text{ if } freq(s_i, u_j) > f_{thre} \\ 0 \text{ otherwise} \end{cases}$ where $f_{thre}$ is a threshold. In this paper, we determine $f_{thre} = 1$ through preliminary experiments.
- IDF: $w_{ij} = \begin{cases} idf(s_i) \text{ if } freq(s_i, u_j) > f_{thre} \\ \quad 0 \quad \text{otherwise} \end{cases}$
- TFIDF: $w_{ij} = tfidf(s_i, u_j)$

For the weights to fall in the [0,1] interval and for the vectors to be of equal length, the weight can be normalized by cosine normalization, given as $w_{ij}^{normalized} = w_{ij}/\sqrt{\sum_{i=1}^{m}(w_{ij})^2}$.

Therefore, we have $4 \times 2$ feature weighting methods, which are compared in the next section. We call those: FREQ, BINARY, IDF, TFIDF, N-FREQ, N-BINARY, N-IDF, and N-TFIDF. Although normalized *tf-idf* (N-TFIDF) is known to perform well for text categorization, different results are revealed in our user-modeling problem.

## 4 Evaluation

For each user property shown in Table 1, a categorization problem is generated. More exactly, because SVM is fundamentally applicable to the two-classes problem, a problem is generated for each value of the property. For example, the AGE property can take five values: five classification problems are generated. We make positive and negative classes for each value, say `under24`, i.e., those who are under 24 and those who are not. Thus the obtained classifier will classify people into those who are under 24 and those

**Table 2.** Classification performance depending on various feature weighting

| | F-value(%) | Recall(%) | Precision(%) |
|---|---|---|---|
| FREQ | 44.45 | 73.56 | 37.75 |
| BINARY | 43.92 | 65.83 | 41.62 |
| TFIDF | 44.28 | 71.62 | 37.38 |
| IDF | 44.37 | 68.45 | 45.33 |
| N-FREQ | 54.46 | 68.83 | 49.23 |
| N-BINARY | 40.73 | 63.80 | 40.97 |
| N-IDF | 41.23 | 61.02 | 41.46 |
| N-TFIDF | 53.00 | 65.50 | 47.88 |

who are not. The SVM is used to learn the categorization and the performance is evaluated by leave-one-out. We employ a radius basis function (RBF) kernel, which performs well in our preliminary experiments. [4]

Average performances on all categorization problems are shown in Table 2. For example, if we use FREQ as a feature weighting, the recall is 73.56%, meaning that we can detect 73.56% of persons with a property having a certain value. As a baseline, we investigate the performance of the straightforward classifier that always outputs positive; F-value is 38.2% and precision is 23.93%. Thus our method is much better than the baseline. As for feature weighting, N-FREQ has the highest F-value, and N-TFIDF is the second best. Normalization seems to function effectively for either feature weighting: it might alleviate the difference of detection frequency among users that was caused by the difference on the working time or individual device/usage characteristics. Depending on feature weighting, the performance varies as much as 10 points, thereby emphasizing the importance of feature weighting for user modeling.

In text categorization, normalized *tf-idf* works well and normalized frequency does not compete [18]. In our case, N-TFIDF performs well, but N-FREQ performs the best. Thus the result is not completely identical to those in the text categorization literature. The reason can be considered as follows: compared to documents that have many functional words and popular words with less information, location data suffers less from such a problem. Therefore, a naive approach using a normalized frequency might work well.

Generally, recall is about 70% and precision is less than 50%. However, we have much better results for a particular set of user properties. For SMOKING, ROOM and COMMUTING, the F-values are as high as 64.13%, 67.00%, and 61.86% respectively with about 60-90% recall and 50-80% precision. The `under24` and `over40` values of AGE, most values of TEAM, and `high` value of COFFEE are all more than 10 points greater than the baseline. Some of them has more than F-values of 80% with 70-100% recall and 50-80% precision.

In summary, some user properties, such as TEAM and ROOM, can be predicted effectively using solely location information. To some degree, AGE, COFFEE, and SMOKING are also predictable. POSITION and WORK-FREQUENCY are difficult to predict.

We investigated feature weights from the learned models and found that some sensors are unexpectedly important; if we take COFFEE property for example, the ones around a coffee server are of the fifth and ninth importance among all sensors. The most important one was that in front of a small table where people gather for break. Some corridors are also recognized as important. Surprisingly a sensor exactly in front of the coffee server was slightly negatively weighted; it may be because around the sensor there is a copy machine and a door, thus the detection has little information for the property. These results show the limitation of our presumption for user behaviors and effectiveness of our approach.

## 5 Sensor Weighting

In actual use-cases, it is not always possible to prepare training data consisting of users' location histories and user properties. Then, the question arises: is there a way to find out whether a sensor is useful for future user modeling in advance without training data? In the real world situation, we often change sensor locations depending on actual user behaviors. Therefore it is useful if we can know the importance of sensors for future user modeling so that we can properly choose sensors to fix locations. This section

[4] Other kernels, such as linear and polynomial kernels, produce similar results overall; the results worsen by a few points.

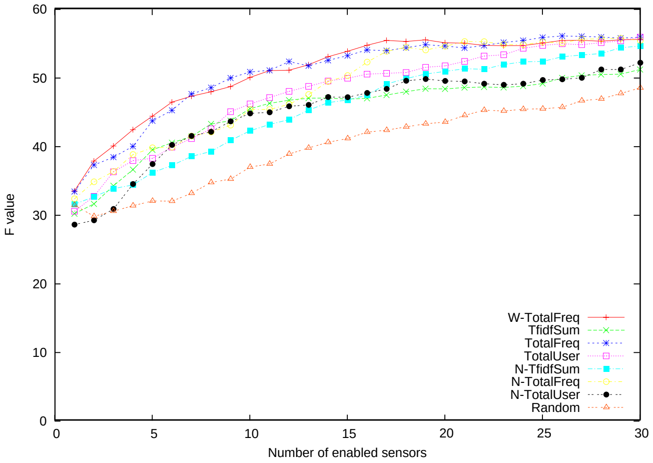


**Fig. 5.** Number of enabled sensors versus F-value.

describes an approach to measure the usefulness of sensors using only location histories. It is similar to *keyword extraction* for indexing documents for future retrieval.

### 5.1 Importance of Sensors

A sensor that does not detect users at all is almost useless, at least for user modeling purposes. Therefore, one definition to measure the usefulness, or the importance, of a sensor is simply the total number of detections: its frequency of detection. Alternatively, sensors that detect many different users might be important.

The importance of a sensor is understood as follows: The user-modeling performance becomes better than the other sets of sensors if we have a set of more ”important” sensors. In the context of information retrieval, several studies have examined finding good indexing terms for document categorization. Better indexing terms improve categorization performance [19].

We compare several importance measures for a sensor derived from text categorization studies. The importance of sensor $s_i$ is defined as follows: (i) Overall frequency (TOTALFREQ): $w(s_i) = \sum_{j=1}^{n} freq(s_i, u_j)$ (ii) Total detected users (TOTALUSER): $w(s_i) = uf(s_i)$ (iii) Total *Tf-idf* sum (TFIDFSUM): $w(s_i) = \sum_{j=1}^{n} tfidf(s_i, u_j)$ These functions can be normalized and taking a summation over every user, denoted as N-TOTALFREQ, N-TOTALUSER, N-TFIDFSUM.

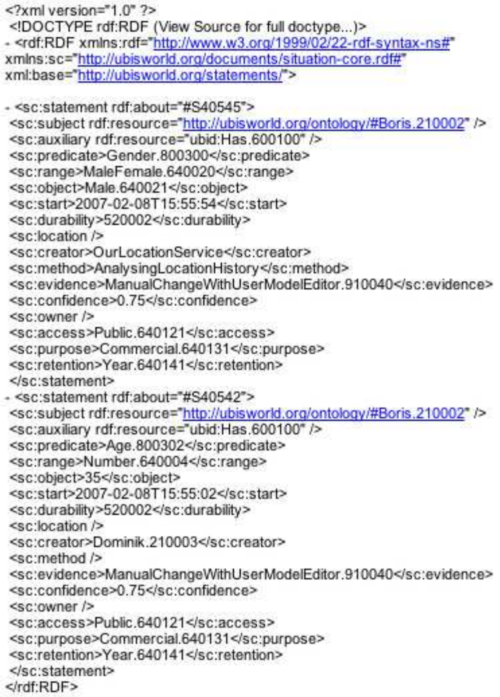
In addition, we use another importance measure, called weighted frequency (W-TOTALFREQ), following the intuition that a sensor that detects users who are detected by fewer sensors might be more important, as (iv) Weighted frequency (W-TOTALFREQ): $w(s_i) = \sum_{j=1}^{n} freq(s_i, u_j) \times \log(m/sf(u_j))$, where $sf(u_j)$ represents the number of sensors that detect $u_j$. This can be regarded as a *tf-idf* measure on the transposed sensor-user matrix. In summary, we have seven sensor importance measures.

### 5.2 Comparison and Results

Assume that we tentatively disable all sensors and enable them one by one in decreasing order of sensor importance. Eventually all sensors are enabled and the results will coincide in any case. However, if a sensor weighting method is superior to the others, the performance will improve faster. If sensor weighting is poor, the performance grows no faster than random selection of sensors does. This approach to evaluate feature selection is found in text categorization research [18, 19].

Figure 5 shows how the categorization performance changes over the number of enabled sensors. We used N-FREQ for feature frequency, and we used user properties that are shown to be predictable as shown in Section 4. In the figure, the undermost line (RANDOM) is plotted by selecting sensors randomly: it is shown as a baseline. The best performance is obtained by W-TOTALFREQ and TOTALFREQ. Other methods such as TOTALUSER, TFIDFSUM, and normalized series (N-TOTALFREQ, N-TOTALUSER, and N-TFIDFSUM) are better than RANDOM, but not as much as the best two.

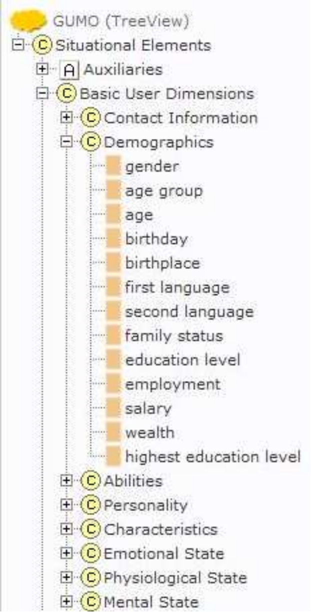
Sensor weighting is beneficial in several situations: we can identify which sensors might contribute to user modeling before learning the user properties. We can move sensors with low importance to elsewhere. These configurations of sensor allocation yield better performance during future user modeling.



```
<?xml version="1.0" ?>
 <!DOCTYPE rdf:RDF (View Source for full doctype...)>
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:sc="http://ubisworld.org/documents/situation-core.rdf#"
xml:base="http://ubisworld.org/statements/">

- <sc:statement rdf:about="#S40545">
 <sc:subject rdf:resource="http://ubisworld.org/ontology/#Boris.210002" />
 <sc:auxiliary rdf:resource="ubid:Has.600100" />
 <sc:predicate>Gender.800300</sc:predicate>
 <sc:range>MaleFemale.640020</sc:range>
 <sc:object>Male.640021</sc:object>
 <sc:start>2007-02-08T15:55:54</sc:start>
 <sc:durability>520002</sc:durability>
 <sc:location />
 <sc:creator>OurLocationService</sc:creator>
 <sc:method>AnalysingLocationHistory</sc:method>
 <sc:evidence>ManualChangeWithUserModelEditor.910040</sc:evidence>
 <sc:confidence>0.75</sc:confidence>
 <sc:owner />
 <sc:access>Public.640121</sc:access>
 <sc:purpose>Commercial.640131</sc:purpose>
 <sc:retention>Year.640141</sc:retention>
 </sc:statement>
- <sc:statement rdf:about="#S40542">
 <sc:subject rdf:resource="http://ubisworld.org/ontology/#Boris.210002" />
 <sc:auxiliary rdf:resource="ubid:Has.600100" />
 <sc:predicate>Age.800302</sc:predicate>
 <sc:range>Number.640004</sc:range>
 <sc:object>35</sc:object>
 <sc:start>2007-02-08T15:55:02</sc:start>
 <sc:durability>520002</sc:durability>
 <sc:location />
 <sc:creator>Dominik.210003</sc:creator>
 <sc:method />
 <sc:evidence>ManualChangeWithUserModelEditor.910040</sc:evidence>
 <sc:confidence>0.75</sc:confidence>
 <sc:owner />
 <sc:access>Public.640121</sc:access>
 <sc:purpose>Commercial.640131</sc:purpose>
 <sc:retention>Year.640141</sc:retention>
 </sc:statement>
</rdf:RDF>
```

**Fig. 6.** An ubiquitous user model with the GUMO ontology.



```
GUMO (TreeView)
⊟ C Situational Elements
    ⊞ A Auxiliaries
    ⊟ C Basic User Dimensions
        ⊞ C Contact Information
        ⊟ C Demographics
              gender
              age group
              age
              birthday
              birthplace
              first language
              second language
              family status
              education level
              employment
              salary
              wealth
              highest education level
        ⊞ C Abilities
        ⊞ C Personality
        ⊞ C Characteristics
        ⊞ C Emotional State
        ⊞ C Physiological State
        ⊞ C Mental State
```

**Fig. 7.** BasicUserDimensions in the GUMO ontology.

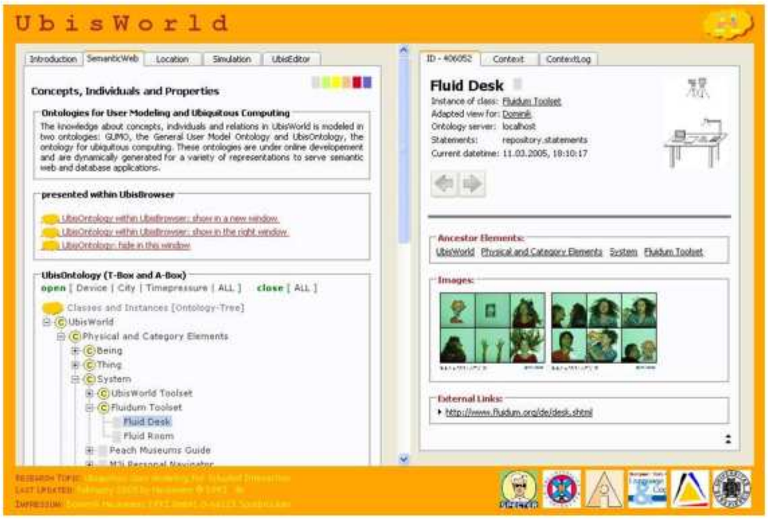
## 6 Ubiquitous User Models

Our algorithm can learn user's long-term properties such as gender, age, profession, and interests from location information. Consequently, the system can automatically produce a user model of a new user coming to the ubiquitous computing environment. As we previously claimed, user modeling and behavior detection in the ubiquitous computing environment are mutually complementary: if we have a more precise user model, we can guess more precisely the user behavior, and vice versa. Our algorithm that automatically obtains a user model will bootstrap creating ubiquitous user models to enable context-aware information services.

In order to realize user modeling for ubiquitous computing, several studies have been done in recent years. Heckmann proposes the concept of *ubiquitous user modeling* [10]. He proposes a RDF-based general user model ontology GUMO and a context markup language *UserML* that lay the foundation for inter-operability using Semantic Web technology. GUMO and *UserML* enable decentralized systems to communicate over user models as well as situational and contextual factors. The idea is to spread the information among all adaptive systems, either with a mobile device or via ubiquitous networks.

*UserML* statements can be arranged and stored in distributed repositories in XML, RDF or SQL. Each mobile and stationary device has an own repository of situational statements, either local or global, dependent on the network accessability. A mobile device can perfectly be integrated via wireless lan or bluetooth into the intelligent environment, while a stationary device could be isolated without network access. The different applications or agents produce or use *UserML* statements to represent the user model information. *UserML* forms the syntactic description in the knowledge exchange process. Each concept like the user model auxiliary hasProperty and the user model dimension timePressure points to a semantical definition of this concept which is either defined in the general user model ontology GUMO, the UbisWorld ontology, which is specialized for ubiquitous computing, or the general SUMO/MILO ontology.

Figure 6 shows the GUMO representation of an ubiquitous user model from location sensor information. Figure 7 shows BasicUserDimensions in the GUMO ontology. GUMO collects the user's dimensions that are modeled within user-adaptive systems like the user's age, the user's current position, the user's birthplace or the user's gender. In the GUMO ontology, long-term user model dimensions are categorized as demographics. Ontologies provide a shared and common understanding of a domain that can be communicated between people and heterogeneous and widely spread application systems. Since ontologies have been developed and investigated in artificial intelligence to facilitate knowledge sharing and reuse,



Concepts, Individuals and Properties

Ontologies for User Modeling and Ubiquitous Computing

The knowledge about concepts, individuals and relations in UbisWorld is modeled in two ontologies: GUMO, the General User Model Ontology and UbisOntology, the ontology for ubiquitous computing. These ontologies are under online developement and are dynamically generated for a variety of representations to serve semantic web and database applications.

presented within UbisBrowser

UbisOntology within UbisBrowser: show in a new window.
UbisOntology within UbisBrowser: show in the right window.
UbisOntology: hide in this window

UbisOntology (T-Box and A-Box)

open [ Device | City | Timepressure | ALL ]   close [ ALL ]

Classes and Instances [Ontology-Tree]

```
UbisWorld
  Physical and Category Elements
    Being
    Thing
    System
      UbisWorld Toolset
      Fluidum Toolset
        Fluid Desk
        Fluid Room
      Peach Museums Guide
      M3i Personal Navigator
```

Fluid Desk

Instance of class: Fluidum Toolset
Adapted view for: Dominik
Ontology server: localhost
Statements: repository.statements
Current datetime: 11.03.2005, 18:10:17

Ancestor Elements:
UbisWorld Physical and Category Elements System Fluidum Toolset

Images:

External Links:
http://www.fluidum.org/de/desk.shtml

**Fig. 8.** Annotating a ubiquitous user model in the Ubisworld.

they should form the central point of interest for the task of exchanging situation models. The web ontology language OWL has more facilities for expressing semantics. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. Thus, OWL is our choice for the representation of user model and context dimension terms and their interrelationships. This ontology should be available for all user-adaptive and context-aware systems at the same time, which is perfectly possible via internet and wireless technology. The major advantage would be the simplification for exchanging information between different systems. The current problem of syntactical and structural differences between existing adaptive systems could be overcome with such a commonly accepted ontology.

UbisWorld (Figure 8) enables users to annotate their user models with the GUMO ontology. UbisWorld represents persons, objects, locations as well as times, events and their properties and features. UbisWorld could be understood as a virtual coloured blocks world where each colour represents a different category in the ontology. The main focus of this approach lays on research issues of ubiquitous computing and user modeling. Apart from the representational funtionality, UbisWorld can be used for simulation, inspection and control of the real world.

## 7 Related Works and Discussion

Hightower distinguishes symbolic location systems and physical positioning technologies [20]. Our algorithm is applicable to symbolic location data. Therefore, some preprocessing is necessary for physical positioning data. For that purpose, studies to cluster position data into significant locations [6] are useful. Anonymous sensors are applicable to our approach if they are used with ID-sensors, as proposed in [21].

We discard timestamps of sensor detection. We are aware that this is a crucial abstraction. Nevertheless, we persisted in our approach for two reasons: First, there are numerous alternatives for converting timestamped sensory data into features. Tailored heuristic rules might improve the results, but we want to retain simplicity in our algorithm to protect its general applicability to many location data and many domains. Second, our algorithm is mainly inspired by works in information retrieval. We discard the ordering of sensor detections so that correspondence of the data structures is maximized. Considering that we would have increasing amount of location data in the real world, simplicity and scalability of information retrieval methods are of great use. For example, in Japan people use RFID cards when taking trains and shopping; almost every cell phone has GPS and broad band communication. In this environment, a vast amount of user location data is potentially available, which needs simple and scalable processing. However, we do not disregard the usability of timestamps; actually they have much information and can be used to improve our results. Our contribution in this paper is to show a bridge between techniques in information retrieval and ubiquitous computing.

Our algorithm can infer user properties if given location data history. A promising application domain of our algorithm is event spaces [14] such as conferences and business showcases, and large-scale shopping malls. Frequently, data mining of sales data is conducted using user demographic properties, which can be inferred by location data.

## 8 Conclusion

This paper proposes a new method to infer long-term user properties from a user's location history. Only the detection frequency is used. Machine learning techniques are applied to learn the pattern. Some user properties are well predictable. We also propose sensor weighting, which enables better allocation of sensors for future uses of modeling.

The algorithms in this paper are inspired by information retrieval. Because of the (proposed) structural similarity between sensor-user matrix and term-document matrix, we consider that many information retrieval techniques are applicable to sensory data. User modeling in ubiquitous computing research will contribute greatly in AI studies for modeling and recognizing human behaviors.

## References

1. Dourish, P.: What we talk about when we talk about context. Personal and Ubiquitous Computing **8**(1) (2004)
2. Jameson, A., Kruger, A.: Preface to the special issue on user modeling in ubiquitous computing. User Modeling and User-Adapted Interaction (3–4) (2005)
3. Want, R., Hopper, A., Falcao, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems **10**(1) (1992) 91–102
4. Wilson, D.H.: The narrator : A daily activity summarizer using simple sensors in an instrumented environment. In: Proc. UbiComp 2003. (2003)
5. Hightower, J., Consolvo, S., LaMarca, A., Smith, I., Hughes, J.: Learning and recognizing the places we go. In: Proc. UbiComp 2005. (2005)
6. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. Personal and Ubiquitous Computing **7**(5) (2003) 275–286
7. Liao, L., Fox, D., Kautz, H.: Location-based activity recognition using relational markov networks. In: Proc. IJCAI-05. (2005)
8. Lester, J., Choudhury, T., Kern, N., Borriello, G., Hannaford, B.: A hybrid discriminative/generative approach for modeling human activities. In: Proc. IJCAI-05. (2005)
9. Wilson, D., Philipose, M.: Credible and inexpensive rating of routine human activity. In: IJCAI-05. (2005)
10. Heckmann, D.: Ubiquitous Use Modeling. Ph.d thesis, University of Saarland (2005)
11. Camichael, D.J., Kay, J., Kummerfeld, B.: Consistent modelling of users, devices and sensors in a ubiquitous computing environment. User Modeling and User-Adapted Interaction **15**(3-4) (2005) 197–234
12. Kobsa, A.: Generic user modeling systems. User Modeling and User-Adaptied Interaction **11** (2001) 49–63
13. Jameson, A.: Modeling both the context and the user. Personal Technologies **5** (2001)
14. Nishimura, T., Itoh, H., Nakamura, Y., Yamamoto, Y., Nakashima, H.: A compact battery-less information terminal for real world interaction. In: Proc. Pervasive 2004. (2004) 124–139
15. Nakamura, Y., Nishimura, T., Itoh, H., Nakashima, H.: Id-cobit: A battery-less information terminal with data upload capability. In: Proc. IECON 2003. (2003)
16. Manning, C., Schütze, H.: Foundations of statistical natural language processing. The MIT Press, London (2002)
17. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag (1995)
18. Joachims, T.: Text categorization with support vector machines. In: Proc. ECML'98. (1998) 137–142
19. Mladenic, D., Brank, J., Grobelnik, M., Milic-Frayling, N.: Feature selection using linear classifier weights: interaction with classification models. In: Proc. SIGIR 2004. (2004) 234–241
20. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. IEEE Computer **34**(8) (2001) 57–66
21. Schulz, D., Fox, D., Hightower, J.: People tracking with anonymous and ID-sensors using Rao-Blackwellised particle filters. In: Proc. IJCAI-03. (2003) 921–928

# Ubiquitous social networks – opportunities and challenges for privacy-aware user modelling

Sören Preibusch[1], Bettina Hoser[2], Seda Gürses[3], and Bettina Berendt[3]

[1] German Institute for Economic Research, spreibusch@diw.de
[2] Universität Karlsruhe (TH), Institute of Information Systems and Management,
hoser@iism.uni-karlsruhe.de
[3] Humboldt University Berlin, Institute of Information Systems,
seda|berendt@wiwi.hu-berlin.de

**Abstract.** Privacy has been recognized as an important topic in the Internet for a long time, and technological developments in the area of privacy tools are ongoing. However, their focus was mainly on the individual. With the proliferation of social network sites, it has become more evident that the problem of privacy is not bounded by the perimeters of individuals but also by the privacy needs of their social networks. The objective of this paper is to contribute to the discussion about privacy in social network sites, a topic which we consider to be severely under-researched. We propose a framework for analyzing privacy requirements and for analyzing privacy-related data. We outline a combination of requirements analysis, conflict-resolution techniques, and a P3P extension that can contribute to privacy within such sites.

## 1 Introduction

With networked computers becoming more and more ubiquitous around the globe, digital social networks are gaining increasing importance for many people's work and leisure, as they allow for interaction independently of a fixed location. In parallel with their huge and growing acceptance among a wide range of users, social networks (SNs) are becoming a focus of attention for researchers and practitioners (especially in marketing). Also, governments and law enforcement (re-)awaken to the need to analyze the SNs of terrorists and other criminals [11]. What is important to all three groups is the huge amount of knowledge that can be discovered by investigating people's textual/multimedia contributions to SNs and the links they set to their "friends" – in this sense, social network analysis is an important topic for Knowledge Discovery for Ubiquitous User Modelling. (In this paper, we focus on SNs on the Web and thus on the ubiquity of the Web; see [8] for a differentiation between different notions of "ubiquity" that are relevant for user-centric analyses.[4])

Surprisingly, a topic that has received a lot of attention over the last years in all other areas of computer and Internet use, is scarcely attended to in current discussions on SNs: privacy. In the privacy statements of social network sites (SNSs), it appears that SNs are just another application on the Web (where "of course your privacy is very important to us"); the implication being that privacy challenges and problems are comparable to other Web applications, such as eCommerce, and therefore can be solved with the same privacy preservation methods.

In this position paper, we argue that while SNs share many privacy problems (and therefore solution possibilities) with other Web applications, there are also important new challenges. Using some simple examples, we highlight the extent of the current commercial interest in SN, point to the interest in SNs in ubiquitous computing environments, and discuss the resulting new challenges. Finally, we outline new research directions for currently existing methods for privacy-preserving data mining / data analysis.

[4] Many SN platforms are currently moving to mobile environments, e.g. [16]; in a separate paper, we will investigate how the issues raised here resurface or change in mobile SNs.

## 2 Background: The importance of social networks

In this section, we want to give an impression of the currently perceived importance of SNs. To this end, we focus on those examples that have recently received the most attention, in particular in terms of the monetary magnitude of takeover deals.

MySpace has grown to be the largest SNS in the world.[5] News Corp. invested 580 mio. USD in MySpace in mid 2005 [6], and one year later, Google signed a 900 mio. USD deal with News Corp. for the search feature [7]. Flickr[6] and del.icio.us[7] are both Yahoo!-owned; LinkedIn and Facebook are other prominent examples of SNs.

In Germany, two deals happened in the last year. First the SNS known as OpenBC went public and changed its name to Xing[8]. It has 1.5 million users; their market capitalization reached 164 mio. Euro [27]. The second deal was even more interesting. The publishing company Holtzbrinck has recently bought StudiVZ[9], a student community with more than 1 million accounts, for around 100 mio. Euro [33].

The next step appears to be Second Life[10]. This Web site is evolving into a parallel world, as more and more companies, universities, and users join. What differentiates this site from all other SNSs is its own flourishing economy. People earn real money within this virtual world. Second Life is likely to generate an enormous und unprecedented amount of social-network data.

The first interesting question behind all these deals is the economic rationale. Marketing and advertisement appear to be the major trigger behind these deals. It seems that companies want to use three characteristics of those sites to their advantage. First, all users voluntarily give information about themselves. This is more information than any company could collect without great expenses. Second, especially in sites for professional SNs like Xing, the company can rely on the correctness of the data, as only a true profile enables successful networking. Finally, networks are made visible through the analysis of simple interactions in the network, and thus provide supporting data sets for validating the classification of potential customers.

## 3 Marketing in social networks

Social Network Analysis has emerged from sociology in the 1970s. But the ground work has been laid in the 1930s when Moreno introduced graph-theoretic approaches to sociology. Since then the analysis of network structures based on mathematical indices has been of growing interest. With the Internet and thus the availability of ever-growing data sets in conjunction with the evolution of computer technology and algorithm design, social network analysts are now capable of analysing structures of large networks of small as well as large networks. This field of research has become highly multi-disciplinary, with research from mathematics, physics, sociology, information sciences and economics, e.g., [22, 23, 38, 3, 19].

The most common use of user data is in marketing, for which profiles, as collected in traditional eCommerce, are supported by data-mining the explicit self-descriptions, the behaviour, and the ratings of users (e.g., Amazon, Yahoo!, Google, and Google Mail). This use is explicitly mentioned, for example, in the MySpace privacy statement: "MySpace.com also collects other profile data including but not limited to: personal interests, gender, age, education and occupation in order to assist users in finding and communicating with each other. [...] MySpace.com also logs non-personally-identifiable information including IP address, profile information, aggregate user data, and browser type, from users and visitors to

[5] Estimates of the real number of users vary widely. While a much-cited blog of August 2006 stated that the threshold of 100 million accounts had been surpassed [12] – a number which was changed in all-too-many subsequent articles into more than 100 million users, an analysis of 303 random accounts showed that only between 30 and 40% of accounts are likely to belong to real users [4].

[6] close to 7 million accounts as of 10 Feb 2007, see http://www.flickr.com/search/people/?q=+

[7] 1 million accounts as of 25 Sep 2006, see http://blog.del.icio.us/blog/2006/09/million.html

[8] http://www.xing.com

[9] http://www.studivz.de

[10] http://www.second-life.com

the site. [...] This non-personally-identifiable information may be shared with third-parties to provide more relevant services and advertisements to members."[11]

Marketing initiatives also actively utilize the relational information in user profiles. (We believe that the under-specification of "profile" in the above privacy statement – 'profile information is information including, but not restricted to, ...' – legally allows MySpace to subsume network information under the profile that may be handed over to third parties. To the best of our knowledge, no legal investigation or lawsuit on this question has been published.)

Developing a functioning marketing strategy for an SNS requires at least two things: First, to find out how to address people in an environment geared towards "friends", who also tend to be highly Internet-savvy and hence may not respond to traditional forms of marketing. Second, to utilize the information inherent in linkage patterns to discover and target high-value customers.

The first strategy can be subsumed under "Guerrilla marketing": unconventional ways of performing promotional activities, often on a very low budget, with high entertainment value and leaving people unaware that they have been marketed to ("undercover marketing"), see [21]. This is one of the currently most-hyped marketing strategies (see the study by the German Society for Consumption Research, [15]), and recommendations specifically tailored to the SNS MySpace exist [14].

However, these recommendations rely more on the creativity and motivation of marketing employees to engage in an SN, than on the utilization of formal models. The question arises what kind of information *is* contained in the network structure. This is a typical question of social network analysis [37]. Combining social network analysis and data mining, [29] proposed to "mine the network value of customers" and to use this knowledge for "viral marketing". Viral marketing denotes "marketing techniques that use existing SNs to produce increases in brand awareness, through self-replicating viral processes, analogous to the spread of pathological and computer viruses. It can often be word-of-mouth delivered and enhanced online; it can harness the network effect of the Internet and can be very useful in reaching a large number of people rapidly" [40]. The core idea of [29] is to exploit measures of "opinion leadership" inherent in SNs and to translate them into measures of customer value. Thus, they distinguish between a customer's intrinsic value (based on the products s/he is likely to purchase) and the network value (the expectation that s/he has a positive influence on others' probabilities of purchasing).

"Customer network value" is but one example of measures of node importance. In the social network literature, many other measures are currently being discussed; it is beyond the scope of this paper to enter the discussion of their relative merits.

In ubiquitous environments, marketing companies are hoping for even more detailed information. Ubiquitous information is expected to return higher granularity data with strong identifiers like location and time, which not only allow persons to be easily identified, but also their interactions in the social realm to become overt, including their belonging to groups of which they are not even aware of. An example is a specific group of commuters who pass by strategically-placed digital billboards. The collection and dissemination of ubiquitous information will allow advertising and marketing companies to optimally make use of the time and places at which persons may best succumb to advertisement, as well as to identify those groups or individuals best suited for various viral marketing strategies.

## 4 Privacy challenges in social networks

In what sense is all this a privacy problem? First, because being an SNS user implies being a Web (platform) user, all the problems arise that are already well-known and documented in the Internet at large, e.g., [35]. Summarized briefly, personal data accrue and can be utilized not only for the primary purposes for which they were collected (finding and communicat-

[11] http://www.myspace.com/Modules/Common/Pages/Privacy.aspx [10 Feb 2007]

ing with other users, cf. the MySpace privacy statement).[12] They can be utilized also for secondary (from the perspective of the user) purposes that are covered in the SNS's terms of use and in that sense accepted by users. Such purposes are usually targeted marketing. However, they can also be utilized for other purposes – illegally or legally for commercial purposes, as many examples, for example from eCommerce, show [26], and by law enforcement, secret services, etc. (the explicit targeting also of information marked as 'private' in law-enforcement analyses of data has been confirmed by leading politicians [28]).

Technically, the use of SNS data for novel purposes is even simpler than in traditional eCommerce. The very essence of social media is that user-profile information is public (as opposed to, for example, Amazon's usage data which are an important and secret business asset of the company). Moreover, the data often carries semantic markup and/or is presented in a uniform (hence easily minable) manner, for example as RSS feeds. Thus, while the legal issues at this level are the same in SNSs as in other sites, technical (ab)uses become simpler.

So at first sight, social-network data describe a person in the same way as other data. For example, a "person" record in a database may contain the attributes "health status", "favourite book", and a (probably set-valued) attribute "friend". The values of these attributes are properties of the data subject of this record (say, person A).

For the subsequent analysis, we propose to extend the common classification of confidentiality levels into "private data" and "public data" agreed upon between the customer (user) and the site operator. We propose to use two further levels that we call "community data" and "group data", specific to SNSs:

**Private data** is disclosed to the SNS operator for its internal purposes only. This data must not be disclosed unless explicit consent is given. An example is the user's email address provided upon registration.

**Group data** is disclosed to the SNS operator and can be accessed by other users of the same SNS that are also in the same group as the user: data disclosure is limited to the group. Here we imagine messages shared among a certain group, almost like a closed mailing-list.

**Community data** has been disclosed to the SNS operator and is available to all registered and logged-in users of the SNS. The data is not accessible for anonymous SNS visitors. Examples are the user's online status, her contacts, her member page details, photos, etc.

**Public data** has been disclosed to the SNS operator and is made accessible for all SNS visitors, including anonymous visitors: this may include the fact that the user is registered in the SNS, her user name, or her guestbook.

The concrete details and the application of these confidentiality levels to data depends on the SNSs implementation. One may not always find disclosure examples of all levels.

A priori, the site operator has diverging privacy goals. On the one hand, he needs enough personal user information to be disclosed in order to attract new users. On the other hand, some information must be kept at the community level to create sufficient benefit from community membership. At the time of signing up, the perceived benefits, including access to secured personal information, must exceed registration costs. A typical situation is that one searches for someone's email address by entering her name in a search engine. The contact is found in an SNS like Xing, but the email address is secured to registered users.

The privacy challenges in the Web portrayed so far arise from the operator-user interaction. In the context of SNSs, new problems arise because of the semantics of social-network relations, i.e. the user-user interaction. As an example, consider friendship relations which are – at least in real life – symmetric. Thus, the record of person A that states that person B

[12] Even accesses that at first sight look like a legitimate usage in this sense are not without problems, and people are beginning to be wary of this. The following is a good example of the new intricacies of the shifting notions of "private" and "public": boyd [10] pointed out that many US teenagers (due to their heavy usage of MySpace both the most sophisticated and the most vulnerable users of SNSs today) feel strongly about preserving a certain form of privacy: they want to be visible and searchable for their friends but not their parents.

is a friend also contains information that is part of B's record. Another example is groups of users. Group attributes may be changed by any member of the group. A user whose group membership is public thereby discloses interests, preferences, or other personal information (for a worked-out example, see Section 6). This means that if A discloses information about himself or groups including himself, he (whether willingly or inadvertently) also discloses information about someone else. Expressed differently, A's treatment of his privacy has a direct effect on B's privacy.

Such social-network data usually concern people who also have an ID in the same system, i.e. this privacy dependency is a problem that affects different users of the same system.

In addition, problems arise when systems support the interaction with the world outside the system. For example, Google Mail (Gmail) users consent to their emails' data being analyzed by Google; however, all incoming mails of a Google Mail account (whether sent by another Gmail user or by somebody else) are also analysed. Thus, A's treatment of his privacy also has direct external effects on the privacy of C, who is a non-user of the system.

The distinction between "in the system" and "outside the system" vanishes in case of loosely coupled networks where members may engage in relationships spontaneously and without a central authority. An example are the "friend-of-a-fried" nets built by publishing FOAF files [31]. A FOAF file describes a persons contact information, as well as his/her relationships to other people and details about them in an RDF-based standard format. As users publish their friendship details autonomously, symmetry of relationships is not enforced. However, revelation of private information is likely to occur for instance by combining real names and email addresses, and legal requirements apply [13].

Because SNs are (by definition) built on interaction, they are typically open systems, and have certain semantic characteristics. Each privacy-related declaration has effects beyond the interaction between one individual data subject and one data collector, effects that may concern a number of stakeholders who may or may not be users of the same system.

In a quest for solutions, we identify two essential steps: First, the potential privacy conflicts that arise by social-network interaction must be identified. To do this in a systematic way, methods from requirements analysis are needed. This includes methods for conflict resolution a priori. Second, privacy preferences and requirements must be formalized sufficiently such that software can automatically detect problems, alert the user, and assist her. In data analysis routines, mechanisms need to be implemented to enforce privacy requirements. We believe that this should be based on existing standards or de facto standards for privacy-enhancing technologies (PETs), in order to make a large-scale adoption of such technological solution approaches realistic. In the following two sections, we investigate the two parts of our solution proposal in turn.

This method of analysis draws attention to an important question: what "privacy" actually means. In the following, we emphasize that privacy is not just about data protection, or about restricting the access to, or the processing of, personal data. It is also about who can edit which data (e.g., information about individuals or groups), how people want to and can interact with a site and other users (e.g., identified, pseudonymized, or anonymized), i.e. what different private, public, and shared spaces they can create for their lives, how they can separate and share identities between these spaces, etc. For an extended discussion of our notion of privacy, see [17].

## 5 Identifying privacy conflicts in the interaction of requirements for social network sites

As mentioned in the examples above, identifying privacy conflicts in SNSs is not trivial. In order to do this in a systematic manner, we make use of the Multilateral Security Requirements Analysis (MSRA) method [17]. The main idea of the MSRA method is to consider the security and privacy interests or needs of all stakeholders related to the system. An important aspect of the method is to identify interest conflicts among these stakeholders and develop mechanisms for negotiating these conflicts. Here we introduce aspects of conflict identification and negotiation mechanisms in multilateral security requirements analysis.

**Stakeholders and their privacy interests** In MSRA, *stakeholders* of a system are all persons who have some functional, knowledge, security or privacy interest in the system. This encompasses all persons involved in the conception, production, use and maintenance of the system. Stakeholders encompass more than *users* (those who will use the functionality of the system).

Stakeholders, for example, include all persons who have a privacy interest in the system. This could be stakeholders representing legal requirements as well as non-users whose data is processed by the system – i.e. patients in a Hospital Information System or customers in a Customer Relationship Management System. As mentioned in Section 4, the sender of an email to a Gmail account may count as a stakeholder of the Gmail platform, although she is not a user of that platform. This stakeholder is likely to have different privacy interests towards the Gmail platform than a user or provider of the platform.

The inclusion of an external sender of an email as a stakeholder of an email platform also points to the fact that further stakeholders may be acquired once the system is running. Subsuming the privacy interests of all prospective stakeholders is not possible during the development of the system. Nevertheless, the potential of discovering new stakeholders requires the conception of negotiation mechanisms during the development process that anticipate potential divergences in privacy interests during run-time. Moreover, the introduction of new stakeholders and their requirements often demands a review of all security and other requirements and hence an iterative approach.

The analysis of the stakeholders security and privacy requirements can be compared to viewpoints-oriented requirements analysis [32]. The collection of different privacy interests from the viewpoints of the stakeholders results in a complex list of requirements that are likely to include inconsistencies, repetitions and conflicts. To identify these, requirements interaction management is necessary.

**Identification of privacy conflicts through requirements interaction management** Requirements interaction [30] can be understood through direct comparisons of requirements descriptions, or through the analysis of the underlying components that can satisfy these requirements. According to the definition in [30], a requirement $R$ is satisfied by a component $C$ if the component exhibits all the properties specified in the requirement. There may be degrees of satisfaction of a requirements and this can be mapped to a range:

**Definition 1.** $Sat_R : C \rightarrow [0, 1]$

As a result, requirements interaction can be defined as follows:

**Perceived interaction** : Two requirements, labeled $R_1$ and $R_2$ *interact* if and only if the satisfaction of one requirement affects the satisfaction of the other.

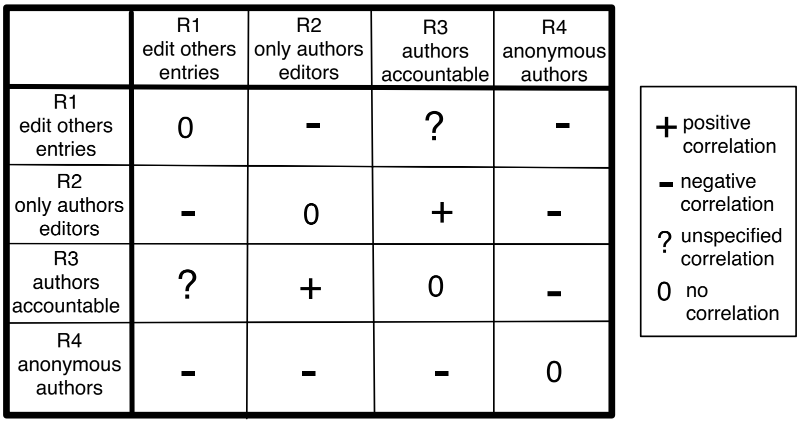
**Operational interaction** : If component $C_1$ satisfies $R_1$ and component $C_2$ satisfies $R_2$, and the run-time behaviour of $C_1$ affects the run-time behaviour of $C_2$, then $C_1$ interacts with $C_2$, and indirectly $R_1$ interacts with $R_2$.

The definition of operational interaction points to the dependency between the requirements and design phases of systems development. Interactions may have different degrees of intensity, and run-time interactions may have varying probabilities of appearing. Interactions between requirements may be positively correlated (they strengthen each other), negatively correlated (they are in conflict), the correlation may be unspecified (the effect is unclear but exists) or non-existent (no effect).

Privacy requirements can be articulated in terms of security goals [17]. Security goals also interact, and may be correlated positively or negatively. For example, the anonymous use of a resource and the accountability for that use – the possibility to prove to a third party the use of the same resource – are conflicting requirements. Design solutions that partially satisfy both are possible; we will refer to these later.

Example 1. In an SNS, the stakeholders may have conflicting interests concerning the authoring and editing of entries:

**R1** The members of the SN may edit parts of entries of other authors that contain information about themselves.
**R2** The authors of entries want to be the sole editors of their own entries.
**R3** All members of the SN want the accountability of authors for all their entries towards all other members.
**R4** All members of the SN want to be able to use the SNS services anonymously.



| | R1<br>edit others entries | R2<br>only authors editors | R3<br>authors accountable | R4<br>anonymous authors |
|---|---|---|---|---|
| R1<br>edit others entries | 0 | - | ? | - |
| R2<br>only authors editors | - | 0 | + | - |
| R3<br>authors accountable | ? | + | 0 | - |
| R4<br>anonymous authors | - | - | - | 0 |

+ positive correlation
- negative correlation
? unspecified correlation
0 no correlation

**Fig. 1.** Requirements interaction for a social network

Figure 1 gives an overview of the interactions between these initial requirements for various sets of stakeholders. For example, the anonymity requirement R4 is obviously in conflict with all the other requirements. If a user uses the services of the SNS anonymously, it is not possible to prove that information in an entry is about oneself (requirement R1), it is not possible to authenticate the users who edited entries through their identities (requirement R2), and accountability for requirements is not possible through user identities (requirement R3). Hence, some negotiation is necessary to resolve the negative and unspecified correlations between the different requirements. Resolutions of conflicts may also introduce new conflicts. Thus, an iterative requirements interaction management approach is needed.

**Activities and negotiation techniques** In [30], Robinson et al. suggest six activities:

**Requirements partitioning** : a subset of the requirements are analysed depending on scenarios, stakeholder views etc ("episodes" in MSRA).
**Interaction identification** : the different kinds of correlations between the requirements are identified.
**Interaction focus** : the requirements are prioritized, since not all interactions can be resolved.
**Resolution generation** : different approaches are used to generate resolutions. A value-oriented approach considers alternative goals, whereas a structure-oriented approach considers new operators and resources.
**Resolution selection** : different methods are used to prioritize generated resolutions, for example, utility theory or decision theory.
**Requirements update** : further stakeholders and/or requirements may become apparent through the requirements interaction management process; these are considered in this activity.

Based on their study of approximately conflict resolution 30 methods, the authors suggest the following methods for resolution generation:

**Relaxation** : the conflicting requirements are relaxed or generalized to avoid conflict.
**Refinement** : the conflicting requirements are partially satisfied.
**Compromise** : a compromise is found between the requirements.
**Restructuring** : a set of methods are used to modify the conflict context, which includes assumptions and related requirements.
**Other** : conflict resolution is postponed, either to later stages of the development, or the attempt is abandoned entirely.

Example 1 (contd.) In the example, the conflict between the requirements R1 through R3 with the anonymity requirement R4 can be solved with one of these resolution methods. A relaxation of the anonymity requirements can be reached by replacing anonymity by pseudonyms of different strength. Refinement could be reached by allowing certain services to be used anonymously, i.e. authoring reserved entries anonymously but using services for which accountability is important with registered pseudonyms. This could also be seen as a compromise. In restructuring, one could divide the services of the SNS into those which include anonymous interactions, and others which exclude anonymous interactions. Further restructuring could be done through keeping the community so small and protected that anonymity ceases to be a requirement.

Recognizing interactions in privacy and security requirements written in natural language is not a trivial activity. We need an adequate modelling language that makes the identification of interactions easier [24]. Further, the interaction between the high-level security requirements of the stakeholders and the data that are related to these requirements needs to be analyzed, which inevitably requires inference analysis to be undertaken.

## 6 Enhancing privacy in social network sites using P3P

What happens after requirements have been analyzed and conflicts identified? How can technology help to resolve conflicts during run time? In this section, we focus on restructuring: a modification of the SNS application logic (and hence the interaction/conflict context) that can help avoid the occurrence of conflicts. We concentrate on privacy in the sense of data protection, i.e. as a restriction on data access and data processing.

Appropriate measures need to be taken to satisfy privacy requirements in an operational SNS. This includes the conception and adaptation of technologies and processes, mainly privacy languages and tools to interpret and enforce these languages. The design goal is twofold:

First, we need mechanisms to ensure that data/information of one privacy level must not be made accessible via data/information of a lower privacy level. For example one should not be able to perform *(data) inferences* [34] towards personal information that is private on a “community level”, from personal information that is private on a “public level”. The AOL privacy breach [5] gives evidence that trivial anonymization is insufficient for preventing data inferences that may even lead to the identification of individuals.

Second, we need mechanisms that prevent users from disclosing personal information about other users inside an SNS.

Both objectives should be addressed within the existing technological and legal infrastructure of Privacy Enhancing Technologies (PETs), Privacy Protocols (especially P3P and APPEL / XPref), and mandatory legislation.

P3P, the Platform for Privacy Preferences, is a protocol designed to inform Web users about the data-collection practices of Web sites. It provides a way for a Web site to encode its data-collection and data-use practices in a machine-readable XML format known as a P3P policy [39]. Moreover, P3P enables Web users to understand what data will be collected by sites they visit, how that data will be used, and what data/uses they may “opt-out” of or “opt-in” to [39]. An SNS operator will post a P3P policy on its Web site to communicate its data handling practices. Visitors and users can receive this policy in a textually presented format. Their decision whether to send data to the site or not can be supported by APPEL rules: APPEL, A P3P Preference Exchange Language, allows a user to express her preferences in a set of preference rules, interpreted by her user agent to make automated or semi-automated decisions regarding the acceptability of P3P Privacy Policies [20]. XPref [1] is a newer privacy preference language, more expressive than APPEL yet easier to use.

In a P3P policy, one or several statements describe data practices that are applied to particular types of data. A statement indicates recipients, usage purposes, and a retention time for data elements. Every potential data usage must be indicated by an appropriate statement; hence statements span a superset over the actually implemented data usage. P3P hereby translates the privacy concepts of, e.g., European privacy legislation and the OECD Fair Information Practices into a machine-readable policy.

Example 2. Consider the P3P fragment below, which expresses the data collection and usage scenario outlined in section 4. A professional SN collects the username, publicly accessible, and the details about the user's job, the latter being secured. Users may join special interest groups based on their industrial and departmental focus, e.g. “Helpdesk Professionals Group”, “Data Protection Officers Group”, or “CEO VIP Club”. Group membership, expressed by the data categories `<political/><preference/>` is public.

**Listing 1.1.** P3P Policy fragment

```
<STATEMENT>
  <PURPOSE>      <current/>  </PURPOSE>
  <RECIPIENT>    <ours/>
                 <public/>  </RECIPIENT>
  <RETENTION>    <indefinitely/>  </RETENTION>
  <DATA-GROUP>   <DATA ref="#user.login.id"/>
                 <DATA ref="#dynamic.miscdata">  <CATEGORIES>
                 <political/> <preference/>  </CATEGORIES>  </DATA>  </DATA-GROUP>
</STATEMENT>
<STATEMENT>
  <PURPOSE>      <current/>  </PURPOSE>
  <RECIPIENT>    <ours/>  </RECIPIENT>
  <RETENTION>    <indefinitely/>  </RETENTION>
  <DATA-GROUP>   <DATA ref="#user.login.password"/>
                 <DATA ref="#user.jobtitle"/>
                 <DATA ref="#user.business.employer"/>
                 <DATA ref="#user.business.department"/>  </DATA-GROUP>
</STATEMENT>
```

However, group details must be public so that users can decide whether they want to join a given group. Even if these details are hidden, the group name is often explicit enough (“Data Protection Officers Group”).

Thus, we can formulate the following inference rule with infix relation notation of is_member_in, focusses_on, and works_in:

$$\forall\, g{:}Group, u{:}User, d{:}Department : u \text{ is\_member\_in } g \wedge g \text{ focusses\_on } d \Rightarrow u \text{ works\_in } d$$

Using this data inference rule, one can infer a user's department from her group membership details; the group details are public and can be accessed freely. The confidentiality of the user's department is not guaranteed any more, and the P3P Policy does not accurately reflect that the recipient of the department information is effectively broadened to `<public/>`.

To avoid such degradation of privacy levels, we have proposed an extension to P3P, the INFERENCE element, together with a logic that blocks the use of data described in the INFERENCE [36]. The new INFERENCE element, realized by P3P's built-in extension mechanism and thus backward-compatible, codes a data inference inside the P3P policy. A user-agent may parse the inference rule and alert the user to possible privacy breaches. Inside an analysis framework, inference rules can be used to lift privacy levels. For instance, access to the group membership information should be restricted to `<ours/>`.

Example 2 (contd.) In P3P, the inference rule is coded as follows:

**Listing 1.2.** P3P Policy Extension for coding inferences

```
<EXTENSION optional="no">
<INFERENCES xmlns="http://preibusch.de/namespaces/SIMT/inferences">
  <INFERENCE>
    <CONSEQUENCE> If group membership is known, group details let
                  conclude on the user's details. </CONSEQUENCE>
    <GIVEN>  <AND>
      <DATA-GROUP>
        <DATA ref="#dynamic.miscdata">  <CATEGORIES>
          <political/>  <preference/>  </CATEGORIES>  </DATA>  </DATA-GROUP>  </AND>
    </GIVEN>
    <INDUCED>
      <DATA-GROUP>  <DATA ref="#user.business.department"/>  </DATA-GROUP>
    </INDUCED>
  </INFERENCE>
</INFERENCES>
</EXTENSION>
```

Example 3. We now consider the second problem of personal information about oneself to be disclosed by other users. Again, we observe `<public/>` as a new recipient where a higher privacy level was intended. As a remedy, the users A and B have to agree on a privacy policy that B will not disclose their friendship. Note that a privacy policy between A and the SNS operator does not cover B's privacy obligations. Nevertheless, the operator may provide privacy policy templates and implement measures to ensure that B does not make public his friendship to A unless A has given her consent.

The choice between an open (public) or hidden (private) friendship can be offered via the mechanisms provided in [25]. Similar to the coding of inferences in P3P, different usage options for the SNS are coded in a single valid P3P Privacy Policy. Therefore, a user agent can seamlessly parse those alternative scenarios of friendship making and select the most appropriate option for the user (see Listing 1.3 below). The policy negotiation and the choice of the right option is automated so that the "overhead" is transparent to the SNS user. The necessary XML schemas and namespaces are available, see [25].

Example 3 (contd.) The scenarios of friendship making are described in P3P as follows:

**Listing 1.3.** Different friendship alternatives (public/hidden)
are coded in a single P3P Privacy Policy

```
    <POLICY xmlns:PRINT="http://preibusch.de/namespaces/PRINT/PRINT.xsd">
  <EXTENSION optional="no">
    <PRINT:NEGOTIATION-GROUP-DEF  id="friendship"
     standard="public_friend"  fallback="public_friend"  selected="public_friend"
     description="Choosing public (open) or private (hidden) friendship" />
  </EXTENSION>
  <STATEMENT>  <EXTENSION optional="no">
      <PRINT:NEGOTIATION-GROUP id="public_friend" groupid="friendship"
       serviceuri="/make-friend/public"
       description="Make this user a public friend of yours" />
    </EXTENSION>
    <CONSEQUENCE>Other visitors will see that you are friends</CONSEQUENCE>
    <RECIPIENT>  <ours/>
                 <public/>  </RECIPIENT>
    <PURPOSE>    <contact/>
                 <other-purpose> friendship </other-purpose>  </PURPOSE>
    <RETENTION>  <indefinitely/>  </RETENTION>
    <DATA-GROUP> <DATA ref="#user.login.id"/>  </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>  <EXTENSION optional="no">
      <PRINT:NEGOTIATION-GROUP id="hidden_friend" groupid="friendship"
       serviceuri="/make-friend/hidden"
       description="Make this user a hidden friend of yours" />
    </EXTENSION>
    <CONSEQUENCE>Other visitors will not see that you are friends</CONSEQUENCE>
    <RECIPIENT>  <ours/>  </RECIPIENT>
    <PURPOSE>    <contact/>
                 <other-purpose> friendship </other-purpose>  </PURPOSE>
    <RETENTION>  <indefinitely/>  </RETENTION>
    <DATA-GROUP> <DATA ref="#user.login.id"/>  </DATA-GROUP>
  </STATEMENT>
</POLICY>
```

As the friendship making process is realized through the SNS, the SNS operator can record the chosen option and integrate enforcement mechanisms into the site [2]. When displaying a user's friends list, only public friends will be listed. The scenario demonstrates that privacy enhancements can be implemented without disturbing the user. The standard-compliant coding in machine-readable privacy policies allows for computer-supported decision-making. Moreover, the content presentation becomes semantics-driven as it is governed by semantic policies; policies will provide for privacy even if the friendship may no longer exist.

## 7 Conclusion

In this paper, we have shown that privacy in SNSs is of growing interest as these sites gain economic relevance. As companies buy SNSs for the inherent marketing potential and sites like Second Life create parallel economic worlds, it should be of interest to the user and even more to researchers and software developers how to implement techniques that provide users with "digital privacy". If this is not achieved, a backlash could result as we observed for eCommerce in the late 1990s.

While SNSs already use some privacy functions and have their own privacy policies, these are still centered around the individual, although SNSs clearly take into account network effects. If for example one user reveals data about himself, as well as a list of his friends, this "network" information could lead to revelations that had not been intended by his friends. Such leaks can prove bothersome or disastrous for individual users. In addition, these users may lose trust in the SNS and leave, which in turn creates problems for the operators of the site and the marketing initiatives financing them (this happened in one of the sites mentioned in Section 2, StudiVZ). This shows that both sides have a vital interest in effective privacy measures. In this paper, we aimed at contributing to the discussion about privacy in SNSs, a topic which we consider to be severely under-researched. We proposed a framework for analyzing privacy requirements and for analyzing privacy-related data.

To build on a comprehensive notion of "privacy", we investigated desired properties of (inter)actions on the one hand and issues of data confidentiality on the other hand. We developed a data confidentiality taxonomy to capture the privacy specificities in SNs: The (intended) interaction with other users, especially with "friends" inside the network, can result in personal data being disclosed by third parties and in other data being inferred from users' communication patterns. We outlined methods for multilateral requirements analysis for identifying, negotiating, and – if possible – resolving conflicts already during system design. The dichotomous distinction between "public data" and "private data" was refined to a set of tiered confidentiality levels. We provided an extension to the Privacy Policy language P3P to code data inferences that may result in confidentiality level breaches. The machine-readable coding of inferences allows for a better-informed consent, as the user becomes aware of side-effects. In particular, symmetric relations like "friendships" are potential privacy pitfalls as one user's disclosure makes it possible to draw conclusions about other users' data. We provided mechanisms how privacy policies can be integrated seamlessly into the interaction among users. These policies give semantics to confidentiality and can be enforced by SNS operators.

Many challenges lie ahead. They include further investigations of the formal characteristics of the proposed inference (avoidance) schemes, practical applications and the development of best practices in requirements analysis and conflict resolution, and last but not least extensive user studies on the usability of concrete, implemented privacy options (these studies could build on the methods of, e.g., [16, 9, 18].

## References

1. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. XPref: a preference language for P3P. *Computer Networks*, 48, 2005.
2. Anne Anderson. The Relationship Between XACML and P3P Privacy Policies, 2004. `http://research.sun.com/projects/xacml/XACML_P3P_Relationship.html`.
3. A. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.
4. Barbarian. Debunking the MySpace Myth of 100 Million Users, 2006. `http://www.netscape.com/viewstory/2006/09/27/the-real-number-of-myspace-users`, 27 Sep 2006.
5. M. Barbaro and T. Zeller. A face is exposed for AOL Searcher No. 4417749. *New York Times*, 9 August 2006.
6. BBC. News Corp in USD 580m internet buy. *BBC News*, 2005. `http://news.bbc.co.uk/2/hi/business/4695495.stm`, 15 Feb 2007.
7. BBC. Google signs USD 900m News Corp deal. *BBC News*, 2006. `http://news.bbc.co.uk/2/hi/business/5254642.stm`, 15 Feb 2007.
8. B. Berendt and E. Menasalvas. Introduction. In *Proceedings of the Workshop on Ubiquitous Knowledge Discovery for Users at ECML/PKDD 2006*, pages 1–2, 2006. `http://vasarely.wiwi.hu-berlin.de/UKDU06/Proceedings/UKDU06-proceedings.pdf`.
9. Shlomo Berkovsky, Nikita Borisov, Yaniv Eytani, Tsvi Kuflik, and Francesco Ricci. Examining users' attitude towards privacy preserving collaborative filtering. In *Proceedings of DM.UM '07*, 2007. `http://vasarely.wiwi.hu-berlin.de/DM.UM07/Proceedings/berkovsky.pdf`.
10. danah boyd. Identity production in a networked culture: Why youth heart MySpace. In *Annual Meeting of the American Association for the Advancement of Science, St. Louis, MO. February 19*, 2006. `http://www.danah.org/papers/AAAS2006.html`.
11. Bundesregierung, 16. Wahlperiode. Antwort der Bundesregierung auf die Kleine Anfrage der Abgeordneten Jan Korte, Petra Pau, Kersten Naumann und der Fraktion DIE LINKE. Drucksache 16/3787. Rechtmäßigkeit und Anwendung von Online-Durchsuchungen, 2006. `http://dip.bundestag.de/btd/16/039/1603973.pdf`.
12. Pete Cashmore. MySpace Hits 100 Million Accounts, 9 Aug 2006. `http://mashable.com/2006/08/09/myspace-hits-100-million-accounts/`.
13. European Court. Judgment of the Court of 6 November 2003. Criminal proceedings against Bodil Lindqvist, 2003. `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:62001J0101:EN:HTML`.
14. Janie Gafford. Guerilla Marketing on MySpace-Smart Do-It-Yourself Online Marketing, 2007. `http://ezinearticles.com/?Guerilla-Marketing-on-MySpace-Smart-Do-It-Yourself-Online-Marketing&id=433759`.
15. GFK Marktforschung GmbH Bereich Online Research. Marktforschungsstudie zur Nutzung alternativer Werbeformen [market research study on the use of alternative forms of marketing.
16. G. Groh. Groups and group-instantiations in mobile communities – detection, modeling and applications. In *Proceedings of the International Conference on Weblogs and Social Media 2007*, 2007. `http://www.icwsm.org/papers/paper7.html`.
17. S.F. Gürses, B. Berendt, and Th. Santen. Multilateral security requirements analysis for preserving privacy in ubiquitous environments. In *Proceedings of the Workshop on Ubiquitous Knowledge Discovery for Users at ECML/PKDD 2006*, pages 51–64, Berlin, September 2006. `http://vasarely.wiwi.hu-berlin.de/UKDU06/Proceedings/UKDU06-proceedings.pdf`.
18. Indratmo and Julita Vassileva. A usability study of an access control system for group blogs. In *Proceedings of the International Conference on Weblogs and Social Media 2007*, 2007. `http://www.icwsm.org/papers/paper33.html`.
19. Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
20. M. Langheinrich. A P3P Preference Exchange Language (APPEL), 26 February 2001. W3C Working Draft., `http://www.w3.org/TR/P3P-preferences`.
21. Jay Conrad Levinson. *Guerrilla marketing*. Houghton Mifflin, 1984.
22. S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.
23. S. Milgram and J. Travers. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.
24. Adeniyi Onabajo and Jens H. Jahnke. Modelling and Reasoning for Confidentiality Requirements in Software Development. In *ECBS*, 2006.
25. Sören Preibusch. Privacy Negotiations with P3P. In *W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement*, 2006. `http://www.w3.org/2006/07/privacy-ws/papers/24-preibusch-negotiation-p3p/`.

26. Privacy Rights Clearinghouse. A Chronology of Data Breaches, 2005–2007. http://www.privacyrights.org/ar/ChronDataBreaches.htm.
27. Inken Prodinger. Ins Netz gegangen. *Börsen-Zeitung*, 15 Feb 2007. http://corporate.xing.com/fileadmin/image_archive/review_Boersen-Zeitung_050107_eng.pdf.
28. Christian Rath. "Terroristen sind auch klug" [Terrorists are clever too] – Interview with Wolfgang Schäuble. *die tageszeitung*, 2007. http://www.taz.de/pt/2007/02/08/a0169.1/textdruck [10 Feb 2007].
29. Matthew Richardson and Pedro Domingos. Mining Knowledge-Sharing Sites for Viral Marketing. In *Proc. of the Eighth Intl. Conf. on Knowledge Discovery and Data Mining (SIGKDD'02)*, 2002.
30. William Robinson, Suzanne Pawlowski, and Vecheslav Volkov. Requirements Interaction Management. *ACM Computing Surveys*, 35(2), 2003.
31. Joseph Smarr. Technical and privacy challenges for integrating FOAF into existing applications, 2001. http://www.w3.org/2001/sw/Europe/events/foaf-galway/papers/fp/technical_and_privacy_challenges/.
32. Ian Sommerville and Peter Sawyer. Viewpoints: Principles, Problems and a Practical Approach to Requirements Engineering. *Annals of Software Engineering*, 3:101–130, 1997.
33. Christian Stöcker. Community-Millionendeal: Holtzbrinck schnappt sich StudiVZ [Community Million-Euro Deal: Holtzbrinck snatches StudiVZ]. *Spiegel Online Netzwelt*, 2007. http://www.spiegel.de/netzwelt/web/0,1518,457536,00.html, 3 Jan 2007.
34. Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty*, 10(5):571–588, 2003.
35. Maximilian Teltzrow and Alfred Kobsa. Impacts of User Privacy Preferences on Personalized Systems: a Comparative Study. http://www.ics.uci.edu/~kobsa/papers/2004-PersUXinECom-kobsa.pdf.
36. Maximilian Teltzrow, Sören Preibusch, and Bettina Berendt. SIMT – A Privacy Preserving Web Metrics Tool. In *Proceedings of CEC*, pages 263–270. IEEE Computer Society, 2004.
37. Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*, volume 8 of *Structural Analysis in the Social Sciences*. Cambridge University Press, Cambridge, 1 edition, 1999.
38. D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.
39. Rigo Wenning and Matthias Schunter. The Platform for Privacy Preferences 1.1 (P3P1.1) Specification, 2006. W3C Working Group Note 13 November 2006, http://www.w3.org/TR/P3P11/.
40. Wikipedia. Viral Marketing, 2007. http://en.wikipedia.org/wiki/Viral_marketing [10 Feb 2007].

# Mining the Structure of Tag Spaces for User Modeling

Eric Schwarzkopf, Dominik Heckmann, Dietmar Dengler, and Alexander Kröner

German Research Center for AI (DFKI)
66123 Saarbruecken, Germany
`Firstname.Lastname@dfki.de`

**Abstract.** We propose an approach for using data from a social tagging application like `del.icio.us` as a basis for user adaptation. We discuss several algorithms for mining taxonomies of tags from tag spaces. The mined taxonomy can be used to define adaptation rules that determine how to adapt a system to a user given the user's personal tag space.

The contributions of this work are a description of an application scenario for taxonomy-mining algorithms, a discussion of algorithms by Mika[3], Heymann et al.[2], and Schmitz et al.[4], and the proposal of an extension to the algorithms that takes the contexts of tags into account when building a taxonomy. We look at the performances of the algorithms on a dataset retrieved from `del.icio.us` and give a tentative recommendation of what algorithm to use.

## 1 Introduction

Tag spaces are an obvious source of data for user modeling. The user of a social tagging tool could provide access to his personal tag space to an e-commerce site which could use the data to tailor its structure and presentation to the user. For example, a book store could determine that a customer who uses the tags *code*, *java*, and *mysql* frequently is most likely a programmer and recommend the most popular programming books.

How can we use a tag space and a user's tagging data to create a user model and adapt a system? The first step in the approach we are proposing is mining a taxonomy of tags from the tag space. The system engineer then creates a set of application-specific adaptation rules based on the mined taxonomy. Finally, a user's personal tags are mapped into the taxonomy to determine which adaptation rules apply to the user. This process is depecited in figure 1.

Not all tag spaces are suitable for this type of user modeling. Because we want to learn something about the user's interests, we require tagging data used by the user for himself (as in `del.ico.us`) and not for others (as in `flickr`).

In a taxonomy of tags, subtags of a tag are specializations of the tag (for example, *pop-music* should be a subtag of *music*). Given a taxonomy of tags, we can compute a value for the association between a user $u$ and a tag $t$ by computing the similarity between the set of tags used by $u$ and the set containing $t$ and all of $t$'s subtags by using, for example, Jaccard's coefficient. For the designer of an adaptive system a taxonomy simplifies identifying the semantics of a tag (by using its predecessors and successors as context) and its generality (the higher it is in the taxonomy, the more users will be associated with it). Hence, we think a taxonomy is a good underlying structure for the task at hand.

Mining a taxonomy from a tag space is the main subject of this paper. We will look at several taxnonomy-mining algorithms proposed in the literature, evaluate their performance on a dataset retrieved from `del.icio.us`, and, based on our results, give a tentative recommendation of what algorithm to use when mining taxonomies.

We do not focus on privacy issues in this paper, but since they are of special relevance in this domain, we provide some ideas that can be implemented easily on top of existing tagging systems. To limit the possibility of misuse, the user should restrict access to his data. To that end, a user could maintain several profiles, where a profile is a subset of the user's tagged resources. For instance, there could be a job profile for data collected in the user's professional role and a personal profile for data related to the user's hobbies. There are a number of ways to create profiles easily — and it has to be easy because otherwise the user

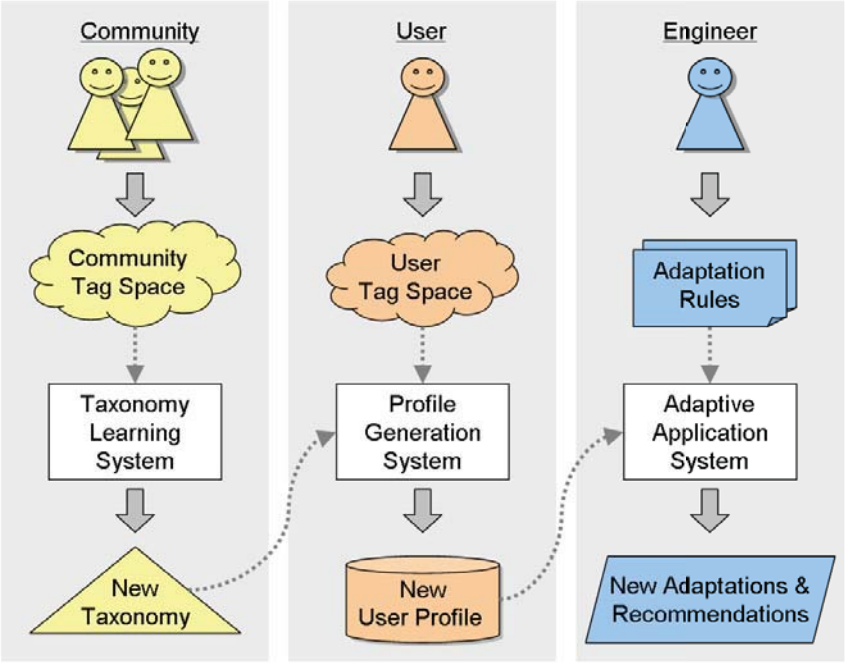


**Fig. 1.** Overview of the adaptation process.

could just as well fill in a questionnaire to create his profile. One is to associate a specific tag with a profile, so that all entries using the tag are automaticaly assigned to the profile. For each profile, the user should be able to create a snapshot in time, for example, resources tagged within the previous two weeks, and to provide only this snapshot to a third party system. Current tagging system only provide for an all or nothing decision: Once a third party knows the account name of the user, it can retrieve all current and future data in that account. But by using the APIs offered by most social tagging services, it is possible to create the described profile service even without modifying the existing services (for example, by building upon a user+tag RSS feed from `del.icio.us`).

## 2 Mining Taxonomies

### 2.1 Test Data

To gain some experience with the algorithms discussed below, we applied them to data retrieved from `del.icio.us`. We collected 2553 account names by periodically polling the RSS feed of recent additions and then downloaded for each account the 100 most recently added bookmarks with the associated tags, resulting in a set of 125801 distinct bookmarks, 158870 user-bookmark pairs (most users had collected less than 100 bookmarks), 23334 shared bookmarks (bookmarks collected by more than one user), and 37697 distinct tags.

The approach taken to sampling accounts resulted in a set of users with diverse interests but low overlap in annotated bookmarks. These characteristics have to be taken into account when interpreting the results of our experiments.

### 2.2 Estimating the Quality of a Taxonomy

In order to compare the performance of the taxonomy-mining algorithms, we need some kind of measure of the quality of the mined taxonomies. For this paper, we use the structure of a taxonomy to assess their quality: Given the characteristics of our test data, we assume that a taxonomy is of low quality if its maximum depth is larger than 5 or most of the tags are on the first level of the taxonomy. A depth larger than 5 indicates that a large number of users have a common, very specific interest, while a flat taxonomy indicates that there are only very few subsumption relationships between tags. Both conditions are very unlikely for our dataset, and manual inspection of some of the mined taxonomies confirmed that an average depth of 3 with most of the tags on levels 1 and 2 is to be expected of a high-quality taxonomy.

The appendix lists the structure of taxonomies generated by the discussed algorithms with different parameter settings.

### 2.3 Algorithms

Before talking about algorithms for learning taxonomies, we define formally what we mean by a tag space. The following definition is adopted from Mika [3] and is used throughout the subsequent discussion.

> A **tag space** is a hypergraph $H := < V, E >$. The set of vertices $V$ is the union of three disjoint sets $A$, $C$, and $I$ representing the set of users (actors), the set of tags (concepts), and the set of annotated objects (items). $E$ is the set of ternary edges $\{\{a, c, i\} \,|\, \text{user}\, a\, \text{labels object}\, i\, \text{with tag}\, c\}$.

Our intention is to discover subsumption relationships between tags as seen by the user community. We say a tag $t$ subsumes a tag $u$ if and only if the intension of $t$ properly contains the intension of $u$. That is, $t$ subsumes $u$ if all imaginable objects that could be sensibly tagged with $u$ can also be sensibly tagged with $t$.[1]

Mika [3] looks at the weighted graphs $O_{ac} := < C, E_{ac}, w_{ac} >$, where $C$ is the set of tags, $E_{ac} := \{(x, y) \,|\, x, y \in C, \exists a \in A\, \exists i, j \in I : \{a, x, i\}, \{a, y, j\} \in E\}$, $w_{ac}((x, y)) := |\{a \in A \,|\, \exists i, j \in I : \{a, x, i\}, \{a, y, j\} \in E\}|$, and $O_{ic} := < C, E_{ic}, w_{ic} >$, where $E_{ic} := \{(x, y) \,|\, x, y \in C, \exists i \in I\, \exists a, b \in A : \{a, x, i\}, \{b, y, i\} \in E\}$ and $w_{ic}((x, y)) := |\{i \in I \,|\, \exists a, b \in A : \{a, x, i\}, \{b, y, i\} \in E\}|$. That is, $O_{ac}$ is the graph of tags in which an edge between two tags is weighted by the number of people who have used both tags, while $O_{ic}$ is the graph of tags in which an edge is weighted by the number of resources that have been tagged with both tags. Mika suggests using $O_{ic}$ for concept mining by applying graph clustering; he reports mining cohesive groups of concepts from `del.icio.us` data using $\alpha$-set analysis. Because $O_{ic}$ does not reflect how popular tags are in the user community (local structure can be determined by very few users), he uses $O_{ac}$ to discover taxonomic relationships between tags using a set-theoretic approach that corresponds to mining association rules as is described further below when we look at the approach proposed by Schmitz et al.[4]. The idea is that the community of users associated with a narrower tag is a sub-community of the community associated with the broader tag.

Heymann at al.[2] create for each tag $t$ a vector representation $v_t := (w(t, i))_{i \in I}$, where $w((c, i)) := |\{a \,|\, (a, c, i) \in E\}|$, and then define a tag similarity graph $S := < C, E_s >$, where $E_s := \{(a, b) \,|\, a, b \in C, \cos(v_a, v_b) > d\}$ with cos denoting the cosine similarity and $d$ a predefined threshold. Note that in general, $S$ does not correspond to Mika's $O_{ic}$, because the latter uses the overlap in tagged resources to determine the weight of an edge while the cosine similarity measures how similar the distribtions of tags are over all resources.

To create a taxonomy of tags, they first sort the tags in non-increasing order of their closeness-centrality in the similarity graph. Closeness-centrality of a node $n_i$ is defined as the inverse of the total distance that $n_i$ is from all other nodes: $(\sum_{j=1}^{g} d(n_i, n_j))^{-1}$, where $d$ is the geodesic distance between $n_i$ and $n_j$.[2] They then start with an empty taxonomy,

[1] We assume here that there is a one-to-one correspondence between semantic concepts and tags, which is, as Mika points out, incorrect.

[2] Why a centrality measure for identifying general tags and not a more efficiently computed, local measure such as the degree of a node? A node far down in the taxonomy can have a large local connectivity (for example, a node pointing to a large number of leafs), but its centrality will be low because the distance to most of the nodes in the taxonomy will be high.

which contains only a root node not associated with a tag, and add each tag in turn starting from the most central. A tag is added as a child of either the tag it is most similar to if the similarity is above a threshold or the root node.

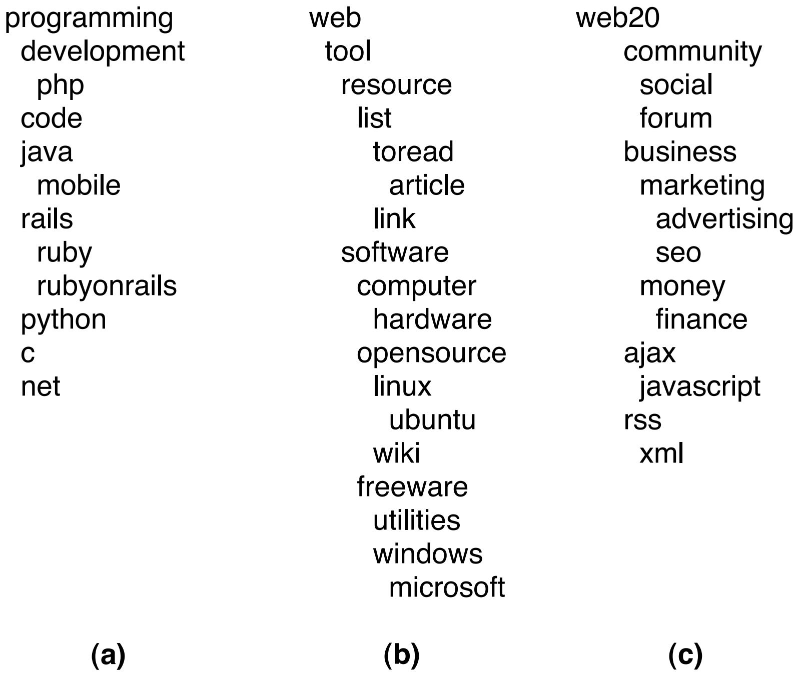


**Fig. 2.** Parts of a taxonomy created using the algorithm by Heymann et al. on the test data. Trees (b) and (c) suggest problems with contextual similarity: `ubuntu` is related to `linux`, but it is not obvious why both are subtags of `web`.

Heymann et al. base their approach on three assumptions: (1) the relationships in the taxonomy also exist in the similarity graph, (2) there are noisy connections between tags that have no matching connection in the taxonomy (hence, the edges in the similarity graph are a superset of the edges in the taxonomy), and (3) noisy connections are more common higher up in the hierarchy (that is, for more general tags).

Their algorithm further assumes that all tags are part of a taxonomy. This is a simplifying assumption because in general only a subset of tags is used for denoting the categories of resources [1]. Furthermore, the algorithm does not take the context of a parent tag into account when adding a child: The similarity between a tag and its potential parent in the taxonomy does not depend on the ancestors of the parent. This results in chains such as *design→web→howto→productivity→business* in which each link seems to make sense but the complete chain does not.

Applied to our test data, this context agnostic assignment results in a poorly structured taxonomy: Depending on how the similarity thresholds are chosen, the taxonomy is either too flat, with a large number of tags not having any children, or a single tag being the root of a deep tree containing most of the other tags, with a large number of tag chains not making sense (see figure 2).

A simple way to take the context into account is to require from a tag that is has a certain minimum average similarity to all predecessors of the parent. We add this test to the original algorithm after the tag $p$ in the taxonomy that is most similar to the tag $t$ to be inserted is determined. If the average similarity between $t$ and the predecessors of $p$ is below a given threshold, a copy $p_c$ of $p$ is added as a new top-level node to the taxonomy and $t$ is made a child of $p_c$.

Applied to our test data, the extended algorithm leads to taxonomies without intuitively incorrect chains, but the overall structure is in general too flat (see figure 3).

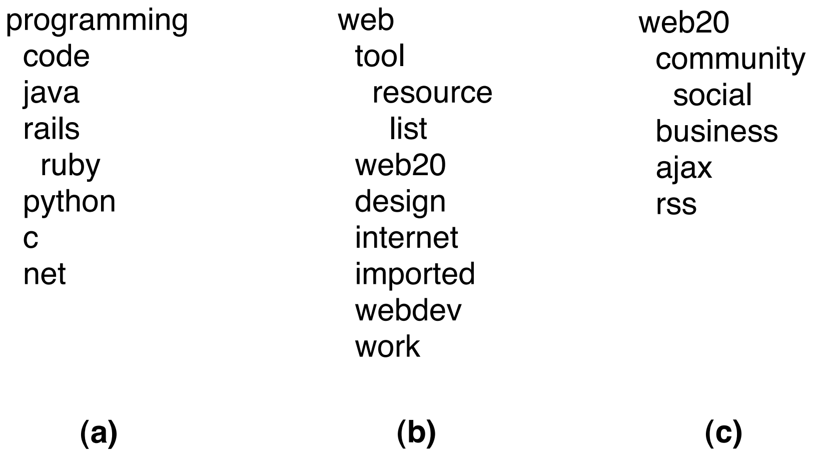


**Fig. 3.** Parts of a taxonomy created using Heymann's algorithm modified to take the context of tags into account.

Another approach is proposed by Schmitz et al.[4]. They mine from a tag space association rules of the form *If users assign the tags from X to some resource, they often also assign the tags from Y to them.* If resources tagged with $t_0$ are often also tagged with $t_1$ but a large number of resources tagged with $t_1$ are not tagged with $t_0$, $t_1$ can be considered to subsume $t_0$.

Formally, Schmitz et al. learn association rules over the set $T := \{\{i \,|\, \{a, c, i\} \in E\} \,|\, a \in A, c \in C\}$. Here, an association rule is a tuple in $2^I \setminus \emptyset \times 2^I \setminus \emptyset$. Whether an association rule $(X, Y)$ is of interest or not can be determined by thresholds on its support $\mathrm{supp}(X, Y) := |\{U \in T \,|\, X \subset U, Y \subset U\}| \,/\, |T|$ and its confidence $\mathrm{conf}(X, Y) := |\{U \in T \,|\, X \subset U, Y \subset U\}| \,/\, |\{U \in T \,|\, X \subset U\}|$.

The cosine similarity measure used by Heymann et al. does not take into account the total count of occurences of a tag because tag vectors are being normalized. For instance, the vector $(1, 2, 3)$ is more similar to $(100, 180, 250)$ than $(100, 180, 250)$ is to $(50, 150, 200)$, although intuitively the latter pair should be more similar in respect to the corresponding tags' positions in the taxonomy.

In contrast, association rules reflect the frequencies of subsets. Assume we have got tags $t_1$, $t_2$, $t_3$ with the same distributions as used in the discussion of cosine similarity $v_{t_1} = (1, 2, 3)$, $v_{t_2} = (100, 180, 250)$, and $v_{t_3} = (50, 150, 200)$, and that the resources tagged with $t_1$ and $t_3$ are proper subsets of the resources tagged with $t_2$. Then $\mathrm{conf}((t_2, t_1)) = 6 \,/\, 530$ and $\mathrm{conf}((t_2, t_3)) = 400 \,/\, 530$. Hence, there is a stronger relationship between $t_2$ and $t_3$ than $t_2$ and $t_1$.

Schmitz et al. do not describe how a taxonomy can be created from the mined rules. One possible approach is the following: Given the set $R$ of interesting association rules in $I \times I$ (that is, associations between single tags), we can define a graph $\mathrm{AR} :=< Var, E_{ar}, w_{ar} >$, where $V_{ar} := \{i \in I \,|\, \exists j \in I : (i, j) \in R \,\mathrm{or}\, (j, i) \in R\}$, $E_{ar} := \{(x, y) \,|\, (y, x) \in R\}$, $w_{ar}((x, y)) := \mathrm{conf}((y, x))$. We then create the graph $\mathrm{AR}'$ by keeping for each node $y$ only the incoming edge $(x, y)$ with the strongest weight $w_{ar}((x, y))$. $\mathrm{AR}'$ is a forest, and a single tree (the taxonomy) can be created by introducing a new node $r$ (the root of the taxonomy) and connecting it to all existing root nodes in the forest.

This algorithm, like the algorithm by Heymann et al., assumes that there is a one-to-one correspondence between tags and concepts, does not attempt to distinguish the different uses of tags, and ignores the context of a tag in the taxonomy.

Overall, we get better results on our test data with association rules than with Heymann's algorithm, but we observe similar issues in respect to the context of tags (see 4).

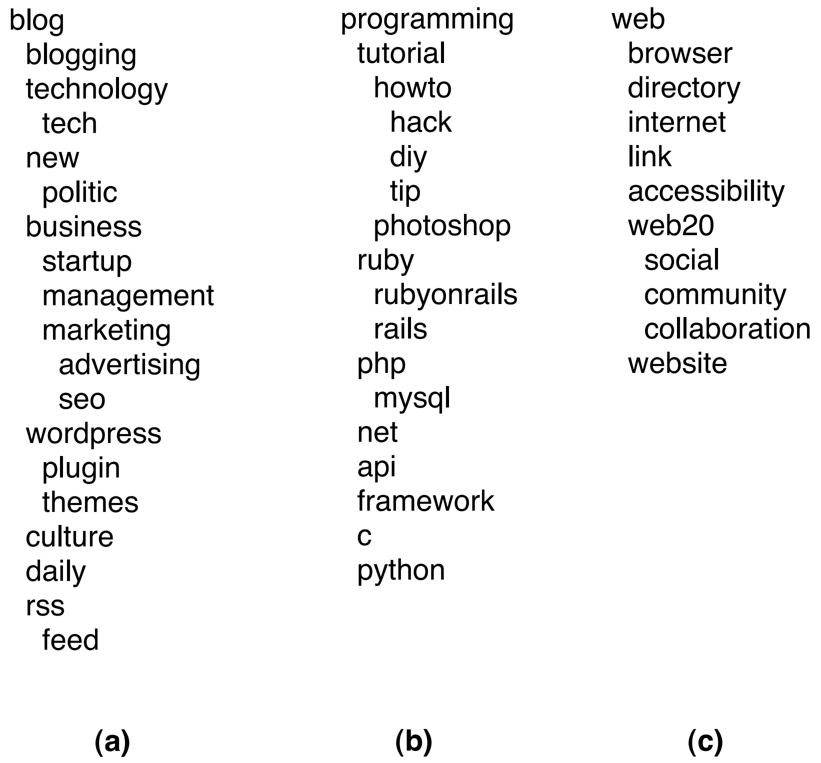
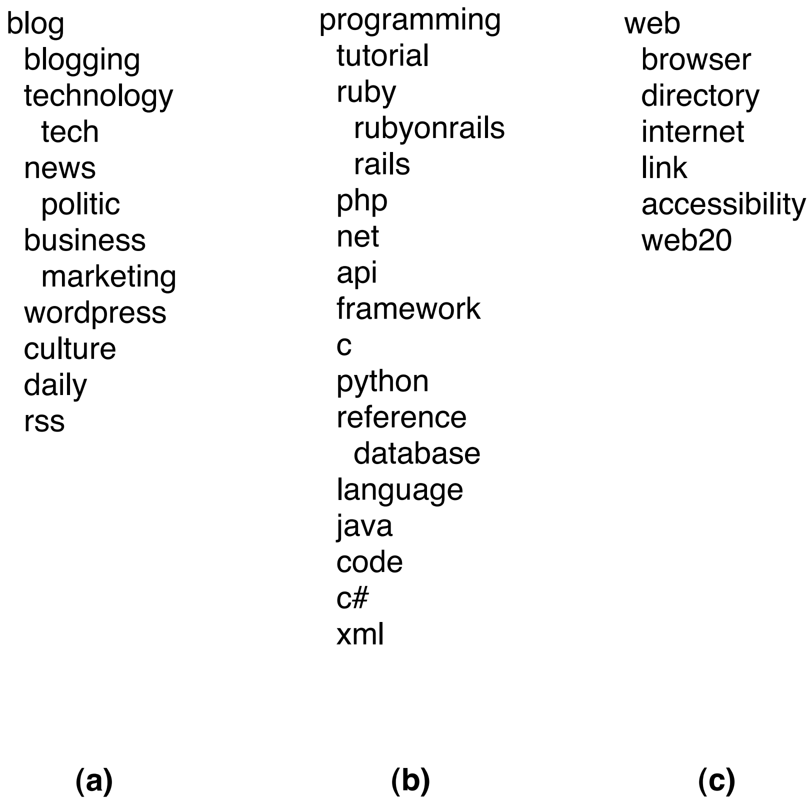


**Fig. 4.** Parts of a taxonomy created using the assocition-rule algorithm.

We can extend the algorithm to take the context into account by requiring that there is an edge from a tag to all of its subtags in AR. For each root in AR′, we traverse the corresponding tree. If we reach a tag that is not adjacent to the root in AR, we make a copy of its parent in AR′ a new root and continue traversal. This corresponds to the modification we made to Heymann's algorithm.

### 2.4 A Tentative Recommendation

For our set of test data and implementations, the association-rule algorithm and its extension generated better taxonomies than Heymann's algorithm for a relatively wide range of parameters. The appendix lists structural features of taxonomies mined by the algorithms using several sets of parameter values. The 'better' structural features (see above) of the association-rule algorithms further support the impression we got from manual inspection of a sample of taxonomies.



- blog
  - blogging
  - technology
    - tech
  - news
    - politic
  - business
    - marketing
  - wordpress
  - culture
  - daily
  - rss

(a)

- programming
  - tutorial
  - ruby
    - rubyonrails
    - rails
  - php
  - net
  - api
  - framework
  - c
  - python
  - reference
    - database
  - language
  - java
  - code
  - c#
  - xml

(b)

- web
  - browser
  - directory
  - internet
  - link
  - accessibility
  - web20

(c)

**Fig. 5.** Parts of a taxonomy created using the assocition-rule algorithm modified to take the context of tags into account.

## 3 Mapping User Interests

Given a taxonomy of tags, we need to identify which of those tags best describe the interests of a user so we can apply the appropriate adaptation rules. We do this by computing the similarity between sets of tags: We represent a tag of the taxonomy by the set containing the tag and all of its subtags. Jaccard's similarity coefficient, defined as the ratio between the size of the intersection of two sets and the union of those sets, $|A \cap B| \,/\, |A \cup B|$, is used to determine how strongly the user, represented by the intersection of his tags with the set of all tags of the taxonomy, is associated with a specific tag of the taxonomy.

For example, one user in our dataset uses the following tags:

*wiki code firefox cc blogger mysql hardware webstv wordpress own search mp3 linux css nba hacker zooomr bloglines java network service bittorrent bookmark vbscript lyrics perl blog teacher book crack teaching net irc homepage album asp assembly dictionary clubbox web20 tool javascript notepad yahoo wargames diagram xml game lifelype proxy regexp translation php ruby security foobar2000 decompiler p2p audio embedded forum database mobile eclipse html server bbs fju freebsd encryption movie sniffer ide maplebbs portalsite pda software*

If we map this set to a taxonomy mined using the association-rule algorithm, we learn that the user is strongly associated with the concepts *programming*, *security*, and *software*. Note that an interest for *programming* is not explicit in the user's tags, but inferred using the mined taxonomy.

## 4 Things We Ignored

The presented approaches to mining taxonomies from tag spaces ignore a number of issues relevant to our application domain:

**dynamics:** How tags are used changes in time. For example, a tag specialization might be introduced that describes a subset of the tagged resources better and thus replaces a more broader tag, or a tag for an entirely new and popular concept might be introduced. This dynamics will affect adversly the quality of the mapping of user interests to the then out-of-date tag taxonomy. An ideal system would adapt the taxonomy to any changes so manual maintenance of the taxonomy or the adaptation rules would not be necessary.

**not a 1-to-1 mapping between tags and concepts:** As Mika points out, a concept might be represented not by a single but by a set of several tags. This suggests that the quality of the learned structure of the tag space can be improved if the simplifying assumption of a 1-to-1 mapping is dropped.

**different uses of tags:** Tags can be used in different functions. In addition to denoting categories and subcategories, they can describe the content, type, and owner of a resource, the opinion of the tagger, what to do with a resource ("toread"), or the relation to the tagger ("mycomments") (see Golder et al.[1]). Distinguishing between the different functions should improve the learned structure of the tag space.

**polysemy, synonymy:** All presented approaches ignore polysemy and synonymy in the tag space. This leads to a reduction in quality of the learned structure because the quantitative relationships between concepts are misrepresented.

## 5 Conclusion

We have begun exploring an application scenario in which data from a tag space is used to adapt a system to an individual user. Algorithms and approaches in this domain are still in their infancy, and with lots of relevant data available on the web and its potential usefulness (not only in the mentioned e-commerce scenario), we see it as a promising area of research.

## References

1. Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32(2):198–208, April 2006.
2. Paul Heymann and Hector Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Stanford University, April 2006.
3. Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 522–536. International Semantic Web Conference 2005, Springer, November 2005.
4. Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. iberna, editors, *Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270, Berlin, Heidelberg, 2006. Springer.

# A Structural Features of Mined Taxonomies

| edge sim | parent sim | #taxa | max depth | #lvl1 | #lvl2 | #lvl3 | #lvl4 | #lvl5 |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 0.05 | 0.02 | 489 | 13 | 2 | 16 | 41 | 76 | 98 |
| 0.05 | 0.04 | 489 | 13 | 2 | 16 | 41 | 76 | 98 |
| 0.05 | 0.06 | 489 | 13 | 16 | 22 | 44 | 76 | 95 |
| 0.05 | 0.08 | 489 | 11 | 40 | 25 | 44 | 75 | 91 |
| 0.05 | 0.10 | 489 | 11 | 85 | 33 | 40 | 71 | 83 |
| 0.05 | 0.12 | 489 | 10 | 134 | 59 | 51 | 72 | 62 |
| 0.05 | 0.14 | 489 | 10 | 178 | 67 | 51 | 59 | 48 |
| 0.05 | 0.16 | 489 | 9 | 209 | 72 | 62 | 57 | 35 |
| 0.05 | 0.18 | 489 | 8 | 238 | 79 | 61 | 46 | 29 |
| 0.05 | 0.20 | 489 | 8 | 265 | 90 | 57 | 39 | 18 |
| 0.05 | 0.22 | 489 | 8 | 293 | 89 | 50 | 31 | 15 |
| 0.05 | 0.24 | 489 | 5 | 324 | 88 | 48 | 20 | 9 |
| 0.05 | 0.26 | 489 | 5 | 349 | 75 | 39 | 19 | 7 |
| 0.05 | 0.28 | 489 | 5 | 367 | 69 | 33 | 16 | 4 |
| 0.05 | 0.30 | 489 | 5 | 386 | 67 | 29 | 6 | 1 |
| 0.05 | 0.32 | 489 | 4 | 402 | 60 | 23 | 4 | 0 |
| 0.05 | 0.34 | 489 | 4 | 414 | 55 | 18 | 2 | 0 |
| 0.05 | 0.36 | 489 | 3 | 424 | 48 | 17 | 0 | 0 |
| 0.05 | 0.38 | 489 | 3 | 430 | 46 | 13 | 0 | 0 |
| 0.05 | 0.40 | 489 | 3 | 441 | 38 | 10 | 0 | 0 |
| 0.1 | 0.02 | 427 | 15 | 8 | 11 | 10 | 11 | 23 |
| 0.1 | 0.04 | 427 | 13 | 13 | 21 | 29 | 30 | 47 |
| 0.1 | 0.06 | 427 | 13 | 13 | 21 | 29 | 30 | 47 |
| 0.1 | 0.08 | 427 | 13 | 13 | 21 | 29 | 30 | 47 |
| 0.1 | 0.10 | 427 | 13 | 13 | 21 | 29 | 30 | 47 |
| 0.1 | 0.12 | 427 | 11 | 65 | 43 | 42 | 41 | 53 |
| 0.1 | 0.14 | 427 | 11 | 115 | 62 | 38 | 32 | 44 |
| 0.1 | 0.16 | 427 | 11 | 147 | 63 | 50 | 28 | 29 |
| 0.1 | 0.18 | 427 | 11 | 175 | 65 | 47 | 24 | 27 |
| 0.1 | 0.20 | 427 | 11 | 201 | 82 | 45 | 19 | 20 |
| 0.1 | 0.22 | 427 | 8 | 232 | 86 | 46 | 17 | 16 |
| 0.1 | 0.24 | 427 | 8 | 261 | 88 | 38 | 15 | 10 |
| 0.1 | 0.26 | 427 | 8 | 285 | 77 | 31 | 11 | 9 |
| 0.1 | 0.28 | 427 | 8 | 306 | 71 | 24 | 8 | 8 |
| 0.1 | 0.30 | 427 | 7 | 326 | 70 | 20 | 5 | 3 |
| 0.1 | 0.32 | 427 | 7 | 339 | 62 | 17 | 4 | 2 |
| 0.1 | 0.34 | 427 | 6 | 351 | 55 | 14 | 4 | 2 |
| 0.1 | 0.36 | 427 | 5 | 362 | 47 | 15 | 2 | 1 |
| 0.1 | 0.38 | 427 | 5 | 368 | 44 | 12 | 2 | 1 |
| 0.1 | 0.40 | 427 | 4 | 378 | 38 | 10 | 1 | 0 |

**Table 1.** Results of Heymann's algorithm applied to the test data. `edge sim` is the minimum similarity required for an edge to be created between two tags in the similarity graph. `parent sim` is the minimum similarity required for a tag to become the child of a taxon.

| edge sim | parent sim | context sim | #taxa | max depth | #lvl1 | #lvl2 | #lvl3 | #lvl4 | #lvl5 |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 0.05 | 0.02 | 0.02 | 601 | 10 | 114 | 186 | 73 | 63 | 63 |
| 0.05 | 0.02 | 0.04 | 655 | 8 | 168 | 301 | 71 | 49 | 33 |
| 0.05 | 0.02 | 0.06 | 685 | 7 | 198 | 377 | 61 | 34 | 10 |
| 0.05 | 0.02 | 0.08 | 700 | 6 | 213 | 412 | 47 | 21 | 6 |
| 0.05 | 0.02 | 0.10 | 709 | 4 | 222 | 439 | 38 | 10 | 0 |
| 0.05 | 0.02 | 0.12 | 718 | 4 | 231 | 457 | 25 | 5 | 0 |
| 0.05 | 0.02 | 0.14 | 719 | 4 | 232 | 469 | 15 | 3 | 0 |
| 0.05 | 0.02 | 0.16 | 720 | 3 | 233 | 476 | 11 | 0 | 0 |
| 0.05 | 0.02 | 0.18 | 721 | 3 | 234 | 481 | 6 | 0 | 0 |
| 0.05 | 0.02 | 0.20 | 722 | 3 | 235 | 483 | 4 | 0 | 0 |
| 0.05 | 0.04 | 0.02 | 601 | 10 | 114 | 186 | 73 | 63 | 63 |
| 0.05 | 0.04 | 0.04 | 655 | 8 | 168 | 301 | 71 | 49 | 33 |
| 0.05 | 0.04 | 0.06 | 685 | 7 | 198 | 377 | 61 | 34 | 10 |
| 0.05 | 0.04 | 0.08 | 700 | 6 | 213 | 412 | 47 | 21 | 6 |
| 0.05 | 0.04 | 0.10 | 709 | 4 | 222 | 439 | 38 | 10 | 0 |
| 0.05 | 0.04 | 0.12 | 718 | 4 | 231 | 457 | 25 | 5 | 0 |
| 0.05 | 0.04 | 0.14 | 719 | 4 | 232 | 469 | 15 | 3 | 0 |
| 0.05 | 0.04 | 0.16 | 720 | 3 | 233 | 476 | 11 | 0 | 0 |
| 0.05 | 0.04 | 0.18 | 721 | 3 | 234 | 481 | 6 | 0 | 0 |
| 0.05 | 0.04 | 0.20 | 722 | 3 | 235 | 483 | 4 | 0 | 0 |
| 0.05 | 0.08 | 0.02 | 577 | 10 | 128 | 154 | 70 | 62 | 62 |
| 0.05 | 0.08 | 0.04 | 629 | 8 | 180 | 265 | 70 | 49 | 32 |
| 0.05 | 0.08 | 0.06 | 661 | 7 | 212 | 339 | 61 | 34 | 10 |
| 0.05 | 0.08 | 0.08 | 676 | 6 | 227 | 374 | 47 | 21 | 6 |
| 0.05 | 0.08 | 0.10 | 685 | 4 | 236 | 401 | 38 | 10 | 0 |
| 0.05 | 0.08 | 0.12 | 694 | 4 | 245 | 419 | 25 | 5 | 0 |
| 0.05 | 0.08 | 0.14 | 696 | 4 | 247 | 431 | 15 | 3 | 0 |
| 0.05 | 0.08 | 0.16 | 698 | 3 | 249 | 438 | 11 | 0 | 0 |
| 0.05 | 0.08 | 0.18 | 699 | 3 | 250 | 443 | 6 | 0 | 0 |
| 0.05 | 0.08 | 0.20 | 700 | 3 | 251 | 445 | 4 | 0 | 0 |
| 0.05 | 0.16 | 0.02 | 513 | 9 | 233 | 101 | 59 | 48 | 32 |
| 0.05 | 0.16 | 0.04 | 540 | 8 | 260 | 139 | 58 | 42 | 17 |
| 0.05 | 0.16 | 0.06 | 558 | 6 | 278 | 178 | 57 | 35 | 6 |
| 0.05 | 0.16 | 0.08 | 573 | 6 | 293 | 204 | 48 | 20 | 5 |
| 0.05 | 0.16 | 0.10 | 587 | 5 | 307 | 229 | 36 | 13 | 2 |
| 0.05 | 0.16 | 0.12 | 597 | 4 | 317 | 249 | 24 | 7 | 0 |
| 0.05 | 0.16 | 0.14 | 599 | 4 | 319 | 257 | 17 | 6 | 0 |
| 0.05 | 0.16 | 0.16 | 605 | 3 | 325 | 269 | 11 | 0 | 0 |
| 0.05 | 0.16 | 0.18 | 607 | 3 | 327 | 274 | 6 | 0 | 0 |
| 0.05 | 0.16 | 0.20 | 608 | 3 | 328 | 276 | 4 | 0 | 0 |

**Table 2.** Results of the extension of Heymann's algorithm applied to the test data. edge sim is the minimum similarity required for an edge to be created between two tags in the similarity graph. parent sim is the minimum similarity required for a tag to become the child of a taxon.

| edge sim | parent sim | context sim | #taxa | max depth | #lvl1 | #lvl2 | #lvl3 | #lvl4 | #lvl5 |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 0.1 | 0.02 | 0.02 | 506 | 13 | 87 | 129 | 61 | 26 | 34 |
| 0.1 | 0.02 | 0.04 | 554 | 9 | 135 | 217 | 61 | 27 | 28 |
| 0.1 | 0.02 | 0.06 | 583 | 9 | 164 | 284 | 63 | 21 | 13 |
| 0.1 | 0.02 | 0.08 | 603 | 8 | 184 | 343 | 41 | 15 | 5 |
| 0.1 | 0.02 | 0.10 | 613 | 8 | 194 | 370 | 26 | 10 | 5 |
| 0.1 | 0.02 | 0.12 | 618 | 8 | 199 | 390 | 19 | 3 | 2 |
| 0.1 | 0.02 | 0.14 | 622 | 6 | 203 | 400 | 12 | 5 | 1 |
| 0.1 | 0.02 | 0.16 | 625 | 5 | 206 | 409 | 6 | 3 | 1 |
| 0.1 | 0.02 | 0.18 | 625 | 5 | 206 | 411 | 5 | 2 | 1 |
| 0.1 | 0.02 | 0.20 | 625 | 5 | 206 | 412 | 4 | 2 | 1 |
| 0.1 | 0.04 | 0.02 | 500 | 13 | 86 | 123 | 61 | 26 | 34 |
| 0.1 | 0.04 | 0.04 | 548 | 9 | 134 | 211 | 60 | 27 | 28 |
| 0.1 | 0.04 | 0.06 | 577 | 9 | 163 | 279 | 63 | 21 | 13 |
| 0.1 | 0.04 | 0.08 | 597 | 8 | 183 | 339 | 42 | 13 | 5 |
| 0.1 | 0.04 | 0.10 | 607 | 8 | 193 | 366 | 27 | 8 | 5 |
| 0.1 | 0.04 | 0.12 | 612 | 8 | 198 | 385 | 19 | 3 | 2 |
| 0.1 | 0.04 | 0.14 | 616 | 6 | 202 | 395 | 12 | 5 | 1 |
| 0.1 | 0.04 | 0.16 | 619 | 5 | 205 | 404 | 6 | 3 | 1 |
| 0.1 | 0.04 | 0.18 | 619 | 5 | 205 | 406 | 5 | 2 | 1 |
| 0.1 | 0.04 | 0.20 | 619 | 5 | 205 | 407 | 4 | 2 | 1 |
| 0.1 | 0.08 | 0.02 | 500 | 13 | 86 | 123 | 61 | 26 | 34 |
| 0.1 | 0.08 | 0.04 | 548 | 9 | 134 | 211 | 60 | 27 | 28 |
| 0.1 | 0.08 | 0.06 | 577 | 9 | 163 | 279 | 63 | 21 | 13 |
| 0.1 | 0.08 | 0.08 | 597 | 8 | 183 | 339 | 42 | 13 | 5 |
| 0.1 | 0.08 | 0.10 | 607 | 8 | 193 | 366 | 27 | 8 | 5 |
| 0.1 | 0.08 | 0.12 | 612 | 8 | 198 | 385 | 19 | 3 | 2 |
| 0.1 | 0.08 | 0.14 | 616 | 6 | 202 | 395 | 12 | 5 | 1 |
| 0.1 | 0.08 | 0.16 | 619 | 5 | 205 | 404 | 6 | 3 | 1 |
| 0.1 | 0.08 | 0.18 | 619 | 5 | 205 | 406 | 5 | 2 | 1 |
| 0.1 | 0.08 | 0.20 | 619 | 5 | 205 | 407 | 4 | 2 | 1 |
| 0.1 | 0.16 | 0.02 | 453 | 11 | 173 | 89 | 61 | 29 | 25 |
| 0.1 | 0.16 | 0.04 | 474 | 9 | 194 | 122 | 62 | 26 | 20 |
| 0.1 | 0.16 | 0.06 | 496 | 8 | 216 | 159 | 61 | 20 | 10 |
| 0.1 | 0.16 | 0.08 | 512 | 8 | 232 | 205 | 46 | 10 | 5 |
| 0.1 | 0.16 | 0.10 | 526 | 8 | 246 | 233 | 27 | 8 | 4 |
| 0.1 | 0.16 | 0.12 | 533 | 8 | 253 | 251 | 19 | 3 | 2 |
| 0.1 | 0.16 | 0.14 | 538 | 6 | 258 | 261 | 12 | 5 | 1 |
| 0.1 | 0.16 | 0.16 | 543 | 5 | 263 | 270 | 6 | 3 | 1 |
| 0.1 | 0.16 | 0.18 | 544 | 5 | 264 | 272 | 5 | 2 | 1 |
| 0.1 | 0.16 | 0.20 | 544 | 5 | 264 | 273 | 4 | 2 | 1 |

**Table 3.** Results of the extension of Heymann's algorithm applied to the test data. edge sim is the minimum similarity required for an edge to be created between two tags in the similarity graph. parent sim is the minimum similarity required for a tag to become the child of a taxon.

| confidence | support | #taxa | max depth | #lvl1 | #lvl2 | #lvl3 | #lvl4 | #lvl5 |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 0.05 | 0.0001 | 1071 | 8 | 57 | 129 | 283 | 285 | 182 |
| 0.05 | 0.0002 | 502 | 8 | 19 | 59 | 172 | 132 | 86 |
| 0.05 | 0.0003 | 344 | 8 | 13 | 44 | 125 | 97 | 47 |
| 0.05 | 0.0004 | 258 | 6 | 11 | 42 | 96 | 70 | 30 |
| 0.05 | 0.0005 | 211 | 6 | 12 | 37 | 77 | 58 | 21 |
| 0.05 | 0.0006 | 162 | 6 | 7 | 29 | 63 | 42 | 16 |
| 0.05 | 0.0007 | 133 | 6 | 7 | 27 | 50 | 32 | 12 |
| 0.05 | 0.0008 | 114 | 6 | 5 | 23 | 44 | 26 | 11 |
| 0.05 | 0.0009 | 106 | 6 | 7 | 26 | 41 | 20 | 9 |
| 0.05 | 0.0010 | 93 | 6 | 9 | 26 | 31 | 17 | 8 |
| 0.10 | 0.0001 | 1067 | 7 | 80 | 323 | 331 | 199 | 105 |
| 0.10 | 0.0002 | 499 | 7 | 33 | 160 | 173 | 86 | 40 |
| 0.10 | 0.0003 | 339 | 7 | 24 | 112 | 120 | 58 | 22 |
| 0.10 | 0.0004 | 255 | 6 | 21 | 89 | 89 | 41 | 14 |
| 0.10 | 0.0005 | 209 | 5 | 20 | 75 | 72 | 32 | 10 |
| 0.10 | 0.0006 | 157 | 5 | 12 | 58 | 53 | 26 | 8 |
| 0.10 | 0.0007 | 130 | 5 | 12 | 52 | 40 | 18 | 8 |
| 0.10 | 0.0008 | 113 | 5 | 10 | 48 | 31 | 17 | 7 |
| 0.10 | 0.0009 | 105 | 5 | 10 | 47 | 27 | 15 | 6 |
| 0.10 | 0.0010 | 92 | 5 | 12 | 41 | 22 | 13 | 4 |
| 0.15 | 0.0001 | 1037 | 6 | 115 | 482 | 304 | 98 | 34 |
| 0.15 | 0.0002 | 482 | 5 | 54 | 245 | 134 | 34 | 15 |
| 0.15 | 0.0003 | 327 | 5 | 36 | 170 | 94 | 18 | 9 |
| 0.15 | 0.0004 | 246 | 5 | 31 | 129 | 69 | 13 | 4 |
| 0.15 | 0.0005 | 201 | 5 | 29 | 103 | 57 | 10 | 2 |
| 0.15 | 0.0006 | 154 | 5 | 20 | 81 | 42 | 9 | 2 |
| 0.15 | 0.0007 | 127 | 5 | 18 | 67 | 32 | 8 | 2 |
| 0.15 | 0.0008 | 108 | 5 | 14 | 57 | 27 | 8 | 2 |
| 0.15 | 0.0009 | 101 | 5 | 14 | 56 | 22 | 8 | 1 |
| 0.15 | 0.0010 | 89 | 5 | 16 | 50 | 17 | 5 | 1 |
| 0.20 | 0.0001 | 988 | 6 | 149 | 576 | 196 | 56 | 10 |
| 0.20 | 0.0002 | 457 | 5 | 75 | 274 | 85 | 20 | 3 |
| 0.20 | 0.0003 | 309 | 5 | 52 | 186 | 58 | 10 | 3 |
| 0.20 | 0.0004 | 227 | 4 | 41 | 134 | 45 | 7 | 0 |
| 0.20 | 0.0005 | 185 | 4 | 38 | 108 | 33 | 6 | 0 |
| 0.20 | 0.0006 | 140 | 4 | 28 | 82 | 25 | 5 | 0 |
| 0.20 | 0.0007 | 114 | 4 | 23 | 66 | 21 | 4 | 0 |
| 0.20 | 0.0008 | 97 | 4 | 19 | 56 | 18 | 4 | 0 |
| 0.20 | 0.0009 | 88 | 4 | 16 | 51 | 17 | 4 | 0 |
| 0.20 | 0.0010 | 79 | 4 | 18 | 46 | 13 | 2 | 0 |
| 0.25 | 0.0001 | 903 | 4 | 170 | 564 | 131 | 34 | 4 |
| 0.25 | 0.0002 | 403 | 4 | 83 | 256 | 52 | 12 | 0 |
| 0.25 | 0.0003 | 267 | 4 | 57 | 168 | 37 | 5 | 0 |
| 0.25 | 0.0004 | 196 | 4 | 44 | 121 | 27 | 4 | 0 |
| 0.25 | 0.0005 | 159 | 4 | 40 | 97 | 19 | 3 | 0 |
| 0.25 | 0.0006 | 120 | 4 | 30 | 74 | 13 | 3 | 0 |
| 0.25 | 0.0007 | 94 | 4 | 24 | 58 | 10 | 2 | 0 |
| 0.25 | 0.0008 | 78 | 4 | 19 | 48 | 9 | 2 | 0 |
| 0.25 | 0.0009 | 71 | 4 | 17 | 44 | 8 | 2 | 0 |
| 0.25 | 0.0010 | 65 | 4 | 17 | 38 | 8 | 2 | 0 |

**Table 4.** Results of the association-rule algorithm applied to the test data.

| confidence | support | #taxa | max depth | #lvl1 | #lvl2 | #lvl3 | #lvl4 | #lvl5 |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 0.05 | 0.0001 | 1355 | 6 | 228 | 756 | 259 | 81 | 30 |
| 0.05 | 0.0002 | 646 | 5 | 107 | 362 | 134 | 34 | 9 |
| 0.05 | 0.0003 | 435 | 5 | 73 | 247 | 89 | 21 | 5 |
| 0.05 | 0.0004 | 331 | 5 | 62 | 199 | 54 | 12 | 4 |
| 0.05 | 0.0005 | 269 | 5 | 51 | 162 | 42 | 10 | 4 |
| 0.05 | 0.0006 | 211 | 5 | 41 | 127 | 31 | 9 | 3 |
| 0.05 | 0.0007 | 170 | 5 | 35 | 101 | 24 | 7 | 3 |
| 0.05 | 0.0008 | 145 | 5 | 30 | 88 | 18 | 7 | 2 |
| 0.05 | 0.0009 | 135 | 5 | 28 | 83 | 17 | 5 | 2 |
| 0.05 | 0.0010 | 117 | 5 | 26 | 70 | 15 | 4 | 2 |
| 0.10 | 0.0001 | 1258 | 5 | 233 | 794 | 193 | 34 | 4 |
| 0.10 | 0.0002 | 594 | 5 | 111 | 382 | 88 | 12 | 1 |
| 0.10 | 0.0003 | 399 | 5 | 74 | 254 | 63 | 7 | 1 |
| 0.10 | 0.0004 | 305 | 5 | 62 | 201 | 36 | 5 | 1 |
| 0.10 | 0.0005 | 246 | 5 | 51 | 162 | 28 | 4 | 1 |
| 0.10 | 0.0006 | 189 | 5 | 40 | 125 | 19 | 4 | 1 |
| 0.10 | 0.0007 | 154 | 5 | 35 | 100 | 14 | 4 | 1 |
| 0.10 | 0.0008 | 135 | 5 | 32 | 88 | 11 | 3 | 1 |
| 0.10 | 0.0009 | 123 | 5 | 28 | 81 | 11 | 2 | 1 |
| 0.10 | 0.0010 | 106 | 5 | 26 | 68 | 9 | 2 | 1 |
| 0.15 | 0.0001 | 1181 | 4 | 242 | 802 | 121 | 16 | 0 |
| 0.15 | 0.0002 | 548 | 4 | 113 | 379 | 52 | 4 | 0 |
| 0.15 | 0.0003 | 372 | 4 | 76 | 256 | 37 | 3 | 0 |
| 0.15 | 0.0004 | 280 | 4 | 62 | 192 | 24 | 2 | 0 |
| 0.15 | 0.0005 | 225 | 4 | 51 | 151 | 21 | 2 | 0 |
| 0.15 | 0.0006 | 175 | 4 | 39 | 119 | 15 | 2 | 0 |
| 0.15 | 0.0007 | 146 | 4 | 36 | 97 | 11 | 2 | 0 |
| 0.15 | 0.0008 | 126 | 4 | 32 | 84 | 8 | 2 | 0 |
| 0.15 | 0.0009 | 116 | 4 | 29 | 78 | 8 | 1 | 0 |
| 0.15 | 0.0010 | 100 | 4 | 27 | 65 | 7 | 1 | 0 |
| 0.20 | 0.0001 | 1087 | 4 | 240 | 756 | 79 | 12 | 0 |
| 0.20 | 0.0002 | 496 | 4 | 112 | 353 | 28 | 3 | 0 |
| 0.20 | 0.0003 | 334 | 4 | 75 | 238 | 18 | 3 | 0 |
| 0.20 | 0.0004 | 248 | 4 | 61 | 174 | 11 | 2 | 0 |
| 0.20 | 0.0005 | 199 | 4 | 51 | 136 | 10 | 2 | 0 |
| 0.20 | 0.0006 | 151 | 4 | 38 | 103 | 8 | 2 | 0 |
| 0.20 | 0.0007 | 124 | 4 | 33 | 82 | 7 | 2 | 0 |
| 0.20 | 0.0008 | 107 | 4 | 29 | 71 | 5 | 2 | 0 |
| 0.20 | 0.0009 | 99 | 4 | 27 | 66 | 5 | 1 | 0 |
| 0.20 | 0.0010 | 86 | 4 | 25 | 55 | 5 | 1 | 0 |
| 0.25 | 0.0001 | 969 | 4 | 229 | 668 | 61 | 11 | 0 |
| 0.25 | 0.0002 | 424 | 4 | 103 | 296 | 22 | 3 | 0 |
| 0.25 | 0.0003 | 281 | 4 | 70 | 193 | 15 | 3 | 0 |
| 0.25 | 0.0004 | 210 | 4 | 57 | 141 | 10 | 2 | 0 |
| 0.25 | 0.0005 | 167 | 4 | 47 | 109 | 9 | 2 | 0 |
| 0.25 | 0.0006 | 125 | 4 | 34 | 81 | 8 | 2 | 0 |
| 0.25 | 0.0007 | 97 | 4 | 27 | 61 | 7 | 2 | 0 |
| 0.25 | 0.0008 | 81 | 4 | 22 | 52 | 5 | 2 | 0 |
| 0.25 | 0.0009 | 75 | 4 | 21 | 48 | 5 | 1 | 0 |
| 0.25 | 0.0010 | 69 | 4 | 21 | 42 | 5 | 1 | 0 |

**Table 5.** Results of the extended association-rule algorithm applied to the test data.

# Is Gaming the System State-or-Trait? Educational Data Mining Through the Multi-Contextual Application of a Validated Behavioral Model

Ryan S.J.d. Baker

Learning Sciences Research Institute
University of Nottingham
Nottingham, UK
ryan@educationaldatamining.org

**Abstract.** In this paper we discuss the use of a validated behavioral model across multiple contexts. We show that such a model can be used to distinguish between classes of explanations for why that behavior occurs. Specifically, we compare between state and trait explanations for why students game. We use the behavior model to predict each student's gaming frequency in a set of 35 tutor lessons, and then use linear models and the Bayesian Information Criterion to determine which class of explanations predicts gaming behavior more successfully.

## 1 Introduction

In recent years, it has been repeatedly documented that students choose to interact with interactive learning environments in an impressive variety of ways. Some students avoid asking for help at all costs [1], some students game the system [4,8,12], and some students even work thoughtfully and carefully in order to learn the material [1,3,7]. In recent years, a variety of models have been developed which can detect many of these behaviors [1,3,4,7], and some of these models have been incorporated into learning environments which use the models' assessments to respond to differences in student behavior [5,13].

However, there is a question that is fundamental to developing systems that can respond to differences in student behavior: why. Why do students choose to use learning environments differently from each other? And within this, why does a specific student choose to engage in a specific behavior? For example: Why did student 73 choose to game the system?

Broadly, there are two types of potential explanations for why a specific person engages in a specific behavior: **state** explanations, and **trait** explanations. State explanations suggest that some aspect of the student's current state or situation guide a student to engage in that behavior. Trait explanations, by contrast, suggest that specific traits that a student has – such as personality characteristics or preferred meta-cognitive strategies – guide a student to engage in that behavior. Trait explanations can include both fairly fixed traits (such as personality characteristics or learning disabilities) and more fluid traits (such as attitudes or preferred meta-cognitive strategies).

Several studies in recent years have attempted to correlate both state and trait explanations with student behavior in interactive learning environments [cf. 3,7,12], combining student responses on questionnaires with some indicator of their behavior within an interactive learning environment. These studies have found that a wide variety of different factors, both state and trait, are associated with specific student behaviors: however, the correlations have generally been low. For example, across Baker et al [7] and Walonoski and Heffernan [12], seven different explanations (4 state, 3 trait) were found to be statistically significantly associated with gaming the system, but none with an $r^2$ greater than 0.07.

An account which only achieves an $r^2$ of 0.07 can not be considered a primary account for why the behavior occurs. Hence, current approaches do not appear to have made large headway on resolving fundamental questions about why students choose specific behaviors in learning environments.

In this paper, we will argue in favor of a different method for analyzing why students engage in a specific behavior: educational data mining, through the broad, multi-context application of a validated model of student behavior. We will use an existing model of a category of student behavior to predict that behavior's incidence among a substantial number of students. More importantly, we will use that model to predict student behavior across a wide variety of contexts (note that this depends upon a model which has been validated across the full variety of contexts). We will show that this method is effective for distinguishing between the relative impact of state and trait explanations for student behavior, and discuss how this method can be expanded to analyze a large set of explanations quickly and efficiently.

In this paper, we focus specifically on a category of behavior known as gaming the system. Gaming the system is defined as attempting to succeed in an interactive learning environment by exploiting properties of the system rather than by learning the material [4]. Gaming has been found to split into two distinct categories of behavior, one of which is associated with significantly poorer learning [4]. As already mentioned, gaming the system has been found to be statistically significantly associated with a variety of state and trait explanations [cf. 7,12] but those explanations have in all cases achieved low $r^2$.

## 2 Data

In order to analyze whether state explanations or trait explanations are better predictors of whether a student will game the system, we obtained data for 240 students' use of a Cognitive Tutor curriculum [2] for middle school mathematics, during an entire school year (August 2001-May 2002). All of the students were enrolled in mathematics classes in one middle school in the Pittsburgh suburbs which used Cognitive Tutors two days a week as part of their regular mathematics curriculum, year round. None of the classes were composed predominantly of gifted or special needs students. The students were in the 6th, 7th, and 8th grades (approximately 10-13 years old).

Each of these students worked through a subset of 35 different lessons within their Cognitive Tutor curriculum, covering a diverse selection of material from the middle school mathematics curriculum. Middle school mathematics, in the United States, generally consists of a diverse collection of topics, and these students' work was representative of that diversity, including lessons on combinatorics, decimals, diagrams, 3D geometry, fraction division, function generation and solving, graph interpretation, probability, and proportional reasoning. On average, each student completed 13.7 tutor lessons (SD = 4.9), for a total of 3292 student/lesson pairs.

In the analyses presented here, we will analyze whether state explanations or trait explanations are better at predicting whether a student will game the system in a fashion associated with poorer learning [cf. 4]. To determine how often each student gamed the system, in each lesson, we applied a detector of a sub-category of gaming behavior associated with poorer learning [cf. 4,6] to a data set composed of each action by each student, in each of the 35 lessons. The data set was composed of approximately 804,000 actions in the tutor, which equaled 182.9 MB of distilled data in a flat database, or 407 MB of log files prior to distillation. The gaming detector is structurally a Latent Response Model [8]. It assesses gaming by first making predictions about whether each individual action is an instance of gaming, and then aggregates these predictions in order to make coarser grain-size predictions about how often each student games the system in each lesson. The detector was trained using data from five tutor lessons (300 students, using the tutor from 2003-2005) drawn from the same middle school mathematics curriculum as the lessons used in the analysis reported in this paper.

Since the detector was trained using data from four tutor lessons, and is being applied to data from thirty-five lessons, it is reasonable to ask whether the detector will produce reliable estimates of gaming frequency in the lessons it was not trained on. In this case, we can have reasonably high confidence, because the detector has been validated to transfer to new tutor lessons it was not trained on, within this specific tutor curriculum for middle school tutor mathematics. In [6], the gaming detector was trained on three lessons and tested it on a fourth lesson, in four different combinations. All four lessons were drawn from the same middle school tutor curriculum as the thirty-five lessons are drawn from. The detector transferred to lessons it was not trained on with only mild and non-statistically significant degradation in performance. Since all lessons used in the analysis here are drawn from that same curriculum, we have reason to believe that the detector, in general, should be reliable for the lessons studied in this analysis.

Hence, the detector gives us a prediction for gaming frequency for 3292 student/lesson pairs, which we can use to study whether gaming frequency is better predicted through state explanations or trait explanations.

## 3 Analysis and Results

We can determine the relative effectiveness of state and trait explanations, by setting up regression models that attempt to predict each student/lesson gaming frequency using a function on either the student, or the lesson. In other words, we treat both student and lesson as nominal variables, assign each student and/or lesson a value, and attempt to predict the gaming frequency associated with each student/lesson pair. Student is a good proxy for all trait explanations put together, because the sum total of each student's traits should be expressable as one value for that student. Similarly, lesson is a good proxy for all state explanations put together, because the sum total of a number of contextual factors should differ lesson-by-lesson and thus should be expressable as a single value for each lesson. (To explain this another way, imagine a model with 8 trait variables and 8 state variables; each student will have a weighted sum value for those 8 trait variables, and each lesson will have a weighted sum value for those 8 state variables).

Hence, we can attempt to predict gaming behavior with trait explanations by assigning a term to each student, i.e.

$$\text{Gaming Frequency} = \text{Student} + \alpha_0$$

The resulting model has 240 parameters (240 students). The model achieves a moderately low $r^2$ of 0.16, with a Bayesian Information Criterion (BiC) value of 1382. BiC values greater than zero mean that a model is over-fit [10], which suggests that despite the fact that the model's $r^2$ is moderately above zero, the model is in fact somewhat worse than what would be expected, by chance, from a model with 240 parameters.

We can attempt to predict gaming behavior with state explanations by assigning a term to each lesson, i.e.

$$\text{Gaming Frequency} = \text{Lesson} + \alpha_0$$

The resulting model has 35 parameters (35 lessons). The model achieves a considerably better $r^2$ of 0.55, with a Bayesian Information Criterion (BiC) value of -2370. BiC values lower than zero mean that a model predicts the data better than could be expected by chance. The distribution of gaming frequencies, lesson by lesson, is shown in Figure 1.

In addition, the large difference between the BiC values indicate that the trait model is a significantly better of gaming frequency than the state model. A difference of 10 is considered to be evidence equivalent to a p value of 0.01 [10]; these two models' BiC values differ by 3,752.

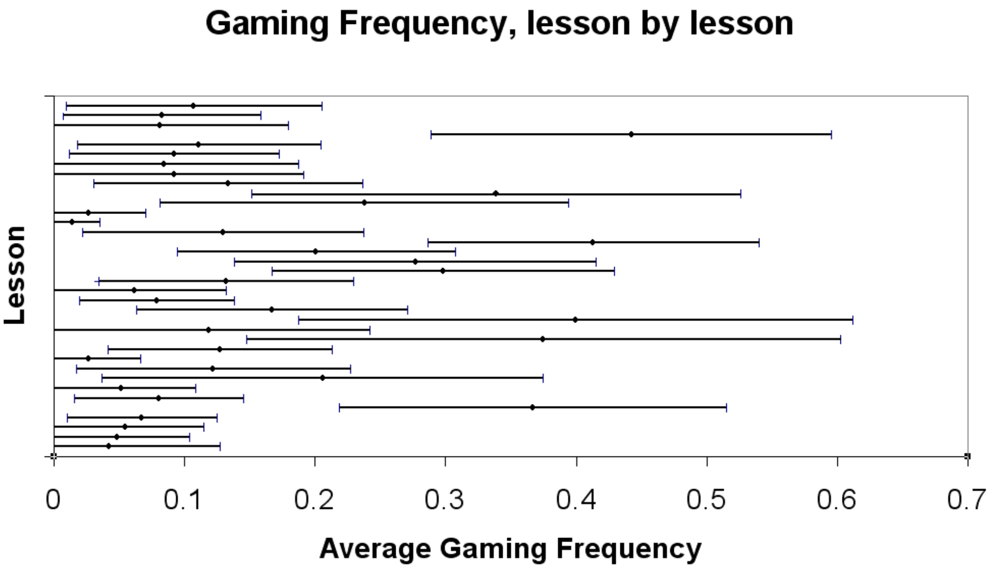


**Fig. 1. Gaming frequency across lessons. Standard deviation bars used instead of standard error bars, in order to show distribution of data rather than statistical significance of difference between groups.**

## 4. Discussion and Conclusions

The models presented here suggest that gaming the system can be generally better understood through state explanations than trait explanations. This suggests that, in order to understand why students game the system, it will be more fruitful for future work to investigate state explanations, rather than trait explanations.

In addition, the relationship found between state explanations and gaming behavior ($r^2$ of 0.55) is much stronger than any of the relationships found through more traditional methods of research ($r^2$ under 0.07). This suggests that the analytical method used here may be more powerful than previous methods used. Analytical methods that dig into the specific contexts, within lessons, which students game with particular frequency may be even more powerful for explaining gaming behavior.

The major difference between the analytical method used here, and prior research, is the number of lesson contexts studied. To our knowledge, previous studies of why students engage in specific behaviors in interactive learning environments either involved only a single lesson/ curricular sub-section [cf. 3,7] or had data from multiple lessons/curricular sub-sections, but used an overall measure of the behavior, which did not make distinctions at the lesson-by-lesson level [cf. 12]. By contrast, the study reported here involved 35 different lessons.

Using data from multiple tutor lessons gives substantial leverage for assessing both state and trait explanations. For assessing state explanations, traditional methods have involved either asking questions that attempt to assess a state’s frequency or existence across the entire use of a system [cf. 3,12], or periodic assessments of a student’s state across a limited amount of time [cf. 11]. Assessing student behavior across a wide variety of states, which is made possible through applying a validated model to many tutor lessons or curricular sub-sections, will inherently have higher power than such traditional approaches.

For assessing trait explanations, dividing data by lessons also gives additional statistical power. Any effective trait explanation should be an effective predictor across multiple contexts. Treating each individual student as a separate predictor of gaming is the strongest possible trait-based explanation of why students game. The fact that this predictor only achieved an $r^2$ of 0.16 is quite strong evidence that trait explanations will not provide the most important explanations for why students game the system.

The analysis discussed here has taken a very high-level view of state explanations and trait explanations. An important area of future work will be to apply these methods to more precise questions, involving individual elements of a student's state. The analysis presented here suggests that states – in general – are an important predictor of why students game. The large, broad, and most importantly, labeled data set that was necessary to conduct the analysis given here will in the future make it possible to conduct very sensitive comparisons of how different aspects of a student's state affects their likelihood of engaging in gaming behaviors.

In the next few years, we believe that the combination of large, broad data sets with models validated across multiple contexts will create a situation where relatively simple techniques for data exploration (such as regression and criterion-based model selection) can answer fundamental questions about why students choose to use learning environments in the ways they do.

**Acknowledgements.** I would like to thank Angela Wagner and Albert Corbett for enabling access to the data used in the analyses here, and Ido Roll, Albert Corbett, and Ken Koedinger for their role in the original development of the detector of gaming behavior used in these analyses. The work presented in this paper was funded by a research fellowship from the Learning Sciences Research Institute, at the University of Nottingham.

## References

1. Aleven, V., McLaren,B.M., Roll, I., and Koedinger, K.R. Toward tutoring help seeking: Applying cognitive modeling to meta-cognitive skills. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems (ITS 2004),* 227-239.
2. Anderson, J.R., Corbett, A.T., Koedinger, K.R., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences,* 4 (2), 167-207.
3. Arroyo, I., Woolf, B. (2005) Inferring learning and attitudes from a Bayesian Network of log file data. *Proceedings of the 12th International Conference on Artificial Intelligence in Education,* 33-40.
4. Baker, R.S., Corbett, A.T., and Koedinger, K.R. (2004) Detecting Student Misuse of Intelligent Tutoring Systems. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems,* 531-540.
5. Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Evenson, E., Roll, I., Wagner, A.Z., Naim, M., Raspat, J., Baker, D.J., Beck, J. (2006) Adapting to When Students Game an Intelligent Tutoring System. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 392-401.
6. Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Roll, I. (2006) Generalizing Detection of Gaming the System Across a Tutoring Curriculum. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems,* 402-411.
7. Baker, R.S., Roll, I., Corbett, A.T., Koedinger, K.R. (2005) Do Performance Goals Lead Students to Game the System? *Proceedings of the 12th International Conference on Artificial Intelligence in Education,* 57-64.
8. Beck, J.E. (2005) Engagement Tracing: using response times to model student disengagement. *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 88-95.
9. Maris, E. Psychometric Latent Response Models. *Psychometrika*
10. Raftery, A.E. (1995) Bayesian Model Selection. Sociological Methodology, 111-196.
11. Rodrigo, M.M.T., Baker, R.S.J.d., Lagud, M.C.V., Lim, S.A.L., Macapanpan, A.F., Pascua, S.A.M.S., Santillano, J.Q., Sevilla, L.R.S, Sugay, J.O., Tep, S., Viehland, N.J.B. (submitted) Affect and Usage Choices in Simulation Problem Solving Environments.
12. Walonoski, J.A., Heffernan, N.T. (2006a) Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems. In Ikeda, Ashley & Chan (Eds.). *Proceedings of the 8th International Conference on Intelligent Tutoring Systems,* 382-391.
13. Walonoski, J.A., Heffernan, N.T. (2006b) Prevention of Off-Task Gaming Behavior in Intelligent Tutoring Systems. In Ikeda, Ashley & Chan (Eds.). *Proceedings of the 8th International Conference on Intelligent Tutoring Systems,* Springer-Verlag: Berlin. pp. 722-724.

# Domain Specific Interactive Data Mining

Roland Hübscher[1], Sadhana Puntambekar[2], and Aiquin H. Nye[3]

[1] Department of Information Design and Corporate Communication, Bentley College, 175 Forest Street, Waltham, MA 02452-4705, U.S.A.
rhubscher@bentley.edu

[2] Department of Educational Psychology, University of Wisconsin, 1025 W Johnson St., Rm 880, Madison, WI 53706, U.S.A.
puntambekar@education.wisc.edu

[3] Raytheon Company, T2SJ01, 50 Apple Hill, Tewksbury, MA 01876, U.S.A.
quin@raytheon.com

**Abstract.** Finding patterns in data collected from interactions with an educational system is only useful if the patterns can be meaningfully interpreted in the context of the student-system interaction. To further increase the chance of finding such meaningful patterns, we extend the mining process with domain and problem specific representations and the pattern detection expertise of qualified users. The user, that is, the researcher looking for patterns, is not just evaluating the result of an automatic data mining process, but is actively involved in the design of new representation and the search for patterns. This approach is used in addition to more traditional methods and has resulted in a deeper understanding of our data.

## 1 Introduction

In the preface to the Educational Data Mining Workshop at ITS 2006, educational data mining is defined as "the process of converting raw data from educational systems to useful information that can be used to inform design decisions and answer research questions" [1]. Information is only useful if it can be meaningfully interpreted in the appropriate context, for instance, in the context of the student-system interaction. Many data and information representations and many mining algorithms exist from which the user,[4] the researcher interested in understanding the data, can choose. It is not uncommon, that the process of developing representations and mining algorithms is separate from mining actual data, done by different groups of researchers taking advantage of their special areas of expertise. However, since knowledge about the problem domain is important to select the appropriate representations and methods, this can also be a disadvantage, especially if the appropriate methods and representations are not readily available. In that case, the user of the mining tool is forced to use whatever is available.

Discovering useful characteristics of data is not a simple method where data is fed into some black box and the interesting characteristics are computed and returned to the user. Mannila, for instance, suggests a process consisting of the following steps: "1. understanding the domain, 2. preparing the data set, 3. discovering patterns (data mining), 4. postprocessing of discovered patterns, and 5. putting the results into use" [2]. Based on our search for informative patterns in our data, we suggest a similar process. However, we emphasize its iterative nature based on a design process and we will describe and illustrate the specific steps with a concrete example from our own mining efforts. Furthermore, although other researchers may implicitly use a similar approach [3–5], it is important that the process is made explicit so that it can be discussed, shared, improved and followed.

Our educational hypermedia system CoMPASS uses dynamic concept maps to support navigation. We are interested in using the logged navigation data to understand how the student-computer interaction can be related to the student's learning strategies and understanding of the subject matter presented by the system. We intend to use the found relationships between student behavior and student learning to provide adaptive prompts to scaffold the learner as well as to provide teachers with realtime feedback about the students performance [6].

[4] We use the following terminology in this paper. The *user* is the person interested in finding patterns. The *learner* or *student* is the person using the educational system.

We only can accept data mining results that can be interpreted meaningfully in the context of the learner using CoMPASS with its specific interface and structure. Sometimes, “interesting” relationships can be found, yet mapping them meaningfully back into the domain where a learner is interacting with a specific system proves very difficult and sometimes even impossible. Thus, we have adopted a method that allows the researcher looking for meaningful patterns in the log data to be part of the mining process. The mining process is an interaction between computer and researcher, both helping each other to find the relationships between log data and student behavior.

This interactive process does not seem to be the norm. Some definitions suggest that the mining process is automatic, for instance, Wikipedia defines data mining as “the process of automatically searching large volumes of data for patterns using tools such as classification, association rule mining, clustering, etc.” [7]. This suggests that the researcher interested in the potential patterns in the raw data is not really involved in the data mining process, but only in the interpretation of whatever the data mining algorithm produces. We propose to use a less narrow view.

The goal of this paper is to discuss the advantages and disadvantages of

- involving the user at various stages in the pattern discovery process,
- the use of domain and problem specific algorithms and representations, and
- the use of an iterative design and discovery process.

In this paper, we describe the interactive mining process we have been using to make sense of the raw data collected from the use of CoMPASS. We illustrate this general method with the domain-specific algorithms and representations used to mine our data for meaningful patterns. We will focus on the method, but will not address in detail the specific algorithms or the insights we have gained in the pedagogical domain. For some of these other results, see [8].

## 2 Interactive Data Mining

Interactive data mining allows the user and the the data mining algorithms to interact with each other. Often, the data is visualized helping the user to understand the patterns better and also allowing the user, and not just the mining tool’s discovery engine, to discover some of the patterns [9, 10]. Efforts to build integrated environments like VDM [11] supporting many data mining and visualization techniques are of great value and we hope, that at some time in the future, we will be able to extend such a system in the way described here. While a tool like VDM gets its power through the many different mining and visualization techniques it provides so that they can be applied almost effortlessly in many domains with different data, we are interested in enabling tools to add specialized algorithms and visualizations relatively easily with some end-user programming tool. This is a long term goal. For now, we simulate this with a set of programs written as needed in the flexible programming language Python.

Our process is based on our work on log data collected from students interacting with CoMPASS. Before we discuss the specifics of CoMPASS and how we analyzed that log data, we present the process in a more general way and then address each step separately.

1. Collect raw data from learner-system interaction
2. Analyze system and its users, use and context
3. Represent raw data in a meaningful way using domain-specific methods
4. Find clusters of similar data points
5. Visualize (members of) clusters
6. Interpret visualizations in the context of the learner-system interaction
7. If results are not good enough (and we have more time), go back to step 2

This process has some similarities with an iterative design process [12] where the understanding of the problem co-evolves with the solution. In other words, as the user is mining the data, the user learns more about the data and will be able to find more appropriate representations and methods. This does indeed require the mining tool to be extensible with some end-user programming language. It also requires the user to be aware that data mining is not a one-shot approach. Initial results need to be used to improve the mining approach to find more interesting results.

We focus on clustering and ignore some other useful methods. Although we do not want to exclude all other methods, clustering allows us to include the user as part of the mining process in a relatively straightforward manner.

Let's start with the first step of the process. Of course, first the raw data has to be collected. It is important that the data is always analyzed with the context in mind in which the data was generated. Thus, it would be a big mistake to collect the data and then hand it off to a data miner who is completely unaware of the learners' characteristics, the educational system and other factors influencing the learner-system interaction and expect that the miner would return anything terribly meaningful. In other words, the patterns are not just in the data. After all, as soon as we talk about patterns there is a bias[5] involved. Since we cannot avoid some bias completely, it should at least be a result of our understanding of the learner-system interaction including the system interface, the learner characteristics and the pedagogical methods used.

Most of the time, we probably do not want to cluster the raw data, but a more meaningful description of it. How the data should be represented depends on the specific circumstances, of what answers the user wants to answer and the data itself. For instance, as we shall show in the next section, we were not so much interested in finding similar behavior, but in finding similar understanding. Thus, we represented the data so that it would capture more the students' understanding than just their behavior.

Clustering the data requires that we develop some kind of distance or similarity measure further biasing the whole discovery process. We again propose to use domain specific metrics that are consistent with the represented data and the questions the user wants to answer with the analysis.

So far, the user has been involved in the process by selecting representations and similarity metrics or possibly developing them anew based on the understanding of the data and the patterns already found in earlier iterations. Once the clusters have been found, the user has to decide how to analyze their members to find common characteristics or patterns. Since we propose to put the burden for finding patterns, at least to some degree, on the user, visualizations may be useful here. And again, domain-specific visualizations should be considered, although standard ones should be used if they are adequate for the current situation. Just creating domain-specific representations for their own sake is a bad idea since developing them is very time consuming.

When the user studies these clusters for interesting patterns, it is important that the interpretation of these patterns must be done within the context in which the raw data was collected. Thus, the circle closes and the user should go back and consider modifying or completely changing some of the representations used based on what was learned during the previous iteration. As the user iterates through the process, the understanding of the data, patterns and their meaning evolves. Finding the answers is not a one-shot approach.

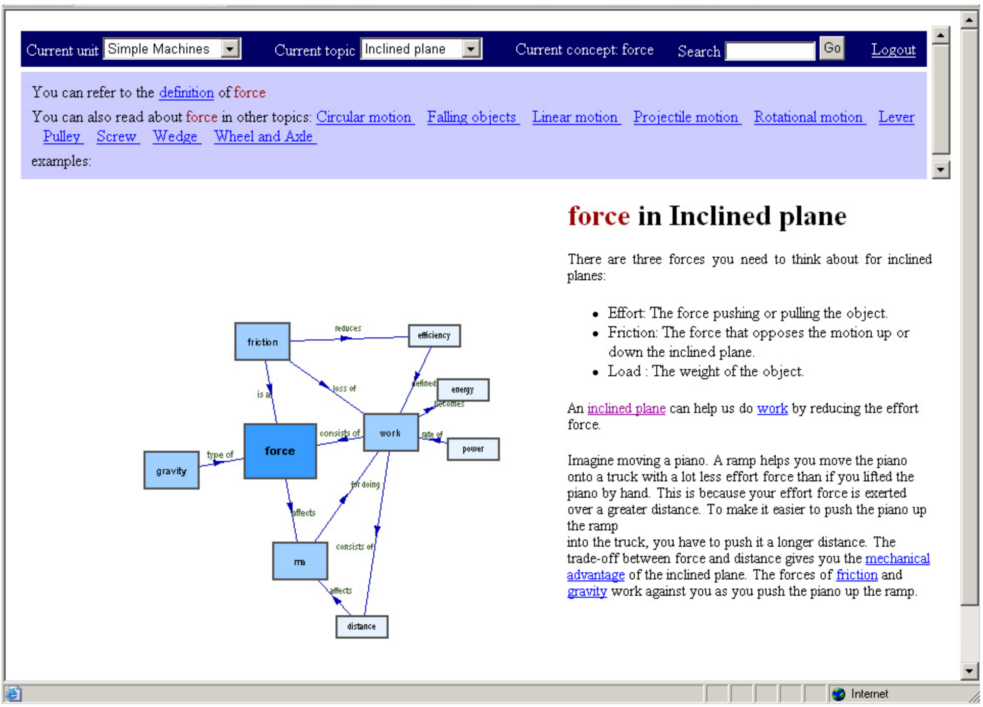
Our proposed method is also somewhat analogical to how expert systems have evolved over the last thirty years. The early expert systems used to ask the user for various inputs, do some reasoning and return the result, or a list of results with some associated confidence factors. Although that mode of operation can be useful under certain circumstances, intelligent systems are viewed now more and more as intelligent assistants helping the user solve the problem collaboratively [13].

## 3 Mining CoMPASS' Navigation Logs

We now illustrate the ideas introduced in the previous section with the data analysis of the navigation data collected with CoMPASS. CoMPASS is an educational hypermedia system with navigation support in the form of dynamic concept maps [8]. CoMPASS helps students understand the relationships between science concepts and principles. It uses two representations, concept maps and text, to support navigation and learning. Each page in CoMPASS represents a conceptual unit such as force or acceleration. A conceptual map of the science concept and other related concepts takes up the left half of the CoMPASS screen, and a textual description takes up the right half of the screen (see Figure 1). The maps are dynamically constructed and displayed with the fisheye technique every time the student selects a concept. The selected (focal) concept is at the center of the map, with the most related concepts at the first level of magnification and those

[5] We use bias in the non-technical sense throughout this paper.

less closely related at the outer level of the map. The maps in CoMPASS mirror the structure of the domain to aid deep learning and are designed to help students make connections, giving students alternative paths to pursue for any particular activity, so that they can see how different phenomena are related to each other.



You can refer to the definition of force

You can also read about force in other topics: Circular motion Falling objects Linear motion Projectile motion Rotational motion Lever Pulley Screw Wedge Wheel and Axle

examples:

**force in Inclined plane**

There are three forces you need to think about for inclined planes:

- Effort: The force pushing or pulling the object.
- Friction: The force that opposes the motion up or down the inclined plane.
- Load : The weight of the object.

An inclined plane can help us do work by reducing the effort force.

Imagine moving a piano. A ramp helps you move the piano onto a truck with a lot less effort force than if you lifted the piano by hand. This is because your effort force is exerted over a greater distance. To make it easier to push the piano up the ramp
into the truck, you have to push it a longer distance. The trade-off between force and distance gives you the mechanical advantage of the inclined plane. The forces of friction and gravity work against you as you push the piano up the ramp.

**Fig. 1.** CoMPASS with navigation support on the left and a description of the concept force in the context of an inclined plane.

We are interested in understanding the navigation paths of the students in CoMPASS for several reasons. The nonlinear nature of hypertext can be used to organize information in multiple ways, reflecting the structure of the described domain. As a result, navigation through hypertext requires the learner to make frequent decisions and evaluations. Providing the proper navigation support is therefore important and understanding how navigation and learning relates to each other is therefore important. Furthermore, we intend to provide adaptive support to the students in form of dynamic prompts triggering metacognitive activities. Such prompts have to be sensitive to the learning context including the students' understanding and potential problems. We hope that we can associate certain navigation patterns with students' understanding to provide the adequate prompts in real time. Similar to [14], we also want to detect in real time students who may have some learning problems so that the teacher, a highly valuable but sparse classroom resource, can focus his or her attention on those students who need it the most.

Before we discuss the steps introduced in the previous section, it is important that we make the questions we are interested in with respect to the data logged in CoMPASS explicit. This allows us to develop the domain and problem specific representations with the research questions and the learner-CoMPASS interaction in mind. One of the goals of CoMPASS is, together with other class room interventions, to scaffold students to gain a deep understanding of the domain specific concepts and their relationships. In other words, we are interested in the students' structural (or relational) knowledge [15]. In the case of the content displayed in Figure 1, the topics are simple machines (e.g., inclined plane, lever, screw) and the concepts are from the domain of mechanics and include energy, force, efficiency and gravity as the concept map shows.

In CoMPASS, navigation data is collected in the form of a sequence of navigation events. Each event consists of the time of the mouse click, the name of the student who clicked on it and the destination page. Since each page contains the description of exactly one concept, every destination page is equivalent to a destination concept. This is not a very rich data source and we were initially worried that we might not find interesting patterns. In addition, the individual interactions are relatively short, that is, the students rarely click on more than twenty links in one session whose duration is normally between 60 and 90 minutes. For each user, the raw data is then collected in an $n \times n$ navigation matrix $N$ such that $N_{ij}$ is the number of transitions from concept $i$ to concept $j$. A transition from $i$ to $j$ simply means that the user, being on the page for concept $i$, has clicked on a link to the page describing concept $j$.

The next step requires to represent the raw data $N$ to increase the chance of finding patterns that address the questions we are interested in. In other word, the new representation needs to have characteristics we consider to be relevant in similar students. Since we are interested in the structural knowledge of a student, we wanted a representation that would emphasize the structure hidden within the navigation data. For this purpose, we applied the Pathfinder Network Scaling procedure computing an approximate representation of the conceptual model of the user [16]. The Pathfinder algorithm was developed to find relevant relations in data that describes the proximity between concepts. Naturally, all concepts are somehow related to all others, however, only the relevant relations should be retained. The Pathfinder algorithm has been successfully used for this task in various domains [17]. We modified the algorithms so that it works for navigation networks where two concepts are closer if there are more direct transitions between them. The resulting Pathfinder network is again an $n \times n$ matrix and can be interpreted as a concept map representing the structural knowledge of an individual learner (see Figure 2 for an example).

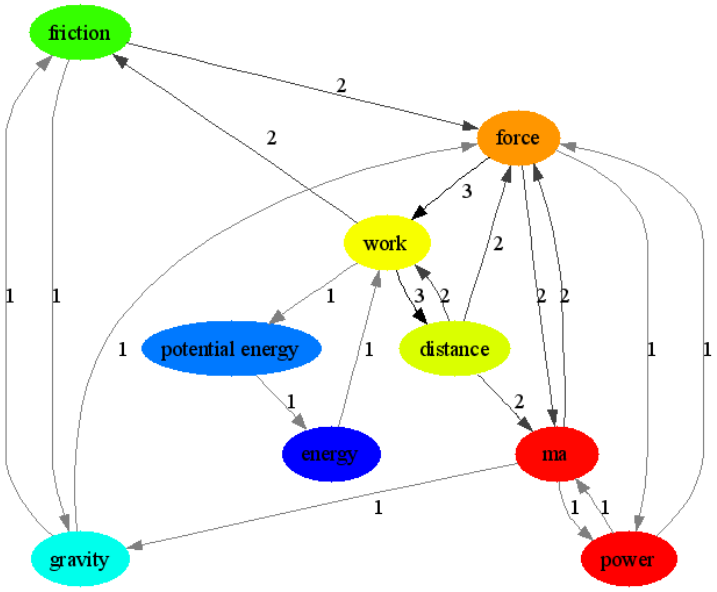


**Fig. 2.** The output of the Pathfinder algorithm which can be interpreted as the concept map describing a student's structural knowledge.

Before the user can look for patterns in the data, the data points need to be clustered [18]. In our case, these data points are the learner models, that is, the Pathfinder networks. We originally applied the k-Means clustering algorithm [19] because of its simplicity and adequate results. Clustering requires some function that measures the similarity (or distance) between two data points. Again, we chose one that was consistent with our interest in the structural characteristics of the learner models. After some testing, we settled on a simple measure suggested by the inventor of the Pathfinder methods [16] which measures the structural similarity of graphs, that is, the Pathfinder networks representing the students' understanding.

Given are two Pathfinder networks $P$ and $Q$ and we want to compute their structural similarity $\mathrm{sim}(P, Q)$. We can assume that they have the same size $n \times n$ and that their node labels are ordered the same in both graphs. If that's not the case, we simply extend both graphs to include all labels and order them lexicographically. However, we do not include any nodes that neither network connects to.

Let $P_{ij}$ and $Q_{ij}$ be the vertex from $i$ to $j$ in $P$ and $Q$, respectively. Since the vertices are ordered, the indices refer to the vertices with the same labels in both networks. Then, the similarity is computed by averaging over the structural similarity of all vertices. The similarity of vertex $i$ in $P$ and vertex $i$ in $Q$ is the the intersection of vertex $i$'s respective outgoing edges divided by the union of the same edges. Since the edges are weighted by the number of transitions, union and intersection are computed as the maximum and minimum, respectively, of the edges' weights. Since the

$$\text{sim}(P, Q) = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum \min_{j=1}^{n}(P_{ij}, Q_{ij})}{\sum \max_{j=1}^{n}(P_{ij}, Q_{ij})}$$

Although, the results we obtained with k-Means were satisfactory, though somewhat unstable—like many other greedy algorithms, k-Means does not always find the optimal solution—we have also used hierarchical clustering which provides more fine-grained information [20]. K-Means clustering creates a partition of the learner models which then are visualized as discussed below. However, when using hierarchical clustering, it is possible to look at many more meaningful subgroups depending on where the cutoff is made as Figure 3 shows. In this figure, the names on the left are the names of the learners. It shows that students *green3* and *red3* are quite similar and so are *purple3* and *yellow3*. So these two clusters can be visualized to see what they have in common, but also the visualizations for the cluster consisting of all four students is generated and so on. As the dendrogram in Figure 3 shows, five meaningful clusters and subclusters are generated and can be visualized.

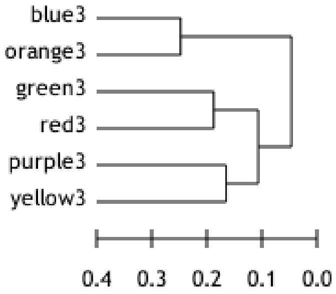


**Fig. 3.** The hierarchical clusterer computes a dendrogram as output.

In the k-Means and the hierarchical clustering algorithm we used the centroid distance function where the distance between two clusters is measured by the distance between the centroids of the two clusters. The centroid of a cluster is the average of all the data points in that clusters, in our case, the average of the Pathfinder matrices.

The next step is, as already mentioned, visualizing the clustered networks. We visualize all clusters in a hierarchical clustering for further study. However, once the similarity becomes small, finding interesting patterns tends to becomes less probable, because the accumulation of several not so similar learner models results in a "washout" effect: in average, each concept is a bit related to each other and nothing characteristic stands out. This, for instance, tends to be true for the trivial cluster including all of the students.

These clusters serve as the starting point for the user to find interesting relationships. Instead of visualizing these clusters in some standard form—we do that, too—we put much effort into finding visualizations that are meaningful with respect to how the students use CoMPASS and how CoMPASS is structured. One obvious representation is the accumulated models, that is, we average the network outputs by Pathfinder for all students in the cluster which results in a network similar to the one show in Figure 2, however, as mentioned the washout effect is a problem.

Before we turned to the type of visualizations described below, we studied the centroids of the clusters like the one in Figure 4. We did indeed find interesting patterns and were able to relate them to the students' learning [8]. Some students were rather focused and explored more or less other topics, others showed a random "pattern" and the ones in Figure 4 a highly linear behavior influenced by the interface. Random and linear behaviors correlated with relatively low learning performance. Although this analysis was quite successful, we are interested in finding additional less obvious patterns with visualizations that hopefully make these patterns easier to recognize.

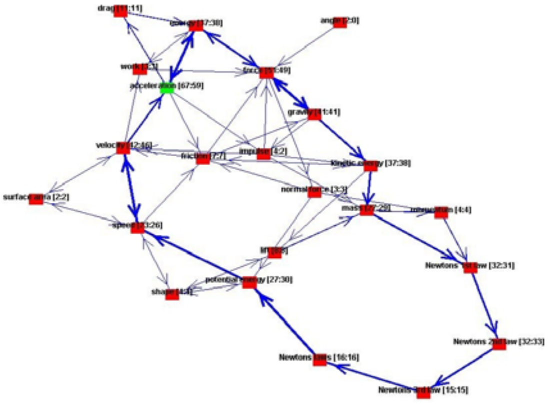


**Fig. 4.** We also analyzed centroids of the clusters.

Examples of visualizations that are much more domain specific are shown in Figures 5 and 6. Instead of providing an aggregate view for a cluster, each cluster member is displayed separately in form of a ring graph (see Figure 5). The ring is based on some important characteristics of CoMPASS and its use as explained below.

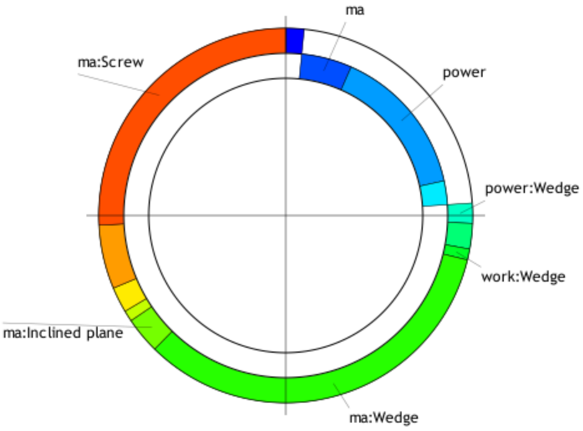


**Fig. 5.** A ring graph describing what descriptions students visited during a session. The outer ring refers to concepts in the context of a topic, the inner ring to context-free definitions.

The domain-specific visualizations are being used by the researchers familiar with CoMPASS and its use. Thus, to understand ring graphs as used here, some details of CoMPASS need to be further explained. CoMPASS provides various types of concept descriptions for middle school students. The types refer to the context in which the concepts are described. For instance, the concept of force can be described in the context of falling objects or in the context of an inclined plane as in Figure 1. In CoMPASS, a concept description without context is called the concept's definition. Since we consider the distinction between descriptions within and without a context pedagogically meaningful in CoMPASS, the two concentric rings in Figure 5 capture this characteristic of CoMPASS. The ring represents a session of using CoMPASS starting at the top and moving clockwise around the ring. The inner ring represents visits by the student to definitions, the outer ring visits to descriptions within some context. Different colors are used to code what concepts are described. We found that it was relatively easy to pick up meaningful patterns by people familiar with CoMPASS and the student-system interaction in which the data is collected.

A new representation we have been working on is the panel graph in Figure 6 where different students are represented with different colors. There are six sections from left to right. The left-most section refers to definitions (context free), the next one to concept descriptions in the context of inclined plane, then in the context of lever, and so on. The navigation events are ordered starting at the top of the graph and going down. Again, this representation captures important relationships of CoMPASS and its use and may support finding interesting behavioral patterns.

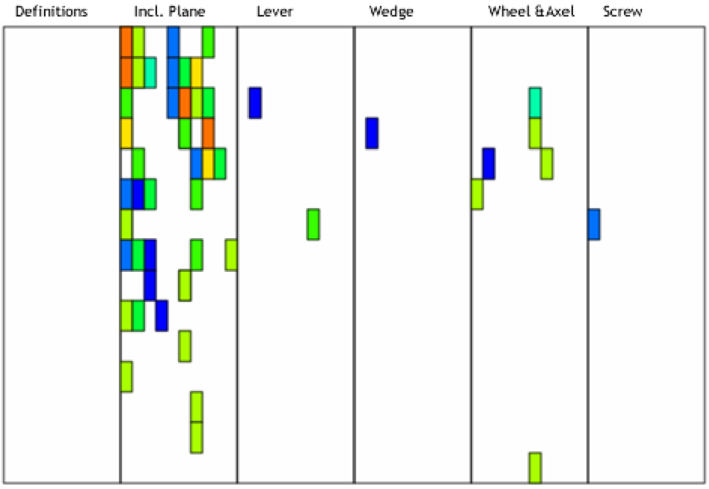


**Fig. 6.** A panel graph comparing the navigation behavior of various student groups. Each group is represented by a specific color.

What is important here is not the ring or panel graph per se, but that it was designed iteratively with the student-system interaction and the research questions in mind. Domain and problem specificity can be quite powerful, though developing these representations is not trivial and takes time. However, having representations that are relatively easy to interpret with respect to the actual research questions makes the representations very useful. Patterns mean immediately something whereas in other situations, patterns are found and then it is sometimes difficult to figure what they actually mean.

## 4 Conclusions

We are interested in finding meaningful patterns in the data collected from the interaction between students and the educational hypermedia system CoMPASS. For instance, we have studied the navigation data also with methods from social network analysis [21] and have found some interesting patterns, however, it has been quite difficult to make sense of these relationships at a pedagogical level. Just pointing out some interesting commonalities that do not have a meaningful interpretation are simply not useful in general.

Therefore, we have proposed an approach that takes advantage of domain and problem specific knowledge and human experts as pattern finders. We do not imply that all data mining should follow the proposed method, but see it more of a way of using and possibly extending existing tools. Our implementation is at the moment still relatively ad hoc where new domain-specific representations and algorithms have to implemented "by hand" in Python. This is quite costly and it is not obvious that an integrated environment could much more easily be extended with new representations.

We have addressed the reasons for using domain specific representations and visualizations and its advantages. However, this approach also has potential disadvantages. As soon as one makes assumptions about what characteristics are interesting and which ones are not, a bias is introduced which may prevent certain patterns from being found. For instance, our focus on the structural knowledge of the students is justified given our research questions, however, it also may keep certain interesting and meaningful relations hidden. After all, one can only see what one displays and as soon as one emphasizes one property, another is being deemphasized [22]. Therefore, the proposed method should be used in addition to more general approaches, not as their replacement.

## Acknowledgments

This material is based upon work supported in part by the NSF under Grants #0437660 (IERI) and #0434624 (NSDL). Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

1. Heiner, C., Baker, R., Yacef, K.: Preface. In: Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems (ITS 2006), Jhongli, Taiwan. (2006)
2. Mannila, H.: Methods and problems in data mining. In Afrati, F., Kolaitis, P., eds.: International Conference on Database Theory, Delphi, Greece, Springer Verlag (1997) 41–55
3. Kay, J., Maisonneuve, N., Yacef, K., Zaïane, O.: Mining patterns of events in students' teamwork data. In: Proceedings of the Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems (ITS 2006), Jhongli, Taiwan. (2006)
4. Merceron, A., Yacef, K.: Educational data mining: a case study. In: Proceedings of the 12th Conference on Artificial Intelligence in Education, Amsterdam, The Netherlands. (2005)
5. Mazza, R., Dimitrova, V.: Visualising student tracking data to support instructors in web-based distance education. In: Proceedings of the Thirteenth International World Wide Web Conference (WWW2004), New York. (2004)
6. Puntambekar, S.: Analayzing navigation data to design adaptive navigation support in hypertext. In Hoppe, U., Verdejo, F., Kay, J., eds.: Artificial Intelligence in Education: Shaping the future of learning through intelligent technologies, IOS Press (2003) 209–216
7. Wikipedia: Data mining. Retrieved January 27, 2007, from: http://en.wikipedia.org/wiki/Data_mining (2007)
8. Puntambekar, S., Stylianou, A., Hübscher, R.: Improving navigation and learning in hypertext environments with navigable concept maps. Human-Computer Interaction **18**(4) (2003) 395–428
9. Aggarwal, C.C.: Towards effective and interpretable data mining by visual. ACM SIGKDD Explorations Newsletter **3** (2002) 11–22
10. Spenke, M., Beilken, C.: Visual, interactive data mining with infozoom – the financial data set. In: "Discovery Challenge" at the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 99, Prague, Czech Republic. (1999)
11. Schulz, H.J., Nocke, T., Schumann, H.: A framework for visual data mining of structures. In: Twenty-Ninth Australasian Computer Science Conference (ACSC2006), Hobart, Tasmania. (2006)
12. Nielsen, J.: Iterative user-interface design. Computer **26**(11) (1993) 32–41
13. Hoschka, P.: Computers As Assistants: A New Generation of Support Systems. Lawrence Erlbaum Associates (1996)
14. Ma, Y., Liu, B., Wong, C.K., Yu, P.S., Lee, S.M.: Targeting the right students using data mining. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco. (2001)

15. Jonassen, D.H., Beissner, K., Yacci, M.: Structural Knowledge: Techniques for Representing, Conveying, and Acquiring Structural Knowledge. Lawrence Erlbaum Associates, Hillsdale, NJ (1993)
16. Schvaneveldt, R.W., ed.: Pathfinder Associative Networks: Studies in Knowledge Organization. Ablex, Norwood (1990)
17. Chen, C.: Visualizing semantic spaces and author co-citation networks in digital libraries. Information Processing & Management **35**(3) (1999) 401–420
18. Merceron, A., Yacef, K.: TADA-Ed for educational data mining. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning (IME$_j$) **7**(1) (2005)
19. Hansen, P., Mladenovic, N.: J-Means: A new local search heuristic for minimum sum-of-squares clustering. Pattern Recognition **34**(2) (2001) 405–413
20. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys **31**(3) (1999) 264–323
21. Wasserman, S., Faust, K.: Social network analysis. Cambridge University Press, New York, NY (1994)
22. Narayanan, N.H., Hübscher, R.: Visual language theory: Towards a human-computer interaction perspective. In Meyer, B., Marriott, K., eds.: Visual Language Theory. Springer Verlag (1998) 85–127

# The Effect of Model Granularity on Student Performance Prediction Using Bayesian Networks

Zachary A. Pardos, Neil T. Heffernan, Brigham Anderson, Cristina L Heffernan

Worcester Polytechnic Institute, Carnegie Mellon University, Worcester Public Schools
{zpardos, nth}@wpi.edu

**Abstract.** A standing question in the field of Intelligent Tutoring Systems and User Modeling in general is what is the appropriate level of model granularity (how many skills to model) and how is that granularity derived? In this paper we will explore varying levels of skill generality within 8th grade mathematics using models containing 1, 5, 39 and 106 skills. We will measure the accuracy of these models by predicting student performance within our own tutoring system called ASSISTment as well as their performance on the Massachusetts standardized state test. Predicting students' state test scores will serve as a particularly stringent real-world test of the utility of fine-grained modeling. We employ the use of Bayes nets to model user knowledge and for prediction of student responses. The ASSISTment online tutoring system was used by over 600 students during the school year 2004-2005 with each student using the system 1-2 times per month throughout the year. Each student answered over 100 state test based items and was tutored by the system with help questions called scaffolding when they made a mistake. Each student answered on average 160 scaffold questions. Our results show that the finer the granularity of the skill model, the better we can predict student performance for our online data. However, for the standardized test data we received, it was the 39 skill model that performed the best. We view the results as support for using fine-grained models even though the finest-grained sized model did not also predict the state test results the best.

## 1 Introduction

There are many researches in the user modeling community working with Intelligent Tutoring Systems (ITS) (i.e, Mayo & Mitrovic [12], Corbett, Anderson et al, [6], Conati & VanLehn [5], Woolf [2]) and many who have adopted Bayesian network methods for modeling knowledge [15, 4, 11]. Even methods that were not originally thought of as Bayesian Network methods turned out to be so; Reye [14] showed that the classic Corbett & Anderson's "Knowledge tracing" approach was a special case of a dynamic belief network.

We seek to address the question of what is the right level of granularly to track student knowledge. Essentially this means how many skills should we attempt to track? We will call a mapping of skills to questions a skill model. We will compare different skill models that differ in the number of skills and see how well the different models can fit a data set of student responses collected via the ASSISTment system [7]. We are not the first to do model-selection based on how well the model fits real student data (i.e., [9, 11]). Nor are we the only ones that have been concerned with the question of granularity; Greer and colleagues [10, 15] have investigated method of using different levels of granularity, and different ways to conceptualize student knowledge. We are not aware of any other work where researchers attempted to specifically answer the question of "what is the right level of granularity to best fit a data set of student responses".

### 1.1 The Massachusetts Comprehensive Assessment System (MCAS)

The MCAS is a Massachusetts state administered standardized test that covers English, math, science and social studies for grades 3rd through 10th. We are focused on 8th grade mathematics only. Our work relates to the MCAS in two ways. First we have built our content based upon ~300 publicly released items from previous MCAS math tests. Secondly, we will be evaluating our models by using the 8th grade 2005 MCAS math test which was taken after the online data being used was collected.

### 1.2 Background on the ASSISTment Project

The ASSISTment system is an e-learning and e-assessing system [7]. In the 2004-2005 school year more than 600 students used the system about once every two weeks as part of their regular classroom curriculum. Eight math teachers from two schools would bring their students to their computer lab, at which time students would be presented with randomly selected question items. Each tutoring item, which we call an ASSISTment, is based upon a publicly released MCAS item which we have added "tutoring" to. If students get the item correct they are advanced to the next question. If they answer incorrectly, they are provided with a small "tutoring" session where they are asked to answer a few questions that break the problem down into steps. The first scaffolding question appears only if the student gets the item wrong. We believe that the ASSISTment system has a better chance of showing the utility of fine-grained skill modeling due to the fact that we can ask scaffolding questions that break the problem down into parts and allow us to tell if the student got the item wrong because they did not know one skill versus another. Most MCAS questions that were presented as multiple-choice were converted into text-input questions to reduce the chance of guess. As a matter of logging, the student is only marked as getting the item correct if they answer the question correctly on the first attempt.

## 2 Models

We define a skill model as a set of skill names and a mapping of those skill names to questions and scaffolding in the ASSISTment tutoring system. The single skill in the coarse grain model called the WPI-1 represents all of 8th grade mathematics, while the finest grain model, the WPI-106, breaks the same subject matter into 106 different skills. Bayesian Belief Networks (BBN) provide the framework to represent these skill models in a relatively straight forward fashion. They also provide powerful inference and inspectability which is essential for skill reporting to teachers, students or parents.

### 2.1 Creation of Fine-Grained Skill Model

In April of 2005, we staged a 7 hour long "coding session", where our subject-matter expert, Cristina Heffernan, with the assistance of the 2nd author, set out to make up skills and tag all of the existing 8th grade MCAS items with these skills. This coding session took place at Worcester Polytechnic Institute (WPI) after most of the tutor interaction had taken place. No student data was used to inform this coding session. There were about 300 released test items to code. Because we wanted to be able to track learning between items, we wanted to come up with a number of skills that were somewhat fine-grained but not too fine-grained such that each item had a different skill. We therefore imposed upon our subject-matter expert that no one item would be tagged with more than 3 skills. She gave the skills names, but the real essence of a skill is what items it was tagged to. To create the coarse-grained models we used the fine-grained model to guide us. For the WPI-5 model we started off knowing that we would have the 5 categories; 1) Algebra, 2) Geometry, 3) Data Analysis & probability, 4) Number Science and 5) Measurement. Both the National Council of Teachers of Mathematics and the Massachusetts Department of Education use these broad classifications as well as a 39 skill classification. After our 600 students had taken the 2005 state test, the state released the items from the test and we had our subject matter expert tag up those test items. Shown bellow, in Figure 1 is a graphical representation of two of the skill models we used to predict the 2005 state test items. The 1 and 5 skills are at the top of each graph and the 29 questions of the test are at the bottom. The intermediary nodes are logic gates which are described in the next subsection.

**Fig 1.a** – WPI-1 MCAS Model

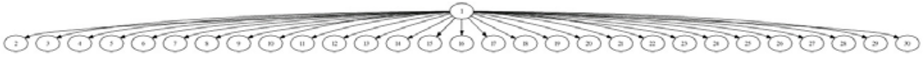
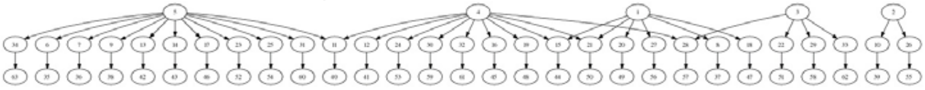


**Fig 1.b** – WPI-5 MCAS Model



It is the case that with the WPI-39 and WPI-106 models, many of the skills do not show up on the final test since each year only a subset of all the skills needed for 8th grade math are represented.

The WPI-1, WPI-5 and WPI-39 models are derived from the WPI-106 model by nesting a group of fine-grained skills into a single category. This mapping is an aggregate or "is a part of" type of hierarchy as opposed to a prerequisite hierarchy [4]. Figure 2 shows the hierarchal nature of the relationship between WPI-106, WPI39, WPI-5 and WPI-1.

**Figure 2**. – Skill Transfer Table

| WPI-106 | WPI-39 | WPI-5 | WPI-1 |
|---|---|---|---|
| Inequality-solving →<br>Equation-Solving →<br>Equation-concept → | setting-up-and-solving-equations | Patterns-Relations-Algebra | The skill of "math" |
| Plot Graph → | modeling-covariation | | |
| X-Y-Graph →<br>Slope → | understanding-line-slope-concept | | |
| Congruence →<br>Similar Triangles → | understanding-and-applying-congruence-and-similarity | Geometry | |
| Perimeter →<br>Circumference →<br>Area → | using-measurement-formulas-and-techniques | Measurement | |

## 2.2 How the Skill Mapping is Used to Create A Bayes Net

In a typical ASSISTment, an original question will be tagged with a few skills, but if the student answers the original question incorrectly they are given scaffolding questions that are tagged with only a single skill. This gives the system a good chance inspecting which skills a student does not know in the case that they get the original question wrong. Figure 3 shows an example part of the Bayes Net. Each circle is a random Boolean variable. The circles on the top row are variables representing the probability that a student knows a given skill, while the circles on the bottom row are the actual question nodes. The original question in this example is tagged with three skills, *scaffold question 1* is tagged with congruence and *scaffold question 2* is tagged with Perimeter. The ALL[1] gates assert that the student must know all skills relating to a question in order to answer correctly. The ALL gates also greatly simplify the network by reducing the number of parameters specified for the question nodes to just two (guess and slip). The prior probabilities of the skills are shown at the top and the conditional probabilities of getting the questions correct are shown at the bottom of the figure. Note that these parameter values were set intuitively (if a student knows all the skills for an item there will be a 0.95 chance they will get the question correct, but only a 0.10 chance otherwise). This specifies a 10% guess and 5% slip (calculated by 1 - P(Q) | Gate). A prior probability of 0.50 on the skills asserts that the skill is just as likely to be known as not know previous to using the ASSISTment system. When we later try to predict MCAS questions, a guess value of 0.25 will be used to reflect the fact that the MCAS items being predicted are all multiple choice, while the online ASSISTment items have mostly been converted from multiple-choice to "text-input fields". This model is simple and assumes all skills are as equally likely to be known prior to being given any evidence of student responses, but once we present the network with evidence it can quickly infer probabilities about what the student knows.

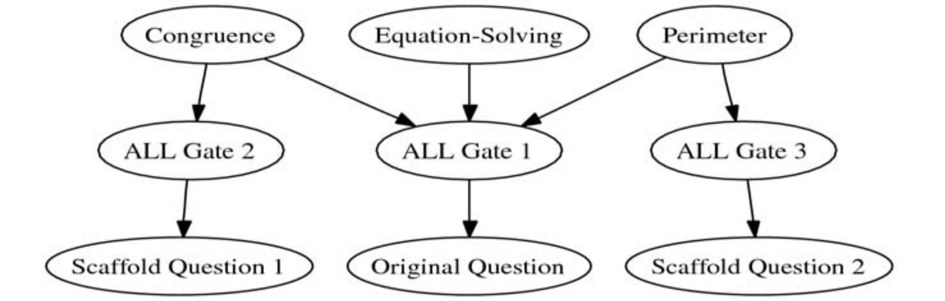
[1] The term 'ALL' gate is used instead of 'AND' gate because our software implementation of Bayesian networks uses AND gates only for nodes with two parents.

**Figure 3**. – Sample of Bayes Directed Graph with default priors and parameters

| P(Congruence) |
|---|
| 0.50 |

| P(Equation-Solving) |
|---|
| 0.50 |

| P(Perimeter) |
|---|
| 0.50 |



| Gate | P(Q) |
|---|---|
| True | 0.95 |
| False | 0.10 |

| Gate | P(Q) |
|---|---|
| True | 0.95 |
| False | 0.10 |

| Gate | P(Q) |
|---|---|
| True | 0.95 |
| False | 0.10 |

## 3 Bayesian Network Application

We created a Bayesian framework using MATLAB and Kevin Murphy's Bayes Net Toolkit (BNT) [(http://bnt.sourceforge.net/)] with Chung Shan's BIF2BNT utility. This framework assesses the skill levels of students in the ASSISTment system and measures the predictive performance of the various models. First the skill model, which has been converted into Bayesian Interchange Format from our database, is loaded into MATLAB. A student-id and Bayesian model are given as arguments to our prediction program. The Bayesian model at this stage consists of skill nodes of a particular skill model which are appropriately mapped to the over 1,400 question nodes in our system (300 original questions + 1,100 scaffolds). This can be referred to as the online model. We then load the user's responses to ASSISTment questions from the database and enter their responses into the Bayesian network as evidence. Using join-tree exact inference, a significant improvement over the sampling likelihood-weighting algorithm previously employed [13], posterior marginal probabilities are calculated for each skill in the model for that student.

We now discuss how student performance prediction is done. After the probabilistic skill levels of a particular student have been assessed using the specified skill model, we load a Bayes model of the MCAS test which is also tagged according to the skill model used for the online model. The MCAS test model looks similar to the training model, with skill nodes at top mapped to ALL nodes, mapped to question nodes. In this case we take the already calculated marginal probabilities of the skill nodes from the online model and import them as soft, probabilistic evidence in to the test model. Join-tree inference is then used to get the marginal probabilities on the questions. The probabilities for all 29 questions are summed to produce the final predicted score.

# 4 RESULTS

An early version of the results in this section (using approximate inference instead of exact inference and without Section 4.2) appears in a workshop paper [13]. Before we present the results we will provide an example, in Table 1, of how we made some of the calculations. To predict each of the 29 questions (rows) we used the skills associated with the question to ask the Bayes Net what the probability is that the user will get the question correct. Question three has two skills, and it consistently viewed as harder by each of the students' (columns). We get a *predicted score* by taking the sum of the probabilities for each question and then taking the ceiling of that to convert it into a whole number. Finally, we find the percent error by taking the absolute value of the difference between predicted and actual score and dividing that by 29. The *Average Error* of 17.28% is the average error across the 600 students for the WPI-5. We repeat this procedure for the WPI-1, WPI-5, WPI-39 and WPI-106 models in Table 2.

| **Test Question** | **Skill Tagging (WPI-5)** | **user 1 P(q)** | **user 2 P(q)** | **…** | **user 600 P(q)** | **Average Error** |
|---|---|---|---|---|---|---|
| 1 | Patterns | 0.2 | 0.9 | … | 0.4 | |
| 2 | Patterns | 0.2 | 0.9 | … | 0.4 | |
| 3 | Patterns & Measurement | 0.1 | 0.5 | … | 0.2 | |
| 4 | Measurement | 0.8 | 0.8 | … | 0.3 | |
| 5 | Patterns | 0.2 | 0.9 | … | 0.4 | |
| :: | :: | :: | :: | | :: | |
| 29 | Geometry | 0.7 | 0.7 | … | 0.2 | |
| | | | | | | |
| | **Predicted Score** | 14.2 | 27.8 | … | 5.45 | |
| | **Actual Score** | 18 | 23 | … | 9 | |
| | **Error** | 10.34% | 17.24% | … | 12.24% | **17.28%** |

**Table 1.** Tabular illustration of prediction calculation and error for the MCAS model.

### 4.1 MCAS Prediction Results

The prediction results in Table 2 are ranked by error rate in ascending order. The error rate represents how far off, on average, the prediction of student test scores were for each model. The MAD score is the mean absolute deviance or the average raw point difference between predicted and actual score. The under/over prediction is our predicted average score minus the actual average score on the test. The actual average score will be the same for all models. The centering is a result of offsetting every user's predicted score by the average under/over prediction amount for that model and recalculating MAD and error percentage. WPI-5, for example, under predicts student scores by 3.6 points on average. For the centered calculations we add 3.6 points to every predicted score of users in that model and recalculate MAD and error. The choice was made to calculate centered scores for a few reasons: 1) student might take the MCAS test situation more seriously than weekly usage of the ASSISTment system, 2) we would expect to be under-predicting since we are using data from as far back as September to predict a test in May and our model, at present, does not track learning over time. Although the centering method also obscures the differences between models, it is used as a possible score to expect after properly modeling the factors mentioned above.

| Model | Error | MAD Score | Under/Over Prediction | Error (After Centering) | Centered MAD Score |
|---|---|---|---|---|---|
| **WPI-39** | 12.86% | 3.73 | ↓ 1.4 | 12.29% | 3.57 |
| **WPI-106** | 14.45% | 4.19 | ↓ 1.2 | 14.12% | 4.10 |
| **WPI-5** | 17.28% | 5.01 | ↓ 3.6 | 13.91% | 4.03 |
| **WPI-1** | 22.31% | 6.47 | ↓ 4.3 | 18.51% | 5.37 |

**Table 2.** Model prediction performance results for the MCAS test. All models' non-centered error rates are statistically significantly different at the p<.05 level.

### 4.2 Internal/Online Data Prediction Results

To answer the research question of how well these skill sets model student performance *within the system* we measure the internal fit. The internal fit is how accurately we can predict student answers to our online question items, original questions and scaffolds. If we are able to accurately predict a student's response to a given question, this brings us closer to a computer adaptive tutoring application of being able to intelligently select the appropriate next questions for learning and or assessing purposes. Results are shown bellow.

| Model | Error | MAD Score | Under/Over Prediction | Error (After Centering) | Centered MAD Score |
|---|---|---|---|---|---|
| **WPI-106** | 5.50% | 15.25 | ↓ 12.31 | 4.74% | 12.70 |
| **WPI-39** | 9.56% | 26.70 | ↓ 20.14 | 8.01% | 22.10 |
| **WPI-5** | 17.04% | 45.15 | ↓ 31.60 | 12.94% | 34.64 |
| **WPI-1** | 26.86% | 69.92 | ↓ 42.17 | 19.57% | 51.50 |

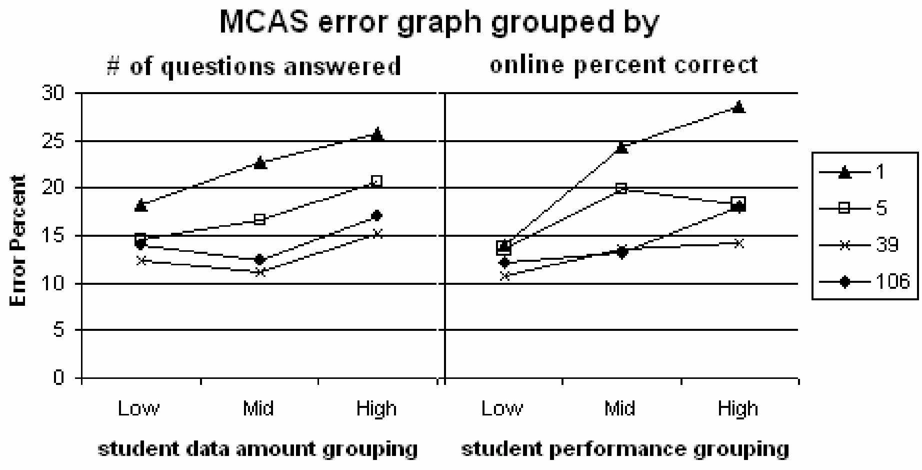
**Table 3.** Model prediction performance results for internal fit

Like with the MCAS prediction, the internal fit was run on a single student at a time. The calculation of error is the same as for the MCAS test except that the probability of getting the question correct is rounded to 0 or 1. For each question answered by the student, that data point was held out and the rest of the student data was offered to the Bayes net as evidence. The inference was then made on that question giving the probability the student will get the question correct. If the probability of correct was greater than 0.5, 1 point was added to the predicted total point score, otherwise no points were added. The absolute difference between the predicted total point score and actual point score was then divided by the total number of questions answered by the student and that is the error percentage score. This method was employed to maintain symmetry with the methodology from the MCAS test predications in the above section. All the differences between the models in Table 3 were statistically significantly different at the p < .05 level.

## 5 Discussion and Conclusions

The results we present seems to be mixed on first blush. The internal fit of the different models had clear results showing that the finer grained the model, the better the fit to the data collected from the ASSISTment system. This result is in accord with some other work we have done using mixed-effect-modeling rather than Bayes nets [8]. Somewhat surprising, at least to us, is that this same trend did not continue as we expected in the result shown in Table 2. In hindsight, we think we have an explanation. When we try to predict the MCAS test, we are predicting only 29 questions, but they represent a subset of the 109 skills that we are tracking. So the WPI-106, which tries to track all 106 skills, is left at a disadvantage since only ¼ of the skills it is tracking are relevant on the MCAS test. Essentially 75% of the data that the WPI-106 collects is practically thrown out and never used. Whereas the WPI-39, which does the best, can benefit from its fine-grained tracking and almost all of its skills are sampled on the 29 item MCAS test.

**Figure 4.** – Analysis Graph



In Figure 4 we decided to try to dig into our results so we could better understand how our models perform. Quite surprising to us, we found that the top performing third of students were predicted much worse than the bottom third with all models. Another surprise was that all models predict worse with high amounts of online data versus low amounts. We do not have a firm explanation for this.

As a field we want to be able to build good fitting models that track many skills. Interestingly, item response theory, the dominate methodology used in assessing student performance on most state tests tends to model knowledge as a unidimensional construct, but allowing the items themselves to vary in difficulty (and other properties of items like discrimination and the probability of guessing). Some of our colleagues are pursuing item response models for this very dataset [1, 3] with considerable success, but we think that item response models don't help teachers identify what skills a students should work on, so even though it might be very good predictor of students, it seems to suffer in other ways. We should remind ourselves if you have two models that can predict the data equally well, the finer-grained model is probably the more intepratable and more usefull to use to give reports to teachers.

### 5.1 Future Work

Our results suggest the 106 skill model as being best for internal fit while 39 skill model is best for the MCAS test, however, a combination of models may be optimal. Building a hierarchy in an aggregate or prerequisite way [4] will likely best represent the various granularities of student understanding and comprehension. These levels of understanding may change over time, so a dynamic Bayes approach will be needed to model these changes as well as model the important variable of learning. This will greatly improve our internal accuracy and will likely show the most benefit to the finer-grained models since the learning of a particular skill will be identifiable. Difficulty is another variable that has the potential to improve model performance. There are many ways to modeling difficulty; the challenge will be to find a method that compliments our current skill models. Additional research into handling the scaffolding selection effect and data filtering will also be explored in future research.

### Acknowledgements

This research was made possible by the US Dept of Education, Institute of Education Science, "Effective Mathematics Education Research" program grant #R305K03140, the Office of Naval Research grant #N00014-03-1-0221, NSF CAREER award to Neil Heffernan, and the Spencer Foundation. All of the opinions in this article are those of the authors, and not those of any of the funders. This work would not have been possible without the assistance of the 2004-2005 WPI/CMU ASSISTment Team that helped make possible this dataset, including Ken Koedinger, Brian Junker, Carolyn Rose & Andrea Knight all at CMU, and Mingyu Feng, Abraao Lourenco, Michael Macasek, Goss Nuzzo-Jones, Kai Rasmussen, Leena Razzaq, Steven Ritter at Carnegie Learning, Terrence Turner, Ruta Upalekar, and Jason Walonoski.

# REFERENCES

[1] Anozie N., & Junker B. W. (2006). Predicting end-of-year accountability assessment scores from monthly student records in an online tutoring system**.** In Beck, J., Aimeur, E., & Barnes, T. (Eds). Educational Data Mining: Papers from the AAAI Workshop. Menlo Park, CA: AAAI Press. pp. 1-6. Technical Report WS-06-05.

[2] Arroyo, I., Woolf, B. (2005) Inferring learning and attitudes from a Bayesian Network of log file data. Proceedings of the 12thInternational Conference on Artificial Intelligence in Education. 33-40.

[3] Ayers E., & Junker B. W. (2006). Do skills combine additively to predict task difficulty in eighth grade mathematics? In Beck, J., Aimeur, E., & Barnes, T. (Eds). Educational Data Mining: Papers from the AAAI Workshop. Menlo Park, CA: AAAI Press. pp. 14-20. Technical Report WS-06-05.

[4] Carmona1, C., Millán, E., Pérez-de-la-Cruz, J.L., Trella1, M. & Conejo, R. (2005) Introducing Prerequisite Relations in a Multi-layered Bayesian Student Model.
In Ardissono, Brna & Mitroivc (Eds) *User Modeling 2005; 10th Internaton Confrence.* Springer. 347-356

[5] Conati, C., Gertner, A., & VanLehn, K. (2002). Using bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, *12*(4), 371–417.

[6] Corbett, A. T., Anderson, J. R. & O'Brien, A. T. (1995) Student modeling in the ACT programming tutor. Chapter 2 in Nichols, P. D., Chipman, S. F. and Brennan, R. L. (eds.) (1995). Cognitively diagnostic assessment. Lawrence Erlbaum Associates: Hillsdale, NJ.

[7] Feng, M., Heffernan, N.T., & Koedinger, K.R. (2006b). Predicting state test scores better with intelligent tutoring systems: developing metrics to measure assistance required. In Ikeda, Ashley & Chan (Eds.). Proceedings of the 8th International Conference on Intelligent Tutoring Systems. Springer-Verlag: Berlin. pp. 31-40. 2006.

[8] Feng, M., Heffernan, N. T., Mani, M., & Heffernan, C. (2006). Using Mixed-Effects Modeling to Compare Different Grain-Sized Skill Models. In Beck, J., Aimeur, E., & Barnes, T. (Eds). Educational Data Mining: Papers from the AAAI Workshop. Menlo Park, CA: AAAI Press. pp. 57-66. Technical Report WS-06-05. ISBN 978-1-57735-287-7.

[9] Mathan, S. & Koedinger, K. R. (2003). Recasting the Feedback Debate: Benefits of Tutoring Error Detection and Correction Skills. In Hoppe, Verdejo & Kay (Eds.), *Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies., Proceedings of AI-ED 2003* (pp. 39-46). Amsterdam, IOS Press.

[10] McCalla, G. I. and Greer, J. E. (1994). Granularity-- based reasoning and belief revision in student models. In Greer, J. E. and McCalla, G. I., editors, Student Modelling: The Key to Individualized Knowledge--Based Instruction, pages 39--62. Springer--Verlag, Berlin.

[11] Mislevy, R.J., & Gitomer, D. H. (1996). The role of probability-based inference in an intelligent tutoring system. User-Modeling and User Adapted Interaction, 5, 253-282.

[12] Mayo, M., Mitrovic, A. Using a probabilistic student model to control problem difficulty. Proc. ITS'2000, G. Gauthier, C. Frasson and K. VanLehn (eds), Springer, pp. 524-533, 2000.

[13] Pardos, Z. A., Heffernan, N. T., & Anderson, B., Heffernan, C. L.. Using Fine-Grained Skill Models to Fit Student Performance with Bayesian Networks. Workshop in Educational Data Mining held at the Eight International Conference on Intelligent Tutoring Systems. Taiwan. 2006.

[14] Reye, J. (2004). Student modelling based on belief networks. *International Journal of Artificial Intelligence in Education: Vol. 14*, 63-96.

[15] Zapata-Rivera, J-D and Greer, J.E. (2004). Interacting with Inspectable Bayesian Models. *International Journal of Artificial Intelligence in Education. Vol. 14*, 127-163.

# Searching for student intermediate mental steps

Vivien Robinet, Gilles Bisson, Mirta Gordon, Benoît Lemaire

Laboratory TIMC-IMAG (CNRS/UJF UMR 5525)
University of Grenoble, Faculté de Médecine
38706 La Tronche Cedex, FRANCE
{first name.last name@imag.fr}

**Abstract.** This paper presents a general method for identifying student intermediate mental steps from sequences of actions stored by problem solving-based learning environments, in order to provide feedback to teachers on knowledge that statistically seems to be used by a particular student. When many intermediate mental steps are possible, ambiguity is removed using what is already known about the student. The system uses a student model to search within a huge space of possible actions, and updates this student model consequently. The user model distinguishes between two different cognitive processes: (1) planning the action by focusing on a particular part of the environment and considering an action type and (2) performing the action.

## 1 Introduction

We are concerned with learning environments in which students are required to perform successive actions. In this paper, we are more specifically interested in the way we may automatically discover student *mental intermediate steps* from a set of observable actions recorded from the environment. This problem compares to the famous *assignment of credit* problem [**1**], in which the goal is to determine knowledge elements directly involved in the observable student behavior. In our case, these knowledge elements are only unitary mental operations. These are called knowledge events by VanLehn [2]. Although our approach is intended to be hooked up to various learning environments, we are currently focusing on algebra learning using the APLUSIX learning environment [3]. Given algebraic equations or inequations to be solved, students using APLUSIX proceed step by step as they would do on a notebook with the only imposed constraint that the expressions entered at any resolution step must be syntactically well formed. In this context, our goal is to discover *mental intermediate steps* of a student modifying an equation. For instance, if a student realizes a wrong transformation from "2x+9=8+6x" to "8x=17", we could assume that he probably performed these mental intermediate steps (Hyp 1), which could be correct or incorrect[1]:

$$\mathbf{2x+9=8+6x} \rightarrow_{\text{correct movement}} 2x-6x+9=8 \rightarrow_{\text{incorrect calculation}} 8x+9=8 \qquad \text{(Hyp 1)}$$
$$\rightarrow_{\text{correct movement}} 8x=8-9 \rightarrow_{\text{incorrect calculation}} \mathbf{8x=17}$$

However, the previous student action could actually be explained in another way, (Hyp 2) involving correct algebraic calculations and incorrect movements:

$$\mathbf{2x+9=8+6x} \rightarrow_{\text{incorrect movement}} 2x+6x+9=8 \rightarrow_{\text{correct calculation}} 8x+9=8 \qquad \text{(Hyp 2)}$$
$$\rightarrow_{\text{incorrect movement}} 8x=8+9 \rightarrow_{\text{correct calculation}} \mathbf{8x=17}$$

Without any additional information, it is not possible to select which path the student has most probably mentally followed. The usual way is to rely on statistical information from huge sets of

[1] Even if calculation precedes algebra in teaching, our students often make wrong calculations when asked for solving algebraic problems

student problem-solving data. Teachers have compiled this information from experience, but other approaches are possible. For instance, Tsiriga & Virvou [4] rely on machine learning techniques to initialize the student model. First, students are assigned a stereotype depending on their ability to perform a preliminary test. Student's degree of knowledge is then estimated using a distance weighted k-nearest neighbor algorithm by positioning student among others whose knowledge is already known.

The specificity of our approach is that we see the problem as a recursive problem: discovering this path is dependent on the student model which is in turn updated from these intermediate steps. In other words our approach is to take into account the information which is already known about the current student to adjust what we know from the general statistical information. Let us illustrate, this point: suppose we know the student had performed the following steps (in bold) just before:

a) **3+2x+9=5+4x-2x** ➔correct movement. **2x+9=5-3+4x-2x**

b) **2x+9=5-3+4x-2x** ➔incorrect calculation 2x+9=8+4x-2x ➔incorrect calculation **2x+9=8+6x**

From this data, our partial student model will be something like: "The student tends to perform correct movements and incorrect algebraic calculations". The first path (Hyp 1), which involves correct movements and incorrect algebraic calculations, will thus be considered more probable for this particular student, even if it is not the case for the majority of students.

## 2 Our user model

The foundation of our model is to consider that in many learning problems, when students are faced with a new state of the environment on which they have to perform an action, they would engage in two kinds of cognitive processes:

1) *planning the action* which reflects the *intention* of the student, consist of focusing on a particular part of the environment in view of a planned type of action. Let us take some Air Traffic Control (ATC) examples, for illustrative purpose only. For instance, an air traffic controller would select a plane with the idea of asking him to wait a bit more before landing, similarly a student faced with "2x+9=8+6x" and asked to solve for x, would select "+6x" with the idea of moving it on the other side of the equation, etc.
2) *performing this action*. For instance, in ATC, the controller would ask the plane to wait a bit more by entering in a well-defined communication procedure. In algebra the student would change "+6x" into "-6x" while moving it to the other side of the equation, etc. Here is an example:

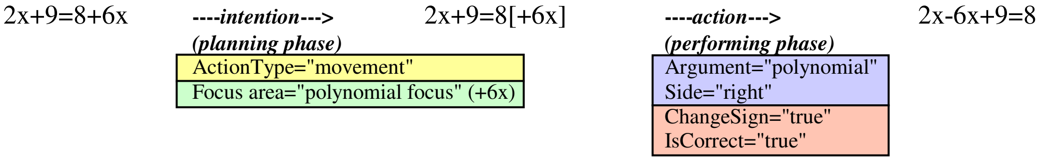


**Fig. 1.** Illustration of the two cognitive processes, (intention, action), leading from one state to the next one.

It is crucial to distinguish among these two steps since a student can be good at identifying useful actions, but fails to perform them, whereas another one may select inappropriate actions but perform them correctly.

We will now present how this model can be implemented in a probabilistic framework. This kind of approach has been already used in the literature, for instance by means of bayesian networks [5].

### 2.1 Modeling the planning phase

In the student model, this phase is represented as a twofold object containing the *focus area* where an action could be performed and the *type of this action*. In our algebra domain, we identified 61 such pairs: <explicit factorization, polynomial focus>, <explicit factorization, negative number>, <reduction, positive number>, <direct calculation, positive number>, <movement, polynomial focus>, etc.

At a step t, to each pair is attached a probability which depends on the prior probability at time t-1, the number of action types that may be applied and the focus area chosen by the student. If several pairs are candidates, the one which is actually applied by the student (or which we guess has been mentally applied) will have its probability increased while probabilities of other possible focus will be decreased (Fig.2).

| | Probability | Possible focus | Actual focus | | Probability | Possible focus | Actual focus | | Probability |
|---|---|---|---|---|---|---|---|---|---|
| <calculation, polynomial focus> | 0.2 | 0 | 0 | | 0.2 | 1 | 0 | | 0.1 |
| <movement, positive number> | 0.2 | 1 | 0 | | 0.15 | 0 | 0 | | 0.15 |
| <movement, polynomial focus> | 0.2 | 1 | 1 | → | 0.3 | 1 | 1 | → | 0.4 |
| <implicit factorization, positive focus> | 0.2 | 1 | 0 | | 0.15 | 0 | 0 | | 0.15 |
| <fraction addition, polynomial focus> | 0.2 | 0 | 0 | | 0.2 | 0 | 0 | | 0.2 |

**Fig. 2.** This example corresponds to the intention presented in Fig.1. Three intentional pairs were possible, but the polynomial movement was used by the student. The latter got its probability increased whereas the other two were decreased according to the actual number of possible focus. In the next step, two pairs were candidates and the polynomial movement action was applied again. Probabilities were updated accordingly.

This part of the user model, which is continuously updated, therefore contains probability values for each kind of action the user is likely to consider. Thus, at a given moment, probability values reflect the student's beliefs.

### 2.2 Modeling the performing phase

This part of the user model describes at a high-level of generalization the user behavior when he does an *action*. We can compare this approach to the one presented by Freyberger, Heffernan and Ruiz [6] in which they construct a transfer model to provide information about what skills are required by the student to solve a particular problem. Similarly, our process will be able to find relevant cross-interactions between attributes and will generalize attributes' values that correspond to similar student behaviors.

An *action* is a generalized vector of *context* and *transformation* attributes that are domain-dependent; their goal is to describe the environment and the student operations. For instance, in the ATC domain, *context* attributes could be "number of planes", "local weather", "fuel level" for each plane, etc. whereas *transformation* attributes could be "ask plane to wait" "ask plane for landing" "ask plane for changing altitude", etc.

In our algebra domain restricted to actionType="movement", we are using 27 *context* attributes such as "sign of focus area", "side of focus area" or "polynomial focus area" and 13 *transformation* attributes such as "change sign of focus area" or "correctness of the transformation".
Each time our system predicts a mental action, a new *context-transformation* vector is generated. Moreover, in order to identify some general student behaviors, these transformation vectors are aggregated using a hierarchical clustering method based on a Manhattan distance between actions. During the aggregation process, context and transformation attributes are generalized inside each cluster to produce generalized vectors of *actions*, as presented below:

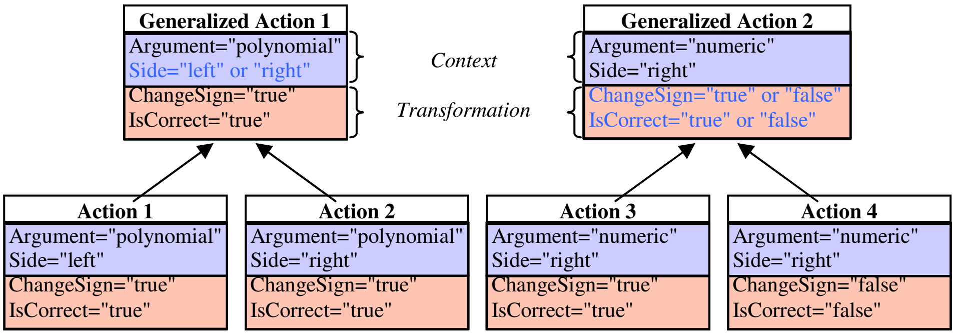


**Fig. 3.** Two examples of generalized-actions. Left: aggregation of two similar actions leading to the generalization of the "left" and "right" values of the side *context* attribute. Right: aggregation of both ChangeSign and IsCorrect *transformation* attributes, to give Generalized Action 2.

Clustering stops at a predefined threshold depending on a generalization level which was experimentally set. The result is a set of generalized *actions* that the student is likely to perform. To each generalized *action* is assigned a probability value that depends on the number of aggregated *actions* in the cluster (i.e. relative frequency). This information is used in the process of detecting mental intermediate steps as we will now describe.

## 3 Predicting student intermediate mental steps

Given two student states produced within the learning environment, the goal is to identify intermediate mental steps in-between, that is a sequence of alternating steps of *intention* (I) and *action* (A). The chain between two consecutive explicit states (initial and final state) may involve N mental steps as follows:

**initial state** ➔ $I_1$ ➔ $A_1$ ➔ mental state ➔ .... ➔ mental state$_{N-1}$ ➔ $I_N$ ➔ $A_N$ ➔ **final state**

Since very many pairs (I, A) could have been performed mentally by the student at each stage, we are faced with a huge search space in which we are looking for the most probable path according to what we know about this student.

The user model gives a probability value to each *intention* (I) and *action* (A) candidate. This value will be used to select the next node in the search space. Searching in this space is done by a best-first search algorithm. This kind of algorithm expands the most promising node, according to a heuristic function. In our case, this function takes into account first the probability of the operations as defined in the student model and second, the distance to the goal, which is the distance between the current state and the final state. In our algebra domain, defining such a distance is tricky because algebraic expressions can be very close while having very different surface forms. For instance, "2-4x=11" appears quite different from "11=-4x+2" at the surface level, although it is the same. Expressions are therefore transformed into trees before computing this distance, and the algorithm recursively tries to match nodes in order to minimize the distance between sub-nodes. Fig.4 presents the searching process.

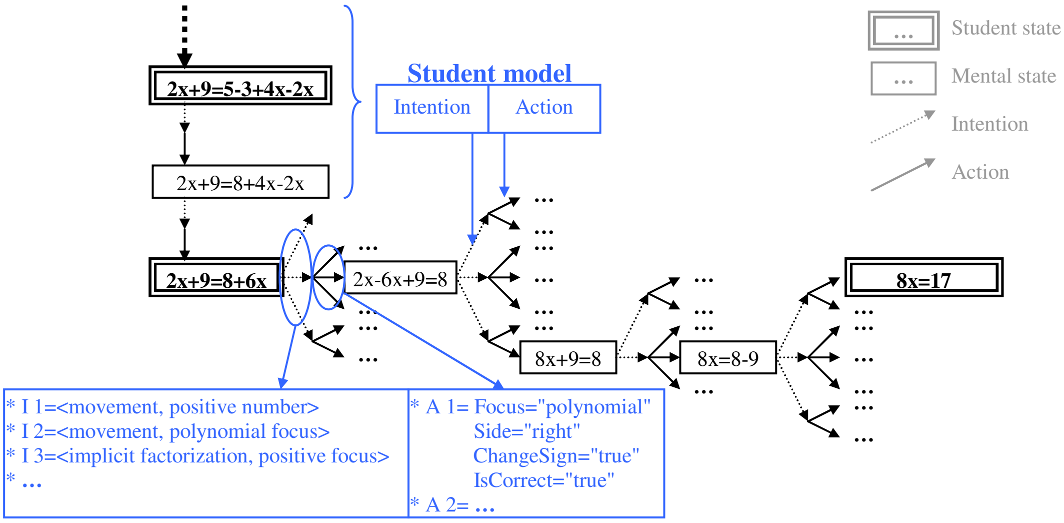


**Fig. 4.** Example of searching process from student initial equation "2x+9=8+6x" to "8x=17" using intention (I) and action (A) given by the partial student model.

## 4 Conclusion

This method has been applied to data produced by 40 French secondary school students. Each student performed about 50 movement steps, from which we discovered about 100 mental steps. Computing takes about two minutes per student, leading to about five generalized actions

We have created a model that is able to adapt to various levels of granularity in the student's production. To reach this goal, it is necessary to make hypotheses about intermediate steps students could have performed mentally. But several interpretations (paths) are possible for a same pairs of initial / final states. Our idea is to supplement the classical approach which tends to choose the most probable actions among a large set of students, by introducing what is already known about the particular student.

To do that, we dynamically use probabilities given by our partial student model at each step of our research tree. It is therefore possible to have an idea of how a student will prepare his/her action, i.e. on which terms he/she will focus on, and which type of action he/she will choose. It is also possible to characterize the way the student will probably perform the chosen action, i.e. what transformation s/he will accomplish given a particular focus.

Given a sequence of equations, we are able to find intermediate steps that are probable for a particular student. We believe this method is quite general because the representation formalism is based on attributes, which are appropriate for most domains.

Most of this work has been implemented: the student intention model is operational and guides the search of intermediate mental steps between student equations. Probabilities evolve over time while the model is built. The only thing which remains to be done is to update our equiprobabilized initial model with a priori statistical knowledge about students. The action phase works independently but it is not yet connected to the detection of mental steps. Consequently, these probabilities do not evolve over time.

## References

1. Beck, J.E., Woolf, B.P. : High-level Student Modeling with Machine Learning. Proceedings of the Intelligent Tutoring Systems Conference (2000) 584-593.
2. VanLehn, K.: The Behavior of Tutoring Systems. International Journal of Artificial Intelligence in Education 16 (2006) 227-265.
3. Nicaud, J.-F., Bouhineau, D., Huguet, T.: The Aplusix-Editor: A New Kind of Software for the Learning of Algebra. Proceedings of the 6th International Conference on Intelligent Tutoring Systems. Lecture Notes In Computer Science, Springer Vol. 2363. Biarritz, France and San Sebastian, Spain (2002) 178-187.
4. Tsiriga, V., Virvou, M.: Initializing the Student Model using Stereotypes and Machine Learning. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (2002).
5. Anthony J.: Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues (1995) 193-251.
6. Freyberger, J., Heffernan, N., Ruiz, C.: Using Association Rules to Guide a Search for Best Fitting Transfer Models of Student Learning. Proceedings of 7th Annual Intelligent Tutoring Systems Conference, Maceio, Brazil. (2004).

# Improving the Prospects for Educational Data Mining

Steven L. Tanimoto

University of Washington
Dept. of Computer Science and Engineering
Box 352350, Seattle, WA 98195, USA
tanimoto@cs.washington.edu

**Abstract.** Data mining is an important paradigm for educational assessment. The usual assumption is that mining is performed after educational activity with that activity having been designed without regard for the mining process. This paper discusses how the prospects for successful mining can be improved by imposing constraints or biases on the activities and instruments that generate the data. These biases involve one or more of the following: (a) encouraging, requiring or training students to communicate effectively and often during the course of learning activities, (b) building more instrumentation into the learning environment to enable capturing more kinds of data, including evidence of student attention, (c), enriching the logged expressions themselves so that more inferences from them can be made more easily and with general purpose tools, and (d) seeding the log files with reliable assessment data to help anchor subsequent inferences. A variation on the mining paradigm integrates mining methods into the learning environment itself, so that various forms of "articulated assessment" can become practical. Articulated assessment is the coordination of unobtrusive but less reliable assessment techniques with traditional direct-questioning methods in such a way as to follow a policy that balances the needs for accuracy and unobtrusiveness.

**Keywords:** educational assessment, data mining, articulated assessment, unobtrusive assessment, online learning environments, intelligent tutoring systems, student modeling.

## 1 Introduction

Data mining is an important data analysis methodology that has been successfully employed in many domains, and which has become especially popular after the World Wide Web made large volumes of data on many topics widely available. It has been used to analyze byproducts of intelligent tutoring system sessions and other educational activities for purposed of evaluating the activity, the systems, or building models of students or their interactions with systems. Data mining has also been considered as a methodology for extracting shorter-term educational assessment data in order to fill out components of student models such as average time on task, attention span, etc.

The arguments put forth in this paper are intended to help reach the goals of greater accuracy in the results of mining, wider latitude in the scope of questions that can be effectively answered by mining, and greater transparency in the inference processes.

## 2 Supporting Unobtrusive Assessment

The University of Washington's project on intensive, unobtrusive assessment seeks to harness the full power of computers in making useful assessments of student learning "behind the scenes." The keystone in this project is a system called INFACT that facilitates the creation and capture of evidence of student learning while students engage in problem-solving and construction activities. Before giving a brief description of INFACT, here is the motivation for unobtrusive assessment.

### 2.1 Goals of Unobtrusive Assessment

Foremost in our motivation is the desire to improve student learning in the context of problem solving and artifact construction. The assessment involves diagnosing student misconceptions and problematical habits. The results are used by teachers and systems to make opportune suggestions to students, select assignments and make pedagogical decisions. The need for unobtrusiveness is a reflection of the cost of interventions with tests, in terms of student motivation to learn and satisfaction with the activity. Another reason to develop unobtrusive assessment is to take advantage of interaction data and records of student communication that are already captured by the computer-based learning environment. Yet one more hope for unobtrusive assessment is that it can be continuous, so that needless gaps in the system's knowledge of the student's cognitive state can be avoided. While unobtrusive techniques might never fully replace traditional testing methods, they may provide new options to teachers and learners that can be used to adapt the pedagogical environment to their needs.

### 2.2 The INFACT Online Learning Environment

When we created INFACT, we set out to "computerize" the facet-based teaching approach successfully developed for physics (Minstrell, 1992). In this method, students are challenged to predict or explain phenomena that go against their intuitions. In their discussions, they reveal their preconceptions. Their ideas are diagnosed by the instructor, using a catalog of previously observed misconceptions as a guide. Then the teacher presents them with special examples that confront their misconceptions.

The original purpose of INFACT was to host these discussions (and thus obtain a record of them) and to facilitate the diagnosis by also hosting the catalog of misconceptions and providing a database facility for recording the diagnoses (Tanimoto et al, 2000). INFACT stands for Integrated, Networked, Facet-based Assessment Capture Tool. Unlike the DIAGNOSER tool, which makes facet diagnoses according to the results of multiple-choice testing (Levidow et al, 1991), INFACT was designed to support the inference of facets directly from the records of student discussions.

To explain what INFACT is, let's consider the services it provides. At the heart of INFACT is the "Forum" which is a group-oriented written discussion area that uses a threaded-newsgroup format. INFACT has special features for controlling the visibility of student messages (Tanimoto et al, 2002). Closely associated with the forum is a graphical communication tool called INFACT-Sketch that supports "conversational sketching." Around this core of communication tools are computation and construction tools for students, such as a programming facility and an image processing system. Teachers have access to administration and assessment tools that include a markup (annotation) facility for making free assessments and facet-based assessments, an editor for facet catalogs, editors and application monitors for rule-based and Bayes-net based automatic assessment, and an editor and administration facility for traditional multiple-choice testing. Facilities are also included for file sharing by students and teachers, and visualization of assessment data by teachers. Additional details are given in (Tanimoto et al, 2005).

### 2.3 The Relationship between Data Mining and Unobtrusive Assessment

Post-logging data mining by itself is unobtrusive on one level, because the session is over, the student has gone home, and is not bothered by the system when the inferences are made. Data gathering and logging, on the other hand, may be obtrusive or unobtrusive (depending on how the data is generated and collected), but data mining is philosophically attuned to unobtrusive assessment because of the decoupling of inference from logging. Nonetheless, we can consider some degree of coupling. We do this not to make data mining an intrusive process, but in order to help it make better inferences. The remainder of this paper discusses several approaches.

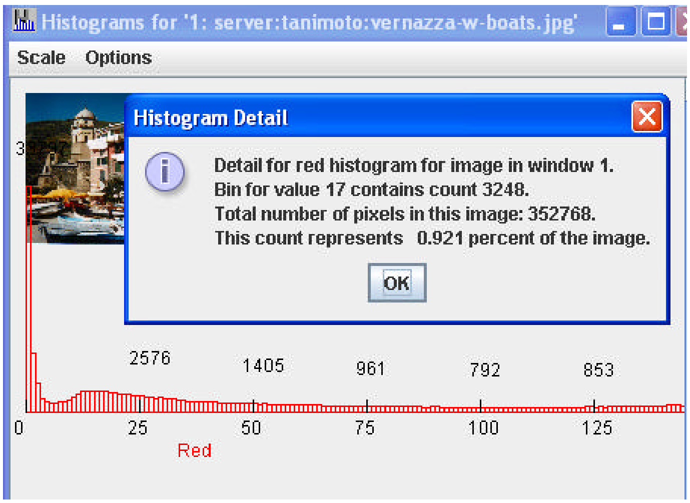
## 3 Helping Students Communicate More Readily and Clearly

One important avenue for "enriching the ore" for mining is to increase the amount and the quality of student communication during sessions, and to incorporate this into the log data stream (David, 2005; Mostow, 2004). With INFACT instructors can easily require such communication in assignments. In addition, students can be taught to communicate more openly through specific training exercises. Graphical communication with INFACT-Sketch is a case in point. We use two particular play activities to get students used to conversational sketching. One activity is "Graphical Telephone," in which one student draws an object and passes the sketch to another, who makes a minor modification, after which it is passed to another, etc. It is particularly amusing when the starting subject is the face of a groupmate. The other activity, "Collaborative Comics," is a group storytelling one, in which the first team member creates the first frame of a comic strip, and passes the sketch on to the next team member, etc. These activities help build habits of group interaction via graphics.

At this time, INFACT does not capture audio or video of students during sessions. However, a written equivalent of "think-aloud" activity ("thinking in the fingers") is a viable methodology for capturing more evidence about student cognition. Such behavior can be encouraged through credit-awarding schemes (participation points, etc). A related idea is to engage students at two levels during their problem solving discussions. While they are direct participants in the discussion, they can also be tasked with evaluating the contributions of their classmates through rating mechanisms. Such ratings can serve as extra hints to data mining methods that particular messages or excerpts are worthy of extra attention, or that they may serve to ground inferences from messages that relate to them.

## 4 "Capturing" Student Attention

Here we really mean capturing *evidence* of student attention. Unless eye-tracking systems are incorporated into the learning environment, it is difficult to know whether a student is actively reading something on the screen or simply daydreaming or tuned out. One approach to better capturing this information is to redesign the interface so that some amount of additional interaction is encouraged and/or required for the reading. This means rethinking observation processes, transforming them from relatively passive activities to explicitly active processes. To turn the experience of reading a page from simply an eye-moving activity to a combined eye-moving and mouse-moving/clicking activity requires two changes: a change of the widget that renders the page, and a change of the information structure to make it hierarchical. A two-level hypertext structure may be sufficient to achieve the goal. This "active observation" strategy thus encourages active observation and it makes passive observation more difficult. The extra level of activity required of students should not be so much as to be a burden. If it easily leads to repetitive stress injury or a much slower rate of reading, then it has gone too far. It is particularly



Histogram Detail

Detail for red histogram for image in window 1.
Bin for value 17 contains count 3248.
Total number of pixels in this image: 352768.
This count represents 0.921 percent of the image.

OK

Figure 1. Illustration of an observation tool that encourages but does not require active observation. When the student clicks on a histogram bin, detailed information about the bin is provided in the dialog box. A log file record corresponding to this instance of active observation is created.

valuable when the material is so dense and rich that students would normally spend a lot of time on it with little explicit interaction, leaving assessment and mining processes in the dark about what they were thinking. An example of an observation instrument for students that takes the unobtrusive approach is a histogram display for images shown in Fig. 1. When the student asks for the display, three full histograms for an image (one histogram for each of the red, green and blue components of the color image) are shown. However, by clicking on individual bins of the histogram, the student can get an exact count of the number of pixels with that value as well as a percentage value for the fraction of the image represented by that bin. When the student clicks, an observation event is registered in INFACT, and the log ends up containing a representation of this observation made by the student.

The image processing system PixelMath, hosted by INFACT, allows students to inspect image pixels in a somewhat unusual way. They can zoom and unzoom as with many image programs, but when they zoom in far enough, they see the numeric pixel values superimposed on the colored pixel squares. In any activity that requires students to work closely with the numeric values of particular pixels, the zooming and unzooming event records represent the student's focus of attention fairly well. Sometimes, students get lost in the details of an image. Navigation can be difficult because of the size and complexity of the image. Dead ends in the navigation can lead to log files polluted with events that do not necessarily reflect an investment of hope by students in their relevance to a task. However, these events do represent the navigation trouble. A smart analysis system needs to be able to distinguish between such navigation problems and intended observations. This is a possible challenge for the incorporation of active observation mechanisms in learning environments.

## 5 Making Log Files More Expressive

While the enrichment methods described in the preceding two sections involve changes to the student experience, another method is not dependent on making such changes. Instead, it involves altering the representation of events in the log file. Foremost in this approach is overcoming a historical tendency to make log files cryptic in order to save file space. The changing economics of disk space should make us adopt the most robust representation techniques, not the absolutely cheapest ones. Four approaches are these: (a) representing each event completely, (b) using English words, (c) using English grammar, and (d) using standard log-file forms. The first of these means making each log-event record almost self-contained. Distribution of meaning in log-file records among code words and corresponding dictionaries elsewhere adds to the challenges of interpreting the records and inferring patterns from them. Another way of describing this approach is having the system that administers the educational experience "connect the dots" within the log to reduce the likelihood of post-activity inference errors.

The use of English words in log file entries can facilitate (a) the use of general data-mining tools, and (b) a human's configuration of such tools. Many tools are designed to process natural-language text. In order to apply them, their target data must be in the form of text, not encoded binary data. Data mining tools are often exploratory pattern analysis tools that benefit from human-expert configuration or guidance. The use of English terms is likely to help these experts keep track of the meanings of record components and apply common-sense reasoning to the task of configuring the data-mining algorithms and evaluating their results. The use of natural language grammar is an extension of the idea of using English terms. The one caveat here is that English grammar admits a wide variety of forms, and it is a good idea to use a small number of simple, standard forms to avoid the need for parsing or ambiguity resolution during data mining. One approach toward standardization is the creation of a language or metalanguage for expressing the format of log files (Iksal and Choquet, 2005). It may be too soon to try to standardize log files, because they require agreement at the level of ontologies, not just formats. However, if log files from multiple environments are to be integrated by data mining systems, it would help if they adhered to standards.

## 6 Seeding Log Files with "Ground Truth"

Another way to make log files easier to interpret is to alter the educational environment and experience somewhat, so that a limited amount of hard-core educational assessment data is captured, analyzed and the results entered into the log file up-front. This may be easy, in intelligent tutoring systems where such assessment may already be performed (Mostow, 2004); but it may require a change in the student experience for constructive tools, such as a dynamic geometry program, a computer software development environment, or a circuit simulation system. Such information would typically have to be obtained with obtrusive interventions involving multiple-choice testing or other highly-directed student tasks. The benefit of such information is that such anchored assessment elements could serve as the seeds that at data-mining time could grow into islands (or a continuum) of highly reliable inferences. An analogy to video representation with the MPEG standard may be instructive: the MPEG data stream includes special video frames that are complete and accurate representations of the video signal at key points in time. The other frames in the video sequence are expressed in terms of these key frames using difference expressions, which tend to be much more compact than full frames. However, the intermediate frames are not completely accurate. The key frames are required to anchor the evolving scene to prevent errors from accumulating too much. With this approach, systems like INFACT need to use a technique we call "articulated assessment." Articulated assessment is a combination of traditional (obtrusive) educational assessment and unobtrusive assessment under the administration of an agent that dynamically optimizes the balance of the two to obtain the requisite accuracy and pedagogical characteristics.

## 7 Human vs. Automatic Data Mining, and Transparency for Students

Does it matter whether the data mining will be performed by humans or machines? Baker et al found that, as a substitute for video and other high-fidelity recordings, textual descriptions can be devised that serve the main purposes almost as well (Baker et al, 2005) when analysis will be done by human coders. The suggestions we have given for enriching log files should apply no matter whether humans or automated agents are performing the analysis. We can imagine that exploratory data mining will best be done by humans interacting with statistical tools. The log files need to be intelligible to both.

There is a possible side benefit of improving the richness of the log file for data mining. That is allowing the capture and assessment processes within the learning environment to be more transparent. By opening up a view of the log stream to the students, they may get a better understanding of how they are being assessed. Such transparency is consistent with the philosophy of supporting open learner models (Bull, 2004) and is a subject of current research. Intelligibility of the log files to students is then a key factor in the success of the transparency in engendering understanding and trust in the system.

## 8 Conclusion

There is a variety of ways that activity logs produced by computer-based learning environments can be enriched to make their subsequent analysis more accurate and fruitful. While some of these involve changing the student experience, others have only to do with the way that logged events are formulated. They all involve thinking of the computer-based educational learning environment and the data mining system as parts of a larger, integrated process.

## References

1. Baker, R., Corbett, A. T., and Wagner, A. Z. Human classification of low-fidelity replays of student actions. *Proceedings of the Workshop on Educational Data Mining* (held at the 8th International Conference on Intelligent Tutoring Systems -- ITS 2006). Jhongli, Taiwan, pp.29-36.
2. Bull, S. Supporting learning with open learner models. *Proc. 4th Hellenic Conference with International Participation: Information and Communication Technologies in Education*, Athens, 2004.

3. David, J.-P. State of art of tracking and analyzing usage. Kaleidescope Project report D32.3.1 Final. Online at http://telearn.noe-kaleidoscope.org/warehouse/JeanPierre-David-2005.pdf
4. Iksal, S., and Choquet, C. Usage analysis driven by models in a pedagogical context. *Proc. AIED Workshop on Usage Analysis for Learning Systems*, 2005. http://lium -dpuls.iut -laval.univ-lemans.fr/aied-ws/PDFFiles/iksal.pdf.
5. Levidow, B. B., Hunt, E., and McKee, C. 1991. The DIAGNOSER: A HyperCard tool for Building theoretically based tutorials. *Behavior Research Methods, Instruments, and Computers*, Vol. 23, 249-252.
6. Minstrell, J. Facets of students' knowledge and relevant instruction. In Duit, R., Goldberg, F., and Niedderer, H. (eds.), *Res. in Physics Learning: Theo. Iss. & Empir. Stud*. Kiel, Germany: Kiel Univ., Inst. for Sci. Educ., 1992.
7. Mostow, J. Some useful design tactics for mining ITS data. *Proceedings of the ITS2004 Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, August, 2004.
8. Tanimoto, S. L., Carlson, A., Hunt, E., Madigan, D., and Minstrell, J. 2000. Computer support for unobtrusive assessment of conceptual knowledge as evidenced by newsgroup postings. *Proc. ED-MEDIA 2000*, Montreal..
9. Tanimoto, S., Carlson, A., Husted, J., Hunt, E., Larsson, J., Madigan, D., and Minstrell, J Text forum features for small group discussions with facet-based pedagogy. *Proc. CSCL 2002*, Boulder, CO (2002).
10. Tanimoto, S. L., Hubbard, S., and Winn, W. 2005. Automatic textual feedback for guided inquiry learning. *Proc. of the 12th Int'l. Conference on Artificial Intelligence in Education* (AIED 2005). Amsterdam.