Workshop on Machine Learning for User Modeling: Challenges

Edinburgh, Scotland
24th and 25th July

Organization Committee :

Colin de la Higuera, (Eurise, Université de Saint-Etienne, France)
Thierry Artières, (Lip6, Université Paris 6, France)

Program Committee :

Thierry Artières, (Lip6, Université Paris 6, France)
Mathias Bauer, (DFKI, Germany)
Patrick Gallinari, (Lip6, Université Paris 6, France)
David R. Hardoon, (University of Southampton, Great Britain)
Colin de la Higuera, (Eurise, Université de Saint-Etienne, France)
Samuel Kaski, (University of Helsinki, Finland)
Mikko Kurimo, (Helsinki University of Technology, Finland)
Thierry Murgue, (Eurise, Université de Saint-Etienne, France)
Lead Rezek, (University of Oxford, Great Britain)
Ingrid Zukerman, (Monash University, Australia)

Preface

In building effective interfaces or computer human interaction devices, a main limitation of traditional presentation design is the inability to meet individual user expectation at run-time. On-line design of individualised presentations that surpass the limits of the ”one size fits all” approaches can be made possible by user modeling techniques. Building models of users can be done through Machine Learning, but this requires techniques that are specific to the task.One particular issue is that the user models cannot remain static, in the sense that during the use of the intended interaction more knowledge can be gathered which should in turn be used to improve the user models. The knowledge from which the construction of the models can be made is many-fold: web logs, speech, images...

Machine learning is needed in order to construct the initial model, to allow reactivity and adaptivity, to build clusters of users, to allow the interface to find out more about the users, even if this does not pay off in the short term.

The workshop is organised by the PASCAL Special Interest Group in User Modeling for Computer Human Interaction. PASCAL (http://www.pascal-network.org/) is a Network of Excellence launched in January 2004 in the context of the 6th European Framework. 56 research teams participate in this network whose primary objective is to build novel tools for interfaces, in which it is expected that machine learning and pattern analysis have a role to play.

Table of contents

Motion-Based Adaptation of Information Services for Mobile Users

Mathias Bauer[1,2] and Matthieu Deru[2]

[1] DFKI, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
[2] mineway GmbH, Im Stadtwald, Geb. 34, 66123 Saarbrücken, Germany

Abstract. Adaptive information systems typically exploit knowledge about the user's interests, preferences, goals etc. to determine what should be presented to the user and how this presentation should take place. When dealing with *mobile* users, however, information about their motions—the places visited, the duration of stays, average velocity etc.—can be additionally exploited to enrich the user model and better adapt the system behavior to the user's needs. This paper discusses what type of positioning data and background knowledge is required to achieve such a motion-based adaptation of information provision and how it can be implemented using a variety of mostly standard machine-learning techniques.

1 Introduction

"Conventional" adaptive systems tend to characterize their users in terms of their preferences, interests, goals, etc. Location-aware systems additionally use the user's position to provide information or support that is somehow associated with a geographical point. When dealing with *mobile* persons, an additional, rich source of information about the user becomes available (at least in principle): the user's motion itself that lead her to her current position.[3] A *motion profile* derived from observations about the user moving in the physical space can reflect both the sequence of places visited by the user before reaching the current position—at various levels of abstraction—and additional features of the movement itself. The latter can be used to infer user characteristics such as her degree of commitment to a certain goal, her cognitive load and so forth. Combining both aspects of the motion profile, a mobile application can determine both what information (or service) is most appropriate to the user in the given situation and how it can (or should) be presented to her.

This paper presents a machine-learning approach to the computation of a two-part motion profile from low-level position data as provided e.g. by the increasingly widespread GPS receivers or any other positioning method based on WLAN, infrared beacons or similar [8]. The first part of such a motion profile refers to the various places visited by the user. Here a new method is introduced that allows a motion sequence to be quickly classified according to a number of dynamically acquired stereotypes. The second part characterizes the user's movement itself by referring to features such as speed, length of stays etc.

[3] Throughout this paper, users will be referred to in the female form.

The rest of this paper is organized as follows. Section 2 first discusses related work before Section 3 describes a learning approach that transforms e.g. GPS data to abstract characterizations of the user in terms of her movements through the physical space—the *motion profile*. Section 4 then describes how this information can be used in the adaptation of a mobile information service before Section 5 concludes with a summary and an outlook on future work.

2 Related Work

[5] describes an approach based on hierarchical Markov models that allows a person's daily movements to be learned from sample data, including changes of transportation means. To this end, the low-level position data are related to *trip segments* describing partial motion sequences which in turn are associated with high-level *goals* representing a potential target location. The result is a probabilistic model that allows to determine the probability of the user's future presence at some location and to detect deviations from her standard behavior. This approach only works in previously known environments that are completely represented as a graph with nodes representing locations and edges standing for the streets connecting them.

This multi-level approach overcomes limitations of [2] where conventional second-order Markov models are used. Using GPS logs, significant places are determined by taking into account the time spent at a certain location. The next goal is then predicted on the basis of the current and the previous location. With the abstraction level missing, the goal prediction cannot be refined as more information becomes available and deviations from standard routines cannot be detected. With respect to the prediction mechanism, this approach can be compared to the one taken in [1] where dynamic Bayesian networks are used to predict a user's next locations (and actions) within a multi-user dungeon, given her current state and information about her immediate past.

In the HIPS system [4] information about the user's movements through physical space is exploited to adapt the presentation of various pieces of art to the user's preferences. Similar to our approach described below, two aspects of the user's motions are taken into account: the set of artworks visited before and the user's *visitor style*. The latter is one of a set of four different stereotypical behaviors of museum visitors as described in [7]. These stereotypes capture typical visiting strategies within a museum and can be distinguished on the basis of speed, geometry of the paths followed etc. While knowledge about artwork seen before is used to determine which references and comparisons to include in the presentation at the user's next stop, the visitor style influences the style of the presentation (e.g. shorter, more focused presentations for less patient visitors and detailed explanations for those visiting at a slower pace). These prototypical visitor styles are specific to museum visitors *in general* and are not adapted to varying locations—a significant difference to the approach described in this paper.

Other approaches to determine a mobile user's attentiveness rely on input data derived e.g. from her speech input (such as articulation rate, disfluencies etc.) [6]. While this provides additional relevant information about the user, we will concentrate on location- and motion-related aspects here.

3 Computing a Motion Profile

This section discusses how an abstract characterization of the user and an estimation of what information might be relevant to her can be derived from observations about her movements between various locations.

Let $S = \langle p_1, ..., p_n \rangle$ be the sequence of positions passed by the user where $p_i = \langle x_i, y_i, z_i, t_i \rangle$. In the case of GPS data, x_i and y_i correspond to latitude and longitude values, respectively, z_i measures the current elevation, t_i is a time stamp.[4]

A *motion profile* $MP_S = \langle mod_p(S), mod_m(S) \rangle$ based on S consists of two components. $mod_p(S)$ encodes properties of the various positions contained in S. As will become clear, the abstraction level of mod_p is affected by the availability of relevant background knowledge. $mod_m(S)$, on the other hand, characterizes the motion itself—without referring to the positions actually visited—in terms of abstract features and mainly serves the purpose to form the basis for adapting the information presentation.

3.1 Identifying Relevant Information

The ultimate objective of modeling the user's motion is to derive recommendations of what information might become useful or relevant to her in the foreseeable future. In the context of location-based information systems considered in this paper, the relevance of some piece of information is naturally connected to the places a user is likely to visit. In order to arrive at a reliable estimation of this relevance, there basically exist two different approaches. We can either

- compare the user's behavior to that of other users, thus deriving a kind of collaborative recommendation; or
- produce a prediction model from structural properties of the user's motion itself.

Collaborative Recommendation Rules Assume we are given some background knowledge in the form of annotations to the locations a user has visited.[5] This knowledge comes in the form of unique identifiers for the various GPS coordinates (e.g. "Kulturcafe") or additionally contains classification information about the type of this place (e.g. restaurant, department store).

In either case, the user's motion history S can be reduced to the set of (named) locations or location types visited. Collecting such information across all users allows the derivation of association rules using standard techniques known from market basket analysis (e.g. the well-known *a-priori* algorithm or the *CAPRI* algorithm in case the temporal order of visits is additionally taken into account).

Depending on the background knowledge available, these rules then have the form

- "If the user has visited 'Kulturcafe' and (then) 'Sport Scheck', then she will also visit 'H&M'. " (*confidence* = 40%, *support* = 85) or

[4] Due to lack of space, preprocessing steps required to cope with imprecise or incorrect measurements will not be discussed here.

[5] In particular, places where she spent some time instead of quickly passing them are considered important.

– "If the user has visited a shoe shop and a department store, then she will also visit a restaurant." (*confidence* = 27%, *support* = 145).

Confidence and *support* are measures for the quality of such rules that represent their accuracy and the number of occurrences in the past, respectively.

Given these rules, information items connected with the places or types of places occurring in the conclusions can be considered relevant, the reliability of these estimations being determined by the rule quality measures. $mod_p(S)$ in this case corresponds to the set of rules applicable.

Abstraction of the Position History Another way of predicting the relevance of some piece of information is trying to extrapolate the user's route observed so far and select the information associated with the locations to be possibly visited in the near future. In Section 2 we already discussed HMM-based approaches for predicting a user's presence at a particular place. Here we present a simple algorithm for quickly assigning an observed motion sequence to one of a number of prototypical patterns which forms the basis for relevance assessment of information.

The basic idea is to collect a number of motion sequences in a certain area and then determine clusters of similar motions. Each cluster then represents a particular type of navigation behavior that differs significantly from the others. To determine these clusters, we need a distance measure between motion sequences.

Let $S_1 = \langle p_1^{(1)}, ..., p_n^{(1)} \rangle$ and $S_2 = \langle p_1^{(2)}, ..., p_m^{(2)} \rangle$ be motion sequences. Then the distance between S_1 and S_2 can be defined as

$$dist_s(S_1, S_2) \; = \; w_e \cdot min(dist_e(S_1, S_2), dist_e(S_2, S_1)) + w_o \cdot disorder(S_1, S_2).$$

Here $dist_e(S_1, S_2)$ corresponds to the *edit distance* between both sequences, $disorder$ measures the number of positions that S_1 visits in a "wrong" order as compared to S_2, and w_e and w_o are weights controling the influence of both factors on the overall distance measure. In particular, $w_o = 0$ means the directions of motion sequences can be ignored.

Remarks:

– The edit distance is a concept known e.g. from string comparisons. It measures the minimum number of insert, delete, and replacement operations required to transform its first argument into its second argument where each such operation contributes a certain "penalty" to the overall distance measure. In our case, this amounts to replacing one position $p_i^{(1)}$ from S_1 by the nearest corresponding position $p_j^{(2)}$ from S_2, thus yielding a penalty with the value of the Euclidean distance between both positions. The penalties for replacement and deletion correspond to 50% of the maximum Euclidean distance that is possible in the area under consideration. $dist_e$ can be efficiently computed using dynamic programming.[6]

– The $disorder$ measure adds a penalty for each pair of positions $p_i^{(1)}$ and $p_{i+1}^{(1)}$ that are mapped onto positions $p_k^{(2)}$ and $p_l^{(2)}$, resp. where $t_l^{(2)} < t_k^{(2)}$, i.e. where the temporal order in S_2 differs from that in S_1.

[6] See e.g. www.csse.monash.edu.au/~lloyd/tildeAlgDS/Dynamic/Edit.

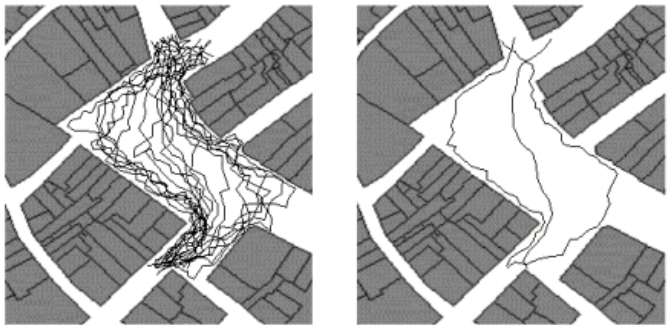


Fig. 1. Typical motion patterns in a city (left) and the medoids of three clusters (right).

With a distance measure defined this way, it is possible to determine clusters of similar motion sequences using a standard algorithm such as k-means.[7] Each cluster can be compactly represented by its *medoid*, the element with the smallest average distance to all cluster members. Figure 1 depicts a number of motion sequences observed (left part) and the medoids of 3 clusters found in these data (right part).

Given this compact representation of all the training sequences, a newly observed sequence S_0 can be efficiently classified by computing its distance to all medoids available and associating it with the nearest one. The medoid found this way then serves as the prototypical representation of the class of similar motions S_0 belongs to. Note that the cluster membership of S_0 can change over time and thus has to be re-confirmed regularly. This classification—including the set of visited places predicted—then constitutes $m_p(S)$, the location-dependent part of the motion profile. Preselection of relevant information can either take place by searching for information associated with the locations predicted by the cluster membership or by combining this prediction with the recommendation rules as discussed above, thus applying an additional filter to the candidate information items.

Remark: First experiments showed that this type of classification of the user's motion works best when only the last n minutes of the user's motion are taken into account. With increasing size of this time window (in particular in the order of magnitude of several hours), relevance predictions tend to be become unreliable.

3.2 Abstract Motion Features

As mentioned above, the HIPS project [4] distinguished between only 4 prototypical motion patterns specific to the museum visit domain. In general, however, it is desirable to be able to identify a greater variety of persons moving in a certain space. Outside a more or less single-purpose area like a museum, people can have all sorts of goals, temporal constraints, or completely idiosyncratic navigation behaviors, depending on their knowledge of the location, preferences for quiet or busy places, shopping or recreational areas etc. In order to be able to appropriately address a wide spectrum ranging

[7] The Python-based data mining environment *Orange*, for example, provides a free implementation available at http://magix.fri.uni-lj.si/orange.

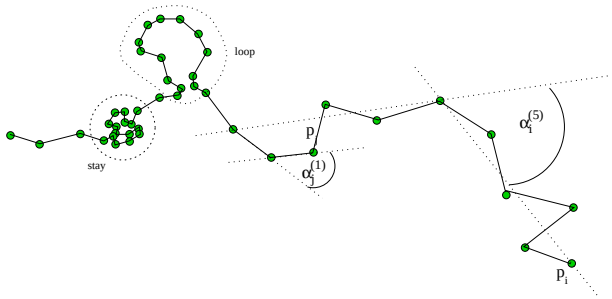


Fig. 2. Computing abstract features in an unfiltered motion sequence.

from people strolling leisurely through a downtown pedestrian zone to people desperately looking for a certain place under great time pressure, it is necessary for the system to reliably identify these different types of behavior.

The criteria for this distinction can be directly derived from a motion sequence S. They characterize motion sequences in terms of *loops* (including average duration and velocity, frequency per km etc.), *stays* at certain positions (average duration, prevalent type of places visited etc.), and *deviations* from the previous direction, measured at various time scales (see Figure 2).

Additionally, we also consider the user's *velocity* (both average and maximum). If available, the quotient of velocity observed in S and the user's normal average speed is taken into account.

The final class of features we consider deals with the user's goal-orientation. We measure the user's deviation from her previous direction at a short-term, mid-term, and long-term scale. Figure 2 depicts the computation of a short-term and mid-term deviation. The mid-term deviation, for example, is defined to be the angle between the motion during the last 5 time units (usually seconds) and the motion of the 5 time units before that. In general, we compute the angles

$$\alpha_i^{(j)} \;=\; angle(\overline{p_{i-2j}p_{i-j}}, \overline{p_{i-j}p_i})$$

where $j \in \{1s, 5s, 20s\}$ and $\overline{p_a p_b}$ is the vector connecting p_a and p_b.

Remarks:

- Depending on the quality of the maps available, the number of "deliberate" changes in direction—i.e. those not enforced by a building or a street—can be determined.
- Whatever features are actually chosen to characterize a motion, it is important for them to be incrementally computable.
- Additional information such as the local weather can help to correctly interpret the values so determined. A heavy shower tends to speed up even the most relaxed ambler.
- If a limited time window is being considered for $mod_p(S)$, then the same temporal limits have to be observed here.

Once a sufficient number of such feature vectors resulting from observed motion sequences is available, there exist two ways to create and characterize classes of mobile

users. In a *supervised* mode, an expert (or the user herself) labels each feature vector with a certain class (e.g. ambling, walking). Using these training data, a decision tree can be learned (using an algorithm such as C4.5) that characterizes each of these classes using the set of features available. In an *unsupervised* mode, a clustering method such as k-means (see above) is applied to the unlabeled data in order to form clusters of similar feature vectors which will then be labeled with some understandable class name.

Classifying the user w.r.t. the categories so identified amounts to computing the current values for the features as described above and determining the class membership of the resulting feature vector $mod_m(S)$ using the decision tree (in the supervised learning case) or its cluster membership (in the unsupervised case). Knowledge about the user's class membership can be used to adapt the presentation of the information determined on the basis of $mod_p(S)$ to her estimated cognitive load, attentiveness, estimated time pressure etc. (see also [6]). Additionally, information that appears inappropriate in the current situation can be filtered out. For example, even the most relevant product offer is little helpful when the user is obviously speeding up to try and catch her bus.

In first experiments, GPS data were collected by several persons moving in downtown Saarbrücken (Figure 1 depicts part of these sequences). While the training data available so far were insufficient to cover all kinds of motions conceivable, at least some of the features discussed above appear to be highly relevant to discriminate between ambling persons, persons moving in a goal-directed way from A to B, and persons without sufficient knowledge of the location.

4 An Application Scenario

The techniques introduced above are currently being investigated in the context of *Saarland Unwired*, a project dealing with the installation of numerous WLAN hotspots throughout the city center of Saarbrücken. The basic idea is to provide the mobile user with up-to-date information about the closer neighborhood of the hotspot she's currently using. As these hotspots do not cover the entire city, there are regions in which the user is disconnected and no high-level information about her interests can be gathered (while at the hotspot, her browsing behavior is a rich source of information). While walking in those parts not covered with WLAN, it is still possible to collect information about the user[8] in the form of motion sequences from which motion profiles as discussed above can be derived. These profiles are then used to provide a personalized collection of information items such as news regarding the user's current environment, navigation aids, or marketing messages from stores in the neighborhood. Here mod_p is used to identify what information might be relevant to the user, while mod_m adapts the presentation to her perceived state and attentiveness.

5 Conclusions and Future Work

We presented an approach to the personalization of location-based information services for mobile users based on the observation and classification of their motions in space.

[8] Provided she is carrying a GPS receiver and agrees to be tracked.

The motion profiles determined using a variety of machine-learning techniques serve the purpose to both identify *what* information might be relevant to the user in the current situation and find out *how* this information can be best presented to her.

As mentioned above, first experiments with real data indicate that the motion profiles so computed actually allow prototypical patterns to be detected in the motion sequences (see e.g. Figure 1) and various types of mobile users to be distinguished.

While the WLAN-based information service mentioned above is an obvious candidate for the application of such techniques, the overall goal is the integration of the motion-profile approach into SPECTER [3], a context- and affect-aware mobile personal assistant in particular for instrumented environments. It aims at observing and recording as much as possible about its user—including actions, emotions, and movements—in order to create a kind of episodic memory called *personal journal*. The latter serves the purpose to support the user in information access and decision making. The experience from the WLAN project will help limit the number of features that have to be recorded in order to arrive at a reliable classification.

To this end, more GPS training data have to be collected and processed in the way discussed above. Additionally, a reasonable set of classes of mobile users as expressed in mod_m as well as optimal values for the various system parameters used have to be determined.

Acknowledgments

This research was supported by the German Ministry of Education and Research (BMB+F) under grant 524-40001-01 IW C03 (project SPECTER). Thanks to all project members.

References

1. D. W. Albrecht, I. Zukerman, and A. E. Nicholson. Bayesian Models for Keyhole Plan Recognition in an Adventure Game. *User Modeling and User-Adapted Interaction*, 8(1-2):5–47, 1998.
2. D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.
3. M. Bauer. Transparent User Modeling in SPECTER . In *Proceedings of the 7th International Conference on Work with Computing Systems (WWCS 2004)*, 2004.
4. G. Benelli, A. Bianchi, P. Marti, E. Not, and D. Sennati. HIPS: Hyper-Interaction within Physical Space. In *Proceedings of IEEE Multimedia Systems '99, International Conference on Multimedia Computing and Systems*, 1999.
5. L. Liao, D. Fox, and H. Kautz. Learning and Inferring Transportation Routines. In *Proceedings of AAAI-04*, pages 348–353, 2004.
6. C. Müller, B. Grossmann-Hutter, A. Jameson, R. Rummer, and F. Wittig. Recognizing time pressure and cognitive load on the basis of speech: An experimental study. In *Proceedings of the 8th International Conference on User Modeling (UM2001)*, pages 24–33, 2001.
7. E. Veron and M. Levasseur. *Ethnographie de l'exposition*. Bibliotheque publique d'Information, Centre Georges Pompidou, 1983.
8. V. Zeimpekis, G.M. Giaglis, and G. Lekakos. A Taxonomy of Indoor and Outdoor Positioning Techniques for Mobile Location Services. *ACM SIGecom Exchanges*, 3(4):19–27, 2003.

Chronological Sampling for Email Filtering

Ching-Lung Fu[2], Daniel Silver[1⋆], and James Blustein[2]

[1] Acadia University, Wolfville, Nova Scotia, Canada
[2] Dalhousie University, Halifax, Nova Scotia, Canada

Abstract. User models for email filtering should be developed from appropriate training and test sets. A k-fold cross-validation is commonly presented in the literature as a method of mixing old and new messages to produce these data sets. We show that this results in overly optimistic estimates of the email filter's accuracy in classifying future messages because the test set has a higher probability of containing messages that are similar to those in the training set. We propose the k-fold chronological cross-validation method that preserves the chronology of the email messages in the test set.

1 Introduction

Our research into spam email filtering began with an investigation of existing filtering systems that are based on learned users models. Often, we found the reported accuracy of these systems to be overly optimistic [1]. We found it difficult to create models with the same high level of accuracy as published when using the same or independent datasets. Other authors have made similar observations [2]. Although much of the difference between the earlier evaluations and ours can be attributed to differences in the mix of legitimate and spam emails in the datasets, we speculated that another important factor is the method of testing that is employed.

Our work with machine learning models for spam filtering has shown that time is an important factor for valid testing; i.e. the order of incoming email messages must be preserved so as to properly test a model. Unfortunately, many results reported in the literature are based on a k-fold cross-validation methodology that does not preserve the temporal nature of email messages. It is common practice with cross-validation to mix the available data prior to sampling training and test sets so as to ensure a fair distribution of legitimate and spam messages [3]. However that practice, by mixing older messages with more recent ones, generates training sets that unfairly contain future messages. The mixing ignores an important dynamic feature of spam e-mail, namely that the generator (spammer) is an adversary who incorporates information about filtering techniques into their next generation spam [4].

If the temporal aspect is not considered, the performance of the model on future predictions may be significantly less than that estimated. A fair test set should contain only messages received after those used to develop the model. Commonly used data sets available on the web, such as the Ling-Spam corpus [5], do not even contain a time stamp in the data. We present a comparison of a spam filtering system tested using cross-validation and a temporal preserving approach.

⋆ Corresponding author (danny.silver@acadiau.ca)

2 Background

The k-fold cross-validation method is a standard method for comparing different models [3]. In k-fold cross-validation the dataset is divided into k subsets of approximately equal size. Model generation and testing is repeated k times. Each time a different subset is selected as a test set and the remaining subsets are selected for training. In some machine learning algorithms (*e.g.* inductive decision trees and artificial neural networks) it is necessary to select a part of the training set as a validation set to reduce the likelihood of over fitting. Each subset can be in the test set exactly once and in the training set $(k-1)$ times. Before splitting the dataset, it is common to randomly sort all examples in the dataset, to ensure that they are evenly distributed, before creating the k subsets. The intention of cross-validation is that it will better estimate the true accuracy of the resulting models, based on the mean accuracy calculated over the k evaluations. In the addition, the standard deviation around the mean can be used to produce confidence intervals and to determine the statistical significance between different models, or machine learning algorithms.

Consider the effect of randomly mixing the legitimate and spam messages prior to undertaking a k-fold standard cross-validation (SCV). When the datasets are mixed, possible future examples are injected into the training set thereby providing the model with an unfair look at future features. Figure 1 illustrates the problem of mixing old and new examples under SCV: If **A**, **B**, and **C** are three main types of examples in the dataset, let **A** be the oldest and **C** the newest. In SCV, all three types of examples are evenly distributed in the training set, validation set, and test set. This distribution provides the best opportunity for a model to perform well on the test set. However, in reality, type **A** and **B** have the highest probability of being available in the data set during the time of model development. **C** may not have appeared until after **A** and **B**. Thus, the mixing of examples in SCV provides an unrealistic set of future examples in the training set. We claim that this is one of the major reasons why many of the reported results on spam filtering are overly optimistic.

3 Chronological Cross-Validation

We propose k-fold Chronological Cross-Validation (CCV) as a more realistic evaluation method of data for any temporally-sensitive model (including e-mail) that selects the training and test sets while the data is in chronological order. The test set will then simulate the classification of future messages based on a model produced from prior messages and, therefore, the test set accuracy will better estimate the true accuracy of the email filter. CCV maintains the chronology of the email messages as the evaluation process moves along the chronological order of the data set. A ten-fold CCV is depicted in Fig. 2. We propose that this new cross-validation approach will reduce the probability of over-estimating the effectiveness of the model. CCV method is as follows:

1. The data set is sorted chronologically;
2. The data set is divided into $2k-1$ blocks;
3. k folds (or repetitions) are undertaken as follows:

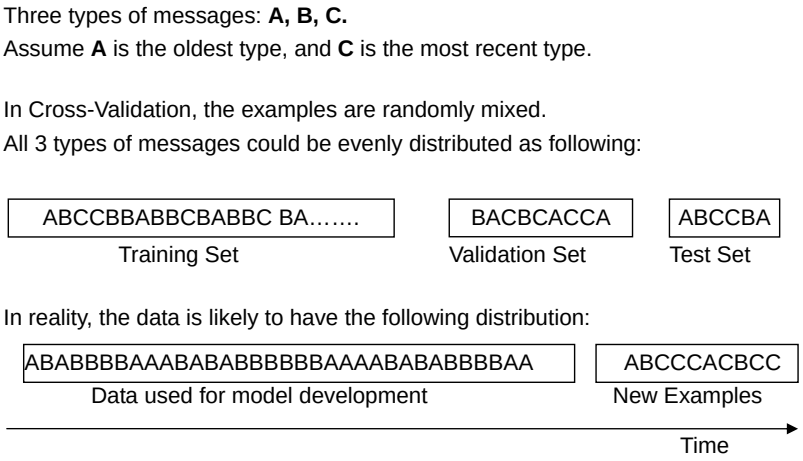


Fig. 1. A problem with the random mix of examples in standard Cross-Validation.

(a) In the first repetition, blocks 1 to k are selected for evaluation;
(b) The first $k-(v+1)$ blocks are selected as the training set, the following v blocks are selected as validation set, and the last block is reserved for the test set;
(c) Each later fold advances one data block in chronological order and the oldest data block is abandoned (for example in Fig. 2, in the second repetition, block 1 is abandoned and blocks 2 to $k+1$ are selected for evaluation);

4. The procedure is repeated k times, until data block $2k-1$ has been tested.

4 Empirical Studies

Two studies were undertaken using one email dataset called AcadiaSpam, collected from a single individual from January to May 2003, working at Acadia University. The set consisted of 1454 spam messages and 1431 legitimate messages. The initial study was conducted with a subset of these emails using one experimental design and the second was conducted with all of the data using a slightly different design.

4.1 Experiment 1

The initial study was undertaken during the development of a prototype intelligent email client. A small subset of the data was chosen so as to quickly determine if the proposal had merit for larger scale testing. The objective was to determine the severity of SCV over-estimates the true accuracy of hypotheses as compared to CCV.

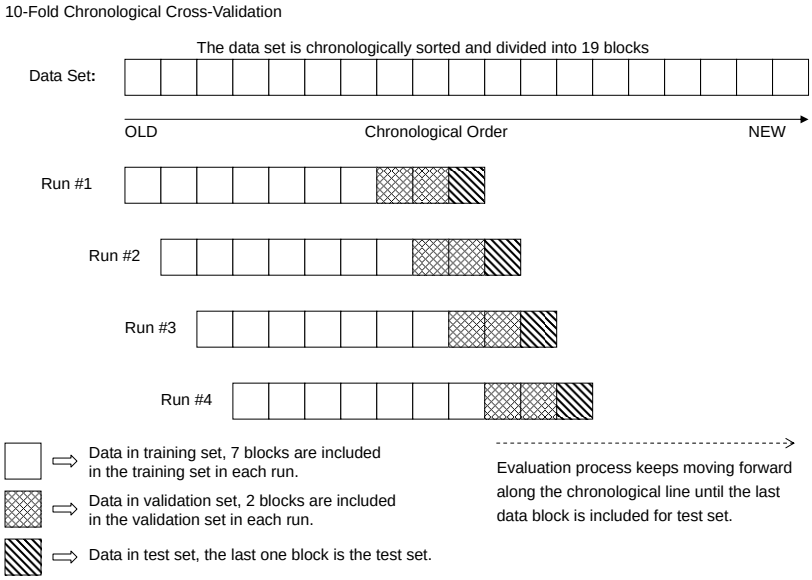


Fig. 2. A 10-fold Chronological Cross-Validation.

Method. The first 500 legitimate messages and 500 spam messages were selected from the AcadiaSpam dataset and stored in chronological order. A $k = 6$ was chosen for this initial experiment; therefore, 6 models were repeatedly developed and tested against their respective test sets under each method.

A block of 500 messages was chosen for each repetition starting at the earliest email message. On each repetition, the block was moved 100 messages forward in the chronology. From each block of messages, 300 were selected for a training set, 100 for a tuning set, and 100 for a test set. Two data sampling methods were used to create the sets. For the SCV method, the message data for the three sets were randomly chosen. For the CCV method, the most recent messages were selected for the test set and the remaining messages randomly distributed to the training and tuning sets. The tuning set was used to prevent over-fitting of the neural network to the training set.

Prior to model development, 200 features were extracted from each message using traditional information retrieval methods (removing stop words, performing word stemming, and collecting word statistics). A standard back-propagation neural network with a momentum term was used to develop the spam filter models [3]. The network had 200 inputs, 20 hidden nodes and 1 output node. A learning rate of 0.1 and a momentum factor of 0.8 were used to train the networks to a maximum of $10,000$ iterations.

Since the output of the network ranges from 0 to 1, messages with output greater than 0.5 were classified as legitimate, and all others were classified as spam. The models were evaluated based on their accuracy of classification, precision and recall of spam email messages. The calculations of accuracy, recall and precision follow [2].

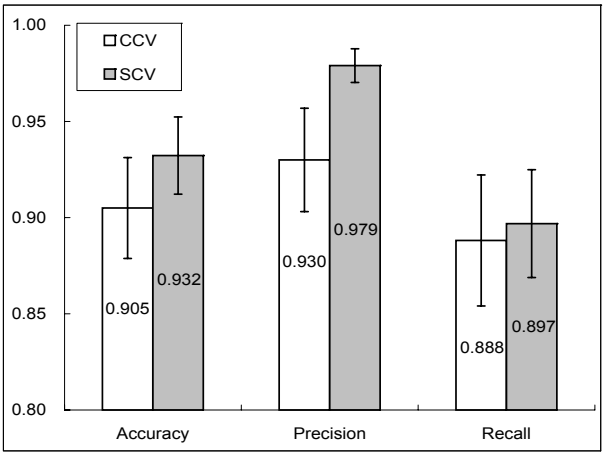


Fig. 3. Comparison of SCV and CCV on the 1000 AcadiaSpam dataset with $k = 6$.

Results and Discussion. Figure 3 shows that the SCV method estimates the true accuracy of the models to be 0.93. That figure is, on average, 2.5% higher than the CCV method's estimate ($p = 0.238$, based on a paired t-test). Similarly, the SCV method consistently produces the higher precision and recall models. The difference in the precision values is most significant at 5% ($p = 0.0249$). Our conjecture is that the SCV method unrealistically allows the modelling system to identify features of future spam emails. The SCV method over-estimates its performance on the test messages because the training set has a higher probability of containing messages that are similar to those in the test set. The CCV method generates a less accurate but more realistic model because the testing procedure simulates the classification of future incoming messages.

A potential flaw in this preliminary study is that it does not use a standard cross-validation method, as not all data was used in every repetition of SCV. This was done to keep the number of examples used by the two systems the same during each repetition. Although we suspect that a more standard SCV would further increase the performance gap between SCV and CCV we undertook a second study to investigate this potential concern about the validity of our results.

4.2 Experiment 2

The second study used all of the available data in a more traditional SCV approach in which all data is used in every repetition. As in the initial study, the objective was to show that SCV over-estimates the true accuracy of hypotheses as compared to CCV.

Method All messages in the AcadiaSpam dataset were used in this experiment (1454 spam messages and 1431 legitimate messages). For SCV, the messages were randomly ordered and divided into $k = 10$ blocks each consisting of 143 legitimate and 145 spam messages. Each repetition used 7 blocks as the training set, two blocks for a validation set, 1 block as a test set.

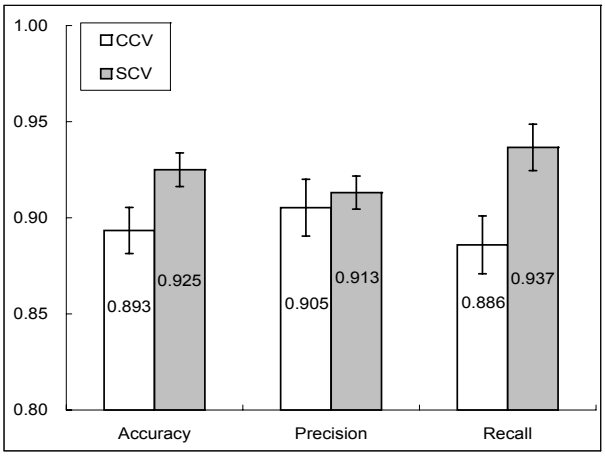


Fig. 4. Comparison of SCV and CCV on the 2880 AcadiaSpam dataset with $k = 10$.

For CCV, the AcadiaSpam dataset was chronologically sorted. Legitimate messages and spam messages were divided into 19 ($2k-1$, where $k = 10$) blocks. Each block has 151 messages consisting of approximately 75 legitimate messages and 76 spam messages. Each repetition of the experiment used a window of $k = 10$ blocks of messages starting with the oldest block. From these 10 blocks, the 7 oldest blocks are used as the training set, the next 2 blocks for the validation set, and the most recent block for the test set. For each new repetition of the experiment, the oldest block was removed from the window of 10 blocks and the next chronological block was added. Note that every message was used by both methods, however fewer examples were used in each repetition by CCV than by SCV. All other aspects of the method were the same as in the first experiment.

Results and Discussion. The results of this larger experiment, shown in Fig. 4, support the findings of the initial experiment. The SCV method produces hypotheses with superior performance in all 3 measurements (accuracy, recall, precision) as compared with CCV. The difference in mean accuracy between the two methods was found to be 3.1% ($p = 0.00057$, based on a paired t-test) up from 2.5% in the initial study. As in the initial study, the SCV method produces the highest precision and recall statistics. In this case, no significant difference was detected in the precision statistics ($p = 0.38$) but the recall statistics differed substantially ($p = 0.000067$).

Although the difference in the statistics for these methods could be attributed solely to the smaller training sets used to develop the CCV models in this study, the results of the initial study do not support that conclusion. When both methods used the same size training sets, the SCV method was still found to over-estimate the model's performance. We conclude that the difference in the statistics is caused by unrealistic mix of old and new examples in the training sets used to develop the SCV models.

5 Related Work

We have recently discovered work by Crawford *et al* [6] that agrees that the k-fold SCV method is unrealistic given the temporal nature of email. They describe a model development approach that accumulates the older messages in the training set and selects only the most recent data for the test set. This testing approach is in accord with how a real email filter must perform; therefore, it should provide a fair evaluation of the model's effectiveness. Beyond this the research emphasis and approach differs from our work. Crawford *et al* focus on model development over time whereas we are interested in a cross-validation method for estimating the true error of a model at any one point in time. Our CCV method purposely abandons older blocks of examples as it moves through its repetitions so as to better estimate model performance on a variety of examples.

6 Summary and Conclusion

We have considered the importance of maintaining the temporal nature of incoming email messages when developing a user model for email filtering. Although the k-fold SCV is commonly presented in the literature as a method of randomly mixing examples to produce training and test datasets, we have demonstrated that the method results in overly optimistic estimates of an email spam filter's accuracy in classifying future messages. Our conjecture is that the SCV method is inappropriate because it allows the modelling system to unfairly identify features of the test set spam emails. The SCV method over-estimates model performance on future messages because the training set has a higher probability of containing messages that are similar to those in the test set.

We propose the k-fold Chronological Cross-Validation (CCV) method as a step towards more realistic estimates of model performance. CCV generates less accurate but more realistic models because the testing procedure more properly simulates the classification of future messages. The CCV method can be used to more properly evaluate any complex user model that will change over time. Thus, it can better estimating a models ability to deal with *concept drift*: the change in a user model over time due to subtle changes in the user, their environment, or both [7]. More broadly, the CCV method can be applied to any learning task where the order of examples must be preserved.

The CCV method highlights the fact that more examples are needed to properly evaluate a user model when the preservation of example order is a requirement. The size of the time window, which depends on k, must be large enough to develop good models but small enough to allow sufficient blocks for cross-validation. Window size must also be sensitive to the mix of training examples and is likely to be different for each individual. These are a couple of the open problems that we would like to investigate in future research.

References

1. Clark, J., Koprinska, I., Poon, J.: A Neural Network Based Approach to Automated E-Mail Classification. In: Proceedings IEEE/WIC International Conference on Web Intelligence (WI2003), Halifax, Canada (2003) 562 – 569

2. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., Stamatopoulos, P.: Learning to Filter Spam E-Mail: A Comparison of a Naïve Bayesian and a Memory-Based Approach. In: Proc. of the Workshop on Machine Learning and Textual Information Access, 4th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000), Lyon, France (2000) 1 – 13
3. Mitchell, T.: Machine Learning. McGraw Hill, New York, USA (1997)
4. Stern, H., Mason, J., Shepherd, M.: A linguistics-based attack on personalised statistical e-mail classifiers. Technical Report CS-2004-06, Dalhousie Univ. (2004)
5. Androutsopoulos, I.: Ling-spam corpus (2000) ⟨`http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz`⟩.
6. Crawford, E., Kay, J., McCreath, E.: Automatic induction of rules for e-mail classification. In: The Sixth Australian Document Computing Symposium, Coffs Harbour, Australia (2001)
7. Widmer, G.: Combining Robustness and Flexibility in Learning Drifting Concepts. In: Proceedings of the 11th European Conference on Artifcial Intelligence (ECAI-94), Wiley, Chichester, UK (1994) 468–472

First Order Logic for Learning User Models in the Semantic Web: Why Do We Need It?*

François Bry[1] and François Jacquenet[2]

[1] University of Munich, IFI,
Oettingenstrasse 67, D-80538 München,
Germany
francois.bry@ifi.lmu.de
[2] University of Saint-Etienne, EURISE,
23 rue Paul Michelon, F-42023 Saint-Etienne,
France
Francois.Jacquenet@univ-st-etienne.fr

Abstract. In this paper we claim that learning in a first order logic framework is crucial for the future of user modeling applications in the context of the Semantic Web (SW in the remaining of the paper). We first present some works that have currently been done for designing first order logic based languages, for reasoning in the SW. In the context of user modeling in the SW, it would then be relevant to use such languages to model user's behaviors and preferences. We show that discovering knowledge in the context of such languages could be done using multi-relational data mining that has already provided efficient prototypes. Nevertheless, some work remains to be done in order to use them in that context and we give some directions for that purpose.

1 Introduction

The main goal of the Web at its early years was to be able to display some information in Web pages, across a network, using the very basic HTML language and the HTTP protocol. Then, people tried to make it more extensible designing the XML language and opening the door to interoperability of heterogeneous systems across the Web. To do so, they introduced the concept of Web services and designed new protocols and languages such as UDDI[3] (Universal Description, Discovery and Integration), SOAP (Simple Object Access Protocol) [33] or WSDL (Web Services Description Language) [17] for example. Nevertheless, the Web still remained syntactic, and the need for a more semantical view of it became urgent. Hence, T. Berners-Lee proposed a new vision of the Web he called the *Semantic Web* [9, 10]. He defined the SW as "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" and proposed the famous SW Layers shown at Figure 1. The first layer is the one of pure XML, where we can specify the

* This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the author's views.

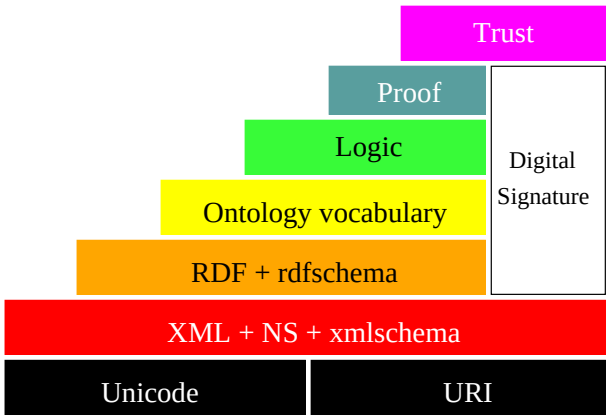


Fig. 1. The Semantic Web Layers

syntactic structure of files using XML Schema [23]. The next layer is dedicated to RDF (Resource Description Framework) [5] and RDF Schema [14]. It provides a way to specify labeled-graph data models on top of XML. The next layer is concerned with the language used for specifying ontologies vocabulary. Several propositions have been made for that purpose such as DAML+OIL [25] and OWL[32]. On top of this layer is a logic layer. Berners-Lee vision of the SW was that ”we need ways of writing logic into documents to allow such things as, for example, rules the deduction of one type of document from a document of another type; the checking of a document against a set of rules of self-consistency; and the resolution of a query by conversion from terms unknown into terms known” [9]. This layer enable the SW to deal with rigorous semantics and to use some proof engines. That leads to the ultimate layer that concerns trust which is probably the core of a reliable, efficient SW for the future.

At the moment, specifications have been proposed to the W3C for the first three layers and some standards or recommendations have been adopted. Over that, no standard or recommendation has currently been adopted and moreover, this remains a large process to being made before being able to specify some standards.

The European Network of Excellence REWERSE[4] (REasoning on the WEb with Rules and SEmantics) is one of the Networks of Excellence working in the context of the SW. As said in its objectives it aims at developing a coherent and complete, yet minimal, collection of inter-operable reasoning languages for advanced Web systems and applications. A huge bibliographic study has been done by various teams of the Network and can be found on the REWERSE Web site. It is not the purpose of this paper to give an exhaustive survey of the domains of use of systems that have been designed today for reasoning on the Web. Rather we just want to point out the fields that are actually investigating the use of first order logic for the SW.

[4] http://rewerse.net/

Next section rapidly presents how rule-based languages are currently being designed for the SW. Then we present some subdomains of the SW, where user models could be built using first order logic. Then section 4 talks about existing techniques that make it possible to learn in the context of first order logic. Finally, before concluding, section 5 points out some topics that need to be addressed in order to be able to use machine learning techniques for user modeling in the SW.

2 How could we model things in the Semantic Web ?

Rules have been identified as a design issue by T. Berners-Lee in [8]. There is now a large consensus on using rules to implement the logic layer of the SW. Thus, the design of rule-based languages for the SW has been the subject of many researches for several years. The most advanced proposition nowadays is certainly the RuleML one [13, 39]. RuleML results from the Rule Markup Initiative that explores rule systems suitable for the Web, their syntax, semantics, efficiency, and compilation. RuleML comes from several other rule languages and the reader may refer to the RuleML Web site[5] for more details. We have no place here to present this language but we can say that RuleML allows the definition of many sublanguages ranging from very general equational logic to more specific datalog like sublanguages. Of course the various specifications of rule languages tried to integrate the logic layer and the ontology layer. That's for example the purpose of SWRL [26] which aims at combining a relatively simple subset of RuleML with OWL to express Horn-like rules that are tightly integrated with OWL ontologies. These tools are emerging from the Web community and they are going to be used intensively in the near future.

3 What could be modeled in the Semantic Web?

Now that we know that rule-based languages exist, what could we do with them? More precisely, what could we model, in the SW, that is related to users?

One of the things we could do in the SW is to specify policies [40]. In fact, being able to do that is crucial in many fields, not only in the SW. It may be notably related to the notion of privacy. In the context of databases, Agrawal has defined the concept of hypocratic databases [2]. In the context of SW we could define the concept of hypocratic Web.

Thus, being able to model policies, we could try to discover policies from user's data. In a closely related domain for example [34], we tried to discover protocols used by a community of agents observing the way they were conversing. Nevertheless, we used Grammatical Inference techniques and that did not allow us to take into account the context of agents and the content of messages exchanged between them. In the field of SW, that could be considered as a universe of agents, it would be crucial to be able to take into account the background knowledge of each agent and the relations between them.

[5] http://www.ruleml.org

In a closely related domain, preference management is another topic that can obviously be addressed in the context of user modeling in the SW. Several workshops have been dedicated to preferences nowadays, for example at AAAI'2000 [28], at a Dagsthul Seminar in 2004[6], and a workshop at IJCAI will be dedicated to this topic this year. It is not the place here to detail the large amount of papers published in this domain, but we can say that many of them present some systems based on first order logic such as [41, 38]. In the field of SW, we can cite for example [19, 20] which propose to use Adaptive Hypermedia techniques to the SW.

Many prototypes have been proposed in the domain of Web usage mining in order to discover preferences of users while navigating on the classical Web. In the context of Web content mining, many recommendation systems have been designed for many years [1]. In the context of SW, new methods will have to be designed in order to take into account the richness of information provided by rule-based languages.

User modeling could also be used in order to adapt Web Services to each client. Indeed, a Web Service registry could observe the requests that are done to it and try to adapt the services it proposes to its clients. At the moment, when one want to disseminate a Web Service, one can register it for example using some registries such as UDDI. The problem is then for users. How can they know that such a service exists and is just the one they need. We face here the same classical problem that was pointed out in the domain of software engineering for reusability. Trying to solve that for example, [29] proposed a logical framework to deal with the discovery of Web services. In fact, some data mining techniques have already been used in order to discover frequently reused software components by programmers. We could imagine here that a similar process could be used for Web services.

Trust on the Web has been identified as one of the major difficulties of the SW. Various strategies can be used in order to manage trust on the Web [37]. In fact we may distinguish policy-based trust management and reputation-based trust management. In the first case we can try to use machine learning techniques as we said previously in order to learn some policies in some open environments. In the second situation, machine learning for user modeling could have a great impact by being able to learn models of good/bad reputation Semantic Web services. The reader may refer to [36] for an exhaustive presentation of trust management systems.

4 Some Techniques are already alive but...

If we want to have the advantage of a first order logic representation and reasoning we should try learning models expressed with the first order rule-based languages that are currently designed for the SW.

Inductive Logic Programming [35] has emerged at the beginning of the 90's. Combining Machine Learning and Logic Programming, it has first led to the design of general purpose systems. Then, since the beginning of 2000's, people have worked

[6] http://www.dagstuhl.de/04271/Materials/

on Multi-Relational Data Mining [22] (MRDM in the remaining of the paper). Thus, many systems have been developed to shift classical data mining algorithms to first order logic. For example TILDE [11] extends the C4.5 algorithm, WARMR [18] extends the apriori algorithm for mining association rules, [4] proposed to extend Markov models to Relational Markov models, etc. A short review of such systems may be found in [24].

MRDM can better deal with Web Mining than actual systems that are not based on first order logic. Indeed, [21] explains for example that MRDM is particularly well suited to Web mining because of its ability to deal with its graph structure. We show for instance in [27] that learning user preferences on the Web with an ILP system leads to the discovery of knowledge that put some links forward between interesting pages, people, etc, thanks to first order logic.

Learning preferences with pure ILP techniques may nevertheless be sometimes impossible. Indeed, there are cases where we have to learn models that express some constraints between objects. For example, in [7] we wanted to model the way people designed newspapers, to do so we had to design an inductive constraint logic programming system. Indeed we wanted to learn the way people laid out, on a page, rectangles of text, image or advertisement; that is we had to learn constraints between objects displayed on that page. Nevertheless constraint ILP systems have not been widely used until now due to the strong biases the users have to design, which is a tricky task.

There are many situations were some data are in a sequential form. Many sequence mining algorithms have been proposed to discover knowledge from that kind of data. Nevertheless, to be able to deal with relationships between objects that occur in the sequences, we have to shift to first order logic. Some prototypes have been designed in order to discover first order sequential patterns such as [31, 30] however, in order to use such techniques for user modeling we think some work remains to be done in order to better integrate background knowledge and constraint learning.

5 Topics to be addressed

In [21] some directions were given for the future of MRDM in general. Moreover, we would like to complete some of them and give some new ones in the context of user modeling in the Semantic Web.

Background Knowledge was one of the main advantages of first general ILP systems that allowed the design of powerful applications (for example in drug discovery). Concerning more specialized MRDM tools, background knowledge is not always taken into account and we think this is an important drawback to be addressed. Indeed, we think that realistic models of users need to integrate knowledge about the context of these one. Nevertheless, such knowledge may not always be expressed in a pure first order form. For example, as we said previously, we may need sometimes to work on numerical constraints. We think this topic is essential to be able to deal with user preferences in the SW in the future.

Mining the SW to discover user models should not be done with classical techniques while scanning the data if we want to take into account the richness of information available. In that sense, integrating tools such as Xcerpt [15] could be useful in order

to query the SW using some techniques similar to XPATH but extended to first order logic.

The (Semantic) Web is a huge evolving area and MRDM techniques should take this into account. On the SW side, some techniques are being proposed in order to deal with changes that can appear on the Web, we may for example cite the XChange language [6, 16] that can be use to program reactive behaviors on the Web. Models learnt should be able to evolve dynamically while the SW is changing, thus such techniques may me very useful to provide this.

Privacy Preserving data mining [3] is an important field of research nowadays. In the context of SW, and moreover for user modeling, it seems obvious that new MRDM systems will have to deal with this feature.

Of course scalability and efficiency of the techniques have to be preserved and [12] has shown that it is a great challenge.

6 Conclusion

In this paper we have shown that various researches are currently being done in order to design the SW in a framework based on first order logic. Specifying preferences, policies, Web services, ontologies, etc, will be done in the near future using rule-based languages. If we want to use machine learning in the context of user modeling in the SW, it seems obvious that the new methods, algorithms or systems that we will designed will have to deal with such languages. ILP and then MRDM have provided some first steps in that directions, nevertheless many work remains to be done. Efficiently dealing with background knowledge, evolving data, privacy, seems to be some of the most important topics that have to be addressed in the near future to be able to learn user models in the Semantic Web. We think that a cross fertilization could be done between the two European Networks of Excellence REWERSE and PASCAL in order to design new tools for user modeling in the Semantic Web.

References

1. G. Adomavicius and A. Tuzhilin. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 2005. to appear.
2. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic Databases. In *Proc. of the 28th VLDB Conference*, pages 143–154, 2002.
3. R. Agrawal and R. Srikant. Privacy-Preserving Data Mining. In *Proc. of the 2000 ACM SIGMOD Conference*, pages 439–450, 2000.
4. C. R. Anderson, P. Domingos, and D. S. Weld. Relational Markov models and their application to adaptive web navigation. In *Proc. of the 8th ACM SIGKDD Conference*, pages 143–152. ACM, 2002.
5. D. Beckett. RDF/XML Syntax Specification (Revised). *http://www.w3.org/TR/rdf-syntax-grammar/*, 2004.
6. S. Berger, F. Bry, B. Lorenz, H. J. Ohlbach, P.-L. Pătrânjan, S. Schaffert, U. Schwertel, and S. Spranger. Reasoning on the Web: Language Prototypes and Perspectives. In *Proc. of European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, pages 157–164, 2004.

7. M. Bernard and F. Jacquenet. Discovering Rules to Design Newspapers: An Inductive Constraint Logic Programming Approach. *Applied Artificial Intelligence*, 12(6):547–567, 1998.
8. T. Berners-Lee. Rules and Facts: Inference engines vs Web. *http://www.w3.org/DesignIssues/Rules.html*, 1998.
9. T. Berners-Lee. Semantic Web Road Map. *http://www.w3.org/DesignIssues/ Semantic.html*, 1998.
10. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 2001.
11. H. Blockeel and L. De Raedt. Top-Down Induction of First-Order Logical Decision Trees. *Artificial Intelligence*, 101(1-2):285–297, 1998.
12. H. Blockeel and M. Sebag. Scalability and Efficiency in Multi-Relational Data Mining. *SIGKDD Explorations*, 5(1):17–30, 2003.
13. H. Boley, S. Tabet, and G. Wagner. Design Rationale for RuleML: A Markup Language for Semantic Web Rules. In *Proc. of the 1st Semantic Web Working Symposium, Stanford University*, pages 381–401, 2001.
14. D. Brickley and R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. *http://www.w3.org/TR/rdf-schema/*, 2004.
15. F. Bry, T. Furche, L. Badea, C. Koch, S. Schaffert, and S. Berger. Querying the Web Reconsidered: Design Principles for Versatile Web Query Languages. *Journal of Semantic Web and Information Systems (IJSWIS)*, 1(2), 2005.
16. F. Bry and P.-L. Pătrânjan. Reactivity on the Web: Paradigms and Applications of the Language XChange. In *Proc. of the ACM SAC Symposium*. ACM, 2005.
17. R. Chinnici, M. Gudgin, J.J. Moreau, J. Schlimmer, and S. Weerawarana. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. *http://www.w3.org/TR/wsdl20*, 2004.
18. L. Dehaspe and H. Toivonen. Discovery of Frequent DATALOG Patterns. *Data Mining and Knowledge Discovery*, 3(1):7–36, 1999.
19. Peter Dolog, Nicola Henze, Wolfgang Nejdl, and Michael Sintek. Towards the Adaptive Semantic Web. In *Proc. of the International Workshop on Principles and Practice of Semantic Web Reasoning*, LNCS 2109, pages 51–68. Springer, 2003.
20. Peter Dolog, Nicola Henze, Wolfgang Nejdl, and Michael Sintek. The Personal Reader: Personalizing and Enriching Learning Resources Using Semantic Web Technologies. In *Proc. of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, LNCS 3137, pages 85–94, 2004.
21. P. Domingos. Prospects and Challenges for Multi-Relational Data Mining. *SIGKDD Explorations*, 5(1):80–83, 2003.
22. S. Dzeroski. Multi-Relational Data Mining: an Introduction. *SIGKDD Explorations*, 5(1):1–16, 2003.
23. D. C. Fallside and P. Walmsley. XML Schema Part 0: Primer Second Edition. *http://www.w3.org/TR/xmlschema-0/*, 2004.
24. F. Giannotti, G. Manco, and J. Wijsen. Logical Languages for Data Mining. In *Logics for Emerging Applications of Databases*, pages 325–361. Springer, 2003.
25. J. Hendler. DAML+OIL (March 2001). *http://www.daml.org/2001/03/daml+oil-index.html*, 2001.
26. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *http://www.daml.org/rules/proposal/*, 2003.
27. F. Jacquenet and P. Brenot. Learning User Preferences on the Web. In *Proc. PAKDD-98,*, LNCS 1394, pages 385–387, April 1998.
28. U. Junker. Preferences in AI and CP: Symbolic Approaches. Papers from the AAAI Workshop. Technical Report WS-02-13, AAAI Press, 2002.

29. M. Kifer, R. Lara, A. Polleres, C. Zhao, U. Keller, H. Lausen, and D. Fensel. A Logical Framework for Web Service Discovery. In *Proc. of the ISWC'2004 Workshop on Semantic Web Services*, 2004. CEUR-WS, vol. 119.
30. S. D. Lee and L. De Raedt. Constraint Based Mining of First Order Sequences in SeqLog. In *Database Support for Data Mining Applications*, LNCS 2682, pages 154–173, 2004.
31. C. Masson and F. Jacquenet. Mining Frequent Logical Sequences with SPIRIT-L^{O}G. In *Proc. of the ILP Conference*, LNCS 2583, pages 166–181, 2002.
32. D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. *http://www.w3.org/TR/owl-features/*, 2004.
33. N. Mitra. SOAP Version 1.2 Part 0: Primer. *http://www.w3.org/TR/2003/REC-soap12-part0-20030624/*, 2003.
34. A. Mounier, O. Boissier, and F. Jacquenet. Conversation Mining in Multi-agent Systems. In *Proc. 3rd CEEMAS Conference*, LNCS 2691, pages 158–167, 2003.
35. S. Muggleton and L. De Raedt. Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming*, 19/20:629–679, 1994.
36. Rewerse Network of Excellence. Rule-based policy specification: State of the art and future work. *REWERSE deliverable I2-D1*, 2004.
37. K. O'Hara, H. Alani, Y. Kalfoglou, and N. Shadbolt. Trust Strategies for the Semantic Web. In *Proc. of the ISWC'04 Workshop on Trust, Security, and Reputation on the Semantic Web*, 2004. CEUR-WS, vol. 127.
38. T. Schaub and K. Wang. A semantic framework for preference handling in answer set programming. *Theory and Practice of Logic Prog.*, 3(4-5):569–607, 2003.
39. Gerd Wagner. How to Design a General Rule Markup Language? In *Proc. of the Workshop on XML Technologies for the Semantic Web*, pages 19–37, 2002.
40. D. Weitzner, J. Hendler, T. Berners-Lee, and D. Connolly. *Web and Information Security*, chapter Creating a Policy-Aware Web: Discretionary, Rule-based Access for the World Wide Web. IOPress, 2005. To appear.
41. D. S. Weld, C. R. Anderson, P. Domingos, O. Etzioni, K. Gajos, T. A. Lau, and S. A. Wolfman. Automatically Personalizing User Interfaces. In *Proc. of the 18th IJCAI*, pages 1613–1619, 2003.

User models from implicit feedback for proactive information retrieval

Samuel Kaski[1], Petri Myllymäki[2], and Ilpo Kojo[3]

[1] University of Helsinki, Department of Computer Science, and Helsinki University of Technology, Neural Networks Research Centre
[2] Helsinki Institute for Information Technology, University of Helsinki and Helsinki University of Technology
[3] Helsinki School of Economics, Center for Knowledge and Innovation Research
http://www.cis.hut.fi/projects/mi/prima

Abstract. Our research consortium develops user modeling methods for proactive applications. In this project we use machine learning methods for predicting users' preferences from implicit relevance feedback. Our prototype application is information retrieval, where the feedback signal is measured from eye movements or user's behavior. Relevance of a read text is extracted from the feedback signal with models learned from a collected data set. Since it is hard to define relevance in general, we have constructed an experimental setting where relevance is known a priori.

1 Introduction

Successful proactivity, i.e. anticipation, in varying contexts requires generalization from past experience. Generalization, on its part, requires suitable powerful (stochastic) models and collection of data about relevant past history to learn the models.

Information retrieval is an example of a task which would benefit from proactivity: the user has a goal—to find relevant pieces of information—which is not directly observable to the system and needs to be inferred from the actions. Information retrieval applications appear in various contexts from traditional searching of text or multimedia documents to computerized manuals and help systems. Analogously, a user entering a room, for instance a shop, in which multiple actions are possible, could be assisted in choosing from a set of "relevant" actions. Some experimental information retrieval systems take the user's goals into account, but they rely heavily on explicit input. To the extent that implicit information is used, it has been restricted to heuristic estimation of relevance from navigation data.

The overall interest profiles of user populations have been modeled in collaborative filtering systems recommending potentially interesting items for groups of similar users. Information about the current interest of a user, on the other hand, has been mostly acquired as explicit relevance feedback, by displaying the user a set of retrieved items and asking which are relevant. The problem with both approaches is that interactive collection of relevance information is laborious, and it becomes outdated quickly. Moreover, only a small amount of information per item can be collected. Hence, methods that would infer multi-faceted relevance from the natural actions of users would be extremely valuable.

Inferring the goals of users requires measuring signals that (i) contain a sufficient amount of information about the intentions of the user, (ii) have a sufficiently high "relevant-signal-to-noise ratio" that at least some aspects of the intentions can be extracted and modeled, and (iii) can be measured feasibly also in practice. We use *eye movement* signals that exhibit both voluntary and involuntary signs of the interests and intentions.

2 Status

We have developed the first versions of machine learning methods for inferring relevance from eye movements [1] and studied coupling of text content to the task of estimating relevance [2]. We are currently working on an information retrieval application, on incorporating other forms of implicit relevance feedback, and on studying implications of the work in perceptual psychology.

3 Eye movement challenge

We have started a competition, with deadline in September 2005, on predicting relevance of sentences from eye movement data. The data has been collected in a controlled experimental setting in which the relevance is known, which makes machine learning methods directly applicable.

The challenge is at `http://www.cis.hut.fi/eyechallenge2005/`; participation is open to all.

Acknowledgements

The project is funded by the Academy of Finland, decision number 122209, and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. The authors would like to thank all other people in the PRIMA project and acknowledge that access rights to the data sets and other materials produced in the PRIMA project are restricted due to other commitments.

References

1. Salojärvi, J., Puolamäki, K., Kaski, S.: Relevance feedback from eye movements for proactive information retrieval. In Heikkilä, J., Pietikäinen, M., Silvén, O., eds.: workshop on Processing Sensory Information for Proactive Systems (PSIPS 2004), Oulu, Finland (2004)
2. Savia, E., Kaski, S., Tuulos, V., Myllymäki, P.: On text-based estimation of document relevance. In: Proc. IJCNN'04. (2004) 3275–3280

Context changes detection by one-class SVMs ⋆

Gaëlle Loosli[1], Sang-Goog Lee[2], and Stéphane Canu[1]

[1] PSI, CNRS FRE2645, INSA de Rouen, FRANCE
[2] Interaction Lab./Context Awareness TG,
Samsung Advanced Institute of Technology, Korea

Abstract. We are interested in a low level part of user modeling, which is the change detection in the context of the user. We present a method to detect on line changes in the context of a user equipped with non invasive sensors. Our point is to provide, in real time, series of unlabeled contexts that can be classified and analyzed on a higher level.

Introduction

For a system that aims at taking into account the user, we need to consider that there are many different behaviors as well as many different users. Hence we need adaptative, unsupervised (or semi-supervised) learning methods. Our idea is to take advantage of wearable computers and wearable sensors (indeed their use is realistic at least for certain categories of people, such as pilots) to retrieve the current context of the user. Wearable sensors can be physiological (EMG, ECG, blood volume pressure...) or physical (accelerometers, microphone...). Contexts are depending on the application using the system and can be behaviors, affective states, combinations of these. Since this problem of context retrieval is very complex, we choose to detect changes at first place instead of labeling directly. Indeed this way we can apply unsupervised and fast methods which saves time for labeling (the labeling task is then applied only when changes are detected). Our interest lies in low level treatments and we present a non parametric change detection algorithm. This algorithm is meant to provide sequences of unlabeled contexts to be analyzed to higher level applications. Detection is made from signals given by non invasive sensors the user is wearing. Note that the methods presented here could as well be adapted to external sensors.

1 Change detection as Novelty detection

As a starting hypothesis we are assuming that the wearer activity and affective state can be represented by a sequence of states. For a given state, the observed time series are the realization of a stationary (or weakly non stationary) process. We assume also that the sequence of states changes slowly in comparison with the measured data. Two

⋆ This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

times series at different time scales have to be considered: the context time series (unobserved) and the measurements. The context C is a discrete random variable while the data is a real multidimensional one. The problem of retrieving C given the measurements is a problem of signal segmentation. Because no prior knowledge is made about the nature of the underlying probability distributions, we are looking for a non parametric signal segmentation technique, *i.e.* a method performing an automatic decomposition of a given multidimensional time series into stationary (or weakly non stationary) segments. A way to perform such a segmentation is to use change detection framework from signal processing [1] together with a kernel learning technique [2] to deal with the non parametric requirement. To perform this segmentation on-line, our detection system has to rely on local characteristics (in time) of the available signals. The method described below implements this hypothesis test using one class support vector machine to model the unknown density. We will use the term *rupture* to refer to a change of state. This covers a sudden high misclassification rate when we consider the classification methods as well as a change of distribution from one point to the next one when we consider the segmentation task.

Presentation of the stationary framework for novelty detection is followed by a quick description of one class SVM. Then we give the detection algorithm and some results on real data.

1.1 Stationary framework: Novelty detection as an approximation of an optimal test

Let $X_i, i = 1, 2t$ be a sequence of random variables distributed according to some distribution $\mathbb{P}_i$. We are interested in finding whether or not a change has occurred at time t. To begin with a simple framework we will assume the sequence to be stationary from 1 to t and from $t+1$ to $2t$, *i.e.* there exists some distributions $\mathbb{P}_0$ and $\mathbb{P}_1$ such that $P_i = P_0, i \in [1, t]$ and $P_i = P_1, i \in [t+1, 2t]$. The question we are addressing can be seen as determining if $\mathbb{P}_0 = \mathbb{P}_1$ (no change has occurred) or else $\mathbb{P}_0 \neq \mathbb{P}_1$ (some change have occurred). This can be restated as the following statistical test:

$$\begin{cases} \mathcal{H}_0 : \ \mathbb{P}_0 = \mathbb{P}_1 \\ \mathcal{H}_1 : \ \mathbb{P}_0 \neq \mathbb{P}_1 \end{cases}$$

In this case the likelihood ratio is the following:

$$\Lambda_l(x_1, ..., x_{2t}) = \frac{\prod_{i=1}^{t} \mathbb{P}_0(x_i) \prod_{i=t+1}^{2t} \mathbb{P}_1(x_i)}{\prod_{i=1}^{2t} \mathbb{P}_0(x_i)} = \prod_{i=t+1}^{2t} \frac{\mathbb{P}_1(x_i)}{\mathbb{P}_0(x_i)}$$

since both densities are unknown the generalized likelihood ratio (GLR) has to be used:

$$\Lambda(x_1, ..., x_{2t}) = \prod_{i=t+1}^{2t} \frac{\widehat{\mathbb{P}}_1(x_i)}{\widehat{\mathbb{P}}_0(x_i)}$$

where $\widehat{\mathbb{P}}_0$ and $\widehat{\mathbb{P}}_1$ are the maximum likelihood estimates of the densities.

Because we want our detection method to be universal, we want it to work for any possible density. Thus some approximations have to be done to clarify our framework. First we will assume both densities $\mathbb{P}_0$ and $\mathbb{P}_1$ belong to the generalized exponential family thus there exists a reproducing kernel Hilbert space $\mathcal{H}$ embedded with the dot product $\langle ., . \rangle_{\mathcal{H}}$ and with a reproducing kernel k such that [10]:

$$\mathbb{P}_0(x) = \exp\langle \theta_0(.), k(x, .)\rangle_{\mathcal{H}} - g(\theta_0) \quad \text{and} \quad \mathbb{P}_1(x) = \exp\langle \theta_1(.), k(x, .)\rangle_{\mathcal{H}} - g(\theta_1)$$

where $g(\theta)$ is the so called log-partition function. Second hypothesis, the functional parameter θ_0 and θ_1 of these densities will be estimated on the data of respectively first and second half of the sample by using the one class SVM algorithm. By doing so we are following our initial assumption that before time t we know the distribution is constant and equal to some $\mathbb{P}_0$. The one class SVM algorithm provides us with a good estimator of this density. The situation of $\widehat{\mathbb{P}}_1(x)$ is more simple. It is clearly a robust approximation of the maximum likelihood estimator. Using one class SVM algorithm and the exponential family model (see annexe 1) both estimate can be written as:

$$\widehat{\mathbb{P}}_0(x) = \exp\left(\sum_{i=1}^{t} \alpha_i^{(0)} k(x, x_i) - g(\theta_0)\right), \widehat{\mathbb{P}}_1(x) = \exp\left(\sum_{i=t+1}^{2t} \alpha_i^{(1)} k(x, x_i) - g(\theta_1)\right)$$

where $\alpha_i^{(0)}$ is determined by solving the one class SVM problem on the first half of the data (x_1 to x_t). while $\alpha_i^{(1)}$ is given by solving the one class SVM problem on the second half of the data (x_{t+1} to x_{2t}). Using these three hypothesis, the generalized likelihood ratio test is approximated as follows:

$$\begin{aligned} \Lambda(x_1, ..., x_{2t}) > s \Leftrightarrow & \prod_{j=t+1}^{2t} \frac{\exp \sum_{i=t+1}^{2t} \alpha_i^{(1)} k(x_j, x_i) - g(\theta_1)}{\exp \sum_{i=1}^{t} \alpha_i^{(0)} k(x_j, x_i) - g(\theta_0)} > s \\ \Leftrightarrow & \sum_{j=t+1}^{2t} \left(\sum_{i=1}^{t} \alpha_i^{(0)} k(x_j, x_i) - \sum_{i=t+1}^{2t} \alpha_i^{(1)} k(x_j, x_i) \right) < s' \end{aligned}$$

where s' is a threshold to be fixed to have a given risk of the first kind a such that:

$$\mathbb{P}_0 \left(\sum_{j=t+1}^{2t} \left(\sum_{i=1}^{t} \alpha_i^{(0)} k(x_j, x_i) - \sum_{i=t+1}^{2t} \alpha_i^{(1)} k(x_j, x_i) \right) < s' \right) = a$$

It turns out that the variation of $\sum_{i=t+1}^{2t} \alpha_i^{(1)} k(x_j, x_i)$ are very small in comparison to the one of $\sum_{i=1}^{t} \alpha_i^{(0)} k(x_j, x_i)$. Thus $\widehat{\mathbb{P}}_1(x)$ can be assumed to be constant, simplifying computations. In this case the test can be performed only considering:

$$\sum_{j=t+1}^{2t} \left(\sum_{i=1}^{t} \alpha_i^{(0)} k(x_j, x_i) \right) < s$$

This is exactly the novelty detection algorithm as proposed in [11]. Thus we show how to derive it as a statistical test approximating a generalized likelihood ratio test, optimal under some condition in the Neyman Pearson framework.

1.2 One class support vector machines: 1-class SVM

The 1-class SVM [2] is a method that aims at learning a single class, by determining it's contours. To explain 1-class SVM, we can begin by giving a kernel. A kernel $k(x, y)$ is a positive and symmetric function of two variables (for more details see [12]) lying in a Reproducing Kernel Hilbert Space with the scalar product: $\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{k} \sum_{j=1}^{l} f_i g_j k(\mathbf{x}_i, \mathbf{x}'_j)$.
In this framework, the 1-class SVM problem with the sample $(\mathbf{x}_i), i = 1, m$ is the solution of the following optimisation problem under constraints for $f \in \mathcal{H}$:

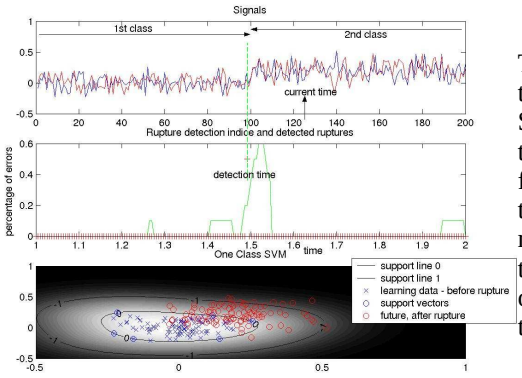
$$\begin{cases} \min\limits_{f,\rho,\boldsymbol{\xi}} \dfrac{1}{2}\|f\|^2_{\mathcal{H}} + C\sum\limits_{i=1}^{m} \xi_i - \rho & \\ \text{s.t.} \quad f(\mathbf{x}_i) > \rho - \xi_i & i = 1, m \\ \text{and} \quad \xi_i \geq 0, & i = 1, m \end{cases} \tag{1}$$

where C is a scalar that adjusts the smoothness of the decision function, ρ is a scalar called bias and ξ_i are slack variables.
The dual formulation is:

$$\begin{cases} \max\limits_{\boldsymbol{\alpha} \in \mathbf{R}^m} -\frac{1}{2}\boldsymbol{\alpha}^\top K \boldsymbol{\alpha} & \\ \text{s.t.} \quad \boldsymbol{\alpha}^\top e = 1 & \\ \text{and} \quad 0 \leq \alpha_i \leq C & i = 1, m \end{cases} \tag{2}$$

where K is the kernel matrix $K_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$ and $\mathbf{e} = [1, \ldots, 1]^\top$. The 1-class SVM solution is then given by solving a quadratic optimization problem of dimension m under box constraints. The decision function is $D(x) = sign(f(x) - \rho)$. The input points are considered as part of the current class as long as the decision function is positive.

1.3 Rupture detection algorithm



This figure shows the relation between rupture detection and 1class SVM. On the first part we show the signals, drawn from two different but close gaussian distributions. The second graph shows the ruptures found by our algorithm on those data and the last part is the output of a 1class SVM trained on the *past* data.

Fig. 1. Rupture Detection and OneClass SVM

Based on the idea of novelty detection using 1-class SVM, our method aims at learning the current state and test how well it can recognise the next points. The delay expected in the detection is the length of the signal used to test the current model. In other words, it depends on the number of *future points* we are considering to compute the rupture detection indices (i.e. the misclassification rate). To relate this method to the statistical test, let's recall that the classification rate is given the proportion of positive values obtained when computing $\sum_{i=1}^{t} \alpha_i^{(0)} k(x_j, x_i)$ which is the quantity we are interested in (*i.e.* we want the probability that $\sum_{i=1}^{t} \alpha_i^{(0)} k(x_j, x_i) < s$ to be small to decide there is no rupture, which is equivalent to decide a rupture when the proportion of misclassification exceed a given threshold).

2 Experiments and Results

Video games. In games, states are defined by objective facts (win, lose, particular events...). The chosen game is XBlast, a game that can be played over the network. The main goal is to kill adversaries with bombs. One player is equipped with the sensors and filmed.

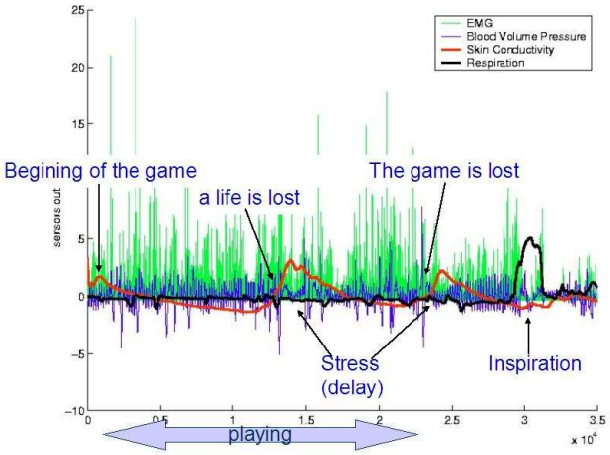


Fig. 2. Example of collected signals. Here are represented the normalized output of the 4 used sensors (EMG, Blood Volume Pressure, Skin Conductivity and Respiration.)

Part of the datasets are labeled according to the video (player looses a life, player kills another player, the game begins, the player wins, dangerous events). In this case the labels are not indicating the state of the user but mark where a rupture should be found. However those labels are not completely reliable. First they are not precise (1 second error is quickly done when manually detecting events and this corresponds to several hundreds of input points) and second they are not exhaustive. We did this labeling to have some indicators for the evaluation of our method. However we did expect to obtain some differences between manual and automatic labels.

Results Biological data are sampled at 256Hz. We work on a sliding window of 64 points (25ms) and take into account the following features: minimum, maximum, mean,

variance, Fourier transform on 32 frequencies. We thus work on points in dimension 36. Taking into account the relatively high frequency compared to our application, we consider every eighth point. We apply the 1-class SVM to these feature points (with a gaussian kernel of bandwidth 0.00008 - value given by cross validation). This algorithm is learned on the previous seconds and we test its performance on the next second.

In table 1, we sum up the results of this experiment (ruptures detected with a threshold of ten percent of misclassified points to decide a rupture). The two main conclusions we draw from it are first, that manual labeling does not seem to be enough and second that automatic rupture detection is efficient enough (good recognition rate, low false detection rate). We see that automatic rupture detection actually detects ruptures that we can't see on a video.

The problem we can point out is the non detection problem. Indeed several events are not detected with our method. Events like *killing an other player* do not appear in biological signals. However, it turns out that those events do not really affect the user's state when we can't notice them in the data. Let's illustrate this point : killing a player who has no chance to win is not an important event during the game while taking the last life of the last adversary in game is relevant. This last case will be detected as it will be combined to the emotion of winning the game. We observed that similar events are not meaning the same depending on when they appear. We also observed that meaningful events do appear in the data. Our conclusion from those observations is that it will be hard to really have an objective criteria on the efficiency of the rupture detection.

	Game		Natural	
Number of meaningful ruptures (a posteriori)	31		24	
Manual Ruptures	26/31	83.9%	17/24	70.1%
Automatic Rupture	21/31	67.7%	22/24	91.7%
In both methods	16/31	51.6 %	15/24	62.5 %
Only in manual	10/31	32.3%	2/24	8.3%
Only in automatic	5/31	16.1%	7/24	29.1%
False automatic detection	6/33	18.8%	12/36	33.3%
Non detected (automatic)	10/31	32.2%	2/24	8.3%

Table 1. This table shows a comparison between manual and automatic rupture for both experiments. The number of meaningful ruptures is determined a posteriori.

Natural behavior In this experiment we try to retrieve information from sensors when the user is doing nothing particularly (with no target application). The idea is then to equip the user with various sensors and let him freely move around with no instruction. This experiment is done with biological and motion sensors and filmed in order to facilitate the interpretation of the signals and labeling.

Results We work on a sliding window of 50 points (50ms) and take into account the following features: minimum, maximum, mean, variance, Fourier transform on 32 frequencies. The gaussian kernel's bandwidth is 0.0003.

In table 1, we sum up the results of this experiment. As in previous experiment with video games, we observe that the automatic rupture detection from the signals enable

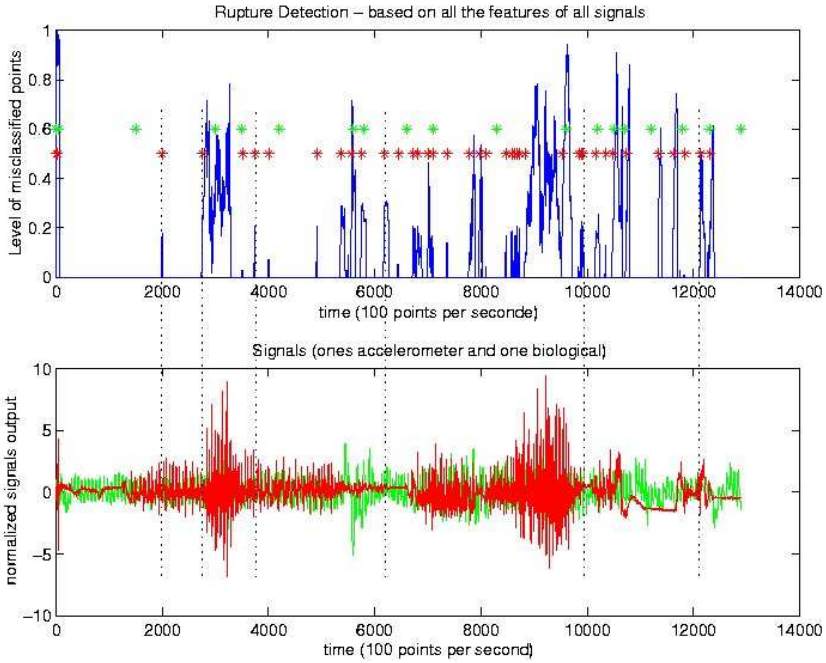


Fig. 3. Here are the ruptures detected with a threshold of ten percent of misclassified points to decide a rupture on the dataset. Red crosses shows the automatically found ruptures and green crosses are the manual labels. For the sake of visibility, the second part of the figure shows one biological signal and one accelerometer signal although the method is applied to all the sensors.

us to find some ruptures that are not observable on the video record. For instance the change of breathing that occurs when running or going up the stairs is pretty obvious in the data whereas we can't see it.

Compared to video games, the automatic rupture detection seems to be more noisy. We face here the problem of false detection. We need here to carry out some more experiments to check whether this problem might be solved by a better control on the different parameters of the method (mainly the kernel bandwidth and the sliding window on the data).

This experiment, coupled to the previous one, let us think that we can rely on such a rupture detection method to separate different states. The next step will be to be able to add some semantic on these sections of signals or to the ruptures themselves. The important point is to keep in mind that we can't apply supervised learning in these problems so our goal is to design an almost unsupervised system.

Perspectives

We presented here a method to separate different contexts on line. This method is fast and robust, can be applied in a unsupervised framework to any person in many situations. It can be improved, using more powerful signal processing methods on raw data. Nevertheless we provide an efficient non parametric rupture detection algorithm base on kernel methods and statistical tests. We are know working on improvement for detection delay based on CUSUM approaches. We expect the sequence of contexts

extracted this way to be convenient for semi-supervised classification applications and more generally context-driven applications.

References

1. Basseville, M., Nikiforov, I.V.: Detection of Abrupt Changes - Theory and Application. Prentice-Hall (1993)
2. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press (2002)
3. Gustafsson, F.: Adaptive filtering and change detection. John Wiley & Sons, Ltd (2001)
4. Markou, M., Singh, S.: Novelty detection: a review, part 2: neural network based approaches. Signal Process. **83** (2003) 2499–2521
5. Fancourt, C., Principe, J.C.: On the use of neural networks in the generalized likelihood ratio test for detecting abrupt changes in signals. In: Intl. Joint Conf. on Neural Networks, pp. 243-248, at Como, Italy. (2000)
6. Roberts, S., Roussos, E., Choudrey, R.: Hierarchy, priors and wavelets: structure and signal modelling using ica. Signal Process. **84** (2004) 283–297
7. Qi, Y., Minka, T.: Expectation propagation for signal detection in flat-fading channels. In: Proceedings of the IEEE International Symposium on Information Theory. (2003)
8. Nguyen, X., Wainwright, M.J., Jordan, M.I.: Nonparametric decentralized detection using kernel methods. IEEE Transactions on Signal Processing (accepted for publication) (2005)
9. Lenser, S., Veloso, M.: Non-parametric time series classification. In: Under review for ICRA'05. (2005)
10. Smola, A.: Exponential families and kernels. Berder summer school (2004) `http://users.rsise.anu.edu.au/ smola/teaching/summer2004/`.
11. Scholkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J.: Support vector method for novelty detection. In Solla, S., Leen, T., Muller, K., eds.: NIPS, MIT Press (2000) 582–588
12. Atteia, M., Gaches, J.: Approxiation Hilbertienne. Presses Universitaires de Grenoble (1999)
13. Scherer, K.R.: Ways to study the nature and frequency of our daily emotions: Reply to the commentaries on "emotions in everyday life". Social Science Information **43** (2004) 667–689

Improving Infoville XXI using Machine Learning techniques*

J.D. Martín-Guerrero[1], P.J.G. Lisboa[2], E. Soria-Olivas[1], A. Palomares[3], E. Balaguer-Ballester[3], A.J. Serrano-López[1], and G. Camps-Valls[1]

[1] Electronics Engineering Department, University of Valencia, Spain.
{jdmg,soriae,ajserran,gcamps}@uv.es
[2] School of Computing and Mathematical Sciences, Liverpool John Moores University, UK.
p.j.lisboa@livjm.ac.uk
[3] Research & Development Department, Tissat, Inc., Spain.
{apalomares,ebalaguer}@tissat.es

Abstract. Infoville XXI is a citizen web portal of Valencia (Spain). This paper presents several approaches based on Machine Learning that help in improving the site. Three ways of improvement are taken into account: (i) user clickstream forecasting, (ii) user profiling by clustering and (iii) recommendation of services to users, being the last two techniques part of a methodological framework with general applicability that tries to be useful for a range of web sites as wide as possible. Results obtained with data sets from this web portal show that the most appropriate techniques for user clickstream forecasting become Support Vector Machines and Multilayer Perceptrons, whilst Adaptive Resonance Theory and Self-Organizing Maps appear to be the most suitable techniques for clustering. Final recommendation and adaptation of the recommender system is currently being developed by using Learning Vector Quantization and Reinforcement Learning.

1 Introduction

Citizen web portals have become an interactive gateway between citizens and administration. The success and acceptance of these portals depend largely on their capability to be attractive for most of the citizens as well as the public and private entities in the area. An easy way to make the site attractive for the majority of the people is to know the characteristics of users, hence being able to carry out a customization of the site.

We focus on Infoville XXI (I-XXI), http://www.infoville.es/, a region web portal for citizens of Valencia, in Spain, which is an official web site supported by Regional Government. The portal provides citizens with more than 2,000 services grouped into 21 descriptors. These descriptors gather similar services under a unified label. Some descriptors include services related to administrative tasks whereas other descriptors are more focused on leisure services. We propose three strategies in order to improve

* This work has been partially supported by the research projects CTIAUT2/A/03/112 and FIT-350100-2004-427. The authors want to express their thanks to *Fundació OVSI (Oficina Valenciana per a la Societat de la Informació—Valencian Office for Information Society)* for providing the data used in this work.

I-XXI by means of better knowing its users: 1) prediction of the next links that will be clicked on by users; 2) profiling of web users using clustering; 3) recommendation of attractive items to users.

The first strategy is related to the development and evaluation of Machine Learning (ML) methods to predict users' web browsing in I-XXI, as it will be described in Section 2. Our approach is focused on next session prediction of accessing a certain descriptor. The used models are the following: Associative Memories (AMs), Multilayer Perceptrons (MLPs), Classification and Regression Trees (CARTs), and Support Vector Machines (SVMs).

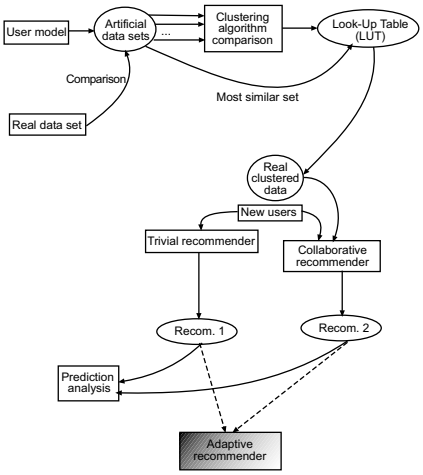


Fig. 1. General methodology.

The two other strategies (user profiling and recommendation) are part of a methodology that allows to provide useful recommendations to users of a generic citizen web portal from the very beginning (See Fig. 1). The methodology is based on the generation of simulated web accesses (by a user model) in order to produce artificial data sets. These artificial data sets are then used to benchmark different clustering algorithms thus deciding the best algorithm for each kind of data set, i.e., the best algorithm for each kind of web portal.

Section 3 deals with clustering of web users. Real data sets are clustered using the algorithm that turns out to be the best one with the most similar artificial data set. The similarity between artificial and real data sets is measured by analyzing the characteristics of both data sets, basically in terms of the number of descriptors and the number of users. This methodology has been tested with data from I-XXI, and it appears to be quite accurate and useful; in fact, it has been included in the software iSUM®[4]. The clustering algorithms used to profile web users are the following: C-Means (CM), Fuzzy C-Means (FCM), Expectation-Maximization (E-M), Hierarchical Clustering Algorithms (HCA), Kohonen's Self-Organizing Maps (SOM) and the Adaptive Resonance Theory (ART).

Finally, a recommender system is developed using the information extracted from clustering, as it is detailed in Section 4. The recommendation procedure is based on collaborative filtering. At present, we are working in making the recommender system adaptive. Methods based on Learning Vector Quantization (LVQ) and Reinforcement Learning (RL) appear to be promising tools.

[4] iSUM is a software product developed by the company *Tissat, Inc.* iSUM is the tool used to develop I-XXI; iSUM is developed using *XML* technology on a *Java* platform.

2 Prediction of accessing descriptors

2.1 Data

We used access records in I-XXI from June 2002 to February 2003. A preprocessing stage removed those users who presented an atypical behavior. Basically, they presented a great number of accesses in a short time, as they were fictitious users generated by the administrators of the portal for test purposes. Users with just one session were also removed since it was not possible to validate their predictions. Finally, the processed data set was formed by 5,129 users who logged between 4 and 413 different sessions, accessing to 447 different services. We used a selection of the 18 most representative descriptors.

2.2 Methods

Four different models were used for this prediction task: AM, CART, MLP and SVM. The function of an AM is to recognize previously learned vectors. AMs are implemented with a single layer of computing units and *Hebbian learning* is used [1].

CART is a binary decision tree algorithm [1], which has two branches in each internal node. Methodology is characterized by a pruning strategy, a binary-split search approach, automatic self-validation procedures, and a splitting criterion.

The MLP is the most widely used neural network. It is composed of a layered arrangement of artificial neurons in which each neuron of a given layer feeds all the neurons of the next layer [1]. The model is usually trained with the backpropagation (BP) learning algorithm. However, we used the Expanded Range Approximation (ERA) algorithm in order to alleviate the problem of falling into local minima [2].

An SVM is an approximate implementation of the method of structural risk minimization [1]. It is trained to construct a hyperplane for which the margin of separation is maximized. SVMs can handle large input spaces efficiently, are robust to noisy samples, and can automatically identify a small subset made up of informative training samples, namely *support vectors* (SVs).

2.3 Prediction Setup

We developed models by using data from the current session s and the two previous sessions ($s-2$ and $s-1$) to carry out a one step ahead prediction (session $s+1$). We did not take into account the clickstream sequence but only the frequency of access to each category. This was because people in charge of I-XXI is interested in knowing the different services accessed by users rather than knowing the exact sequence of accesses within a certain session. Nevertheless, further work will include clickstream prediction, as well. Data were transformed into patterns in order to be used for training and validating the models. Every pattern was formed by a matrix whose size was 18×4. Each column corresponded to sessions $s-2$, $s-1$, s and $s+1$, respectively. Each row stood for the frequency of accesses to the different descriptors in the corresponding session. The total number of patterns was 11,617. The data were split into three sets: a training set formed by 1,743 patterns; a validation set formed by other 1,743 patterns; and a test set formed by 8,131 patterns.

2.4 Results

In order to carry out a model comparison, we transformed the frequency-based prediction in a measure of Success Rate (SR) by considering a correct prediction when the Absolute Average Error was lower than 0.1 (values were normalized between -1 and $+1$). This SR was an intuitive and straightforward way to carry out a preliminary comparison of different algorithms. This criterion might be overoptimistic in terms of the percentages achieved because of the high number of non-accessed categories which were easily learned by the models. High SRs in the test set were obtained for all the models, the highest values corresponding to SVMs.

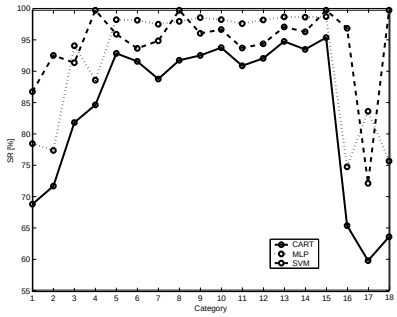


Fig. 2. Success Rates (SRs) [%] obtained by CART (solid line), MLP (dotted) and SVM (dashed).

The SRs using AMs were near 92%. In Fig. 2, CART, MLP, and SVM are benchmarked. SRs are slightly higher for SVM than for MLP, which are, in turn, much better than those obtained by CART. There are some categories whose prediction become considerably more difficult for all the algorithms. The latter is particularly dramatic in the case of CART, since they show low SRs (around 60%) for some categories. MLP and SVM also show this drawback but to a lesser extent.

This prediction of accessing descriptors is still in a preliminary stage. The goal is to know the services that will be rated by users in advance, thus being able to customize the appearance of the portal. Since the achieved results are encouraging, its actual applicability will be studied for I-XXI. Further work will involve (1) a more appropriate criterion for model comparison, (2) developing models capable to predict with less information than accesses from three sessions, and (3) to predict beyond one-session ahead.

3 Clustering web users

3.1 Data

Two kind of data sets were taken into account: artificial data sets and real data sets. Artificial data sets were produced by the user model simulating web user accesses (Fig. 1), and were used to benchmark the clustering algorithms in a wide range of scenarios.

Real data sets came from I-XXI. Since dealing with a real data set, the groups of users were not known, and therefore, the evaluation could not be so straightforward as with artificial data sets. Due to this fact, a previous experiment with only five descriptors was used to determine the interpretability of the clustering achieved. The selected descriptors were representative-enough to carry out an interpretability test, namely, public administration, town councils, channels, shopping and entertainment. This reduced data set was formed by 1,676 users who interact with the site from November 2002 through

January 2003. The complete data set consisted of 31,483 accesses from June 2002 to February 2003. The 17,404 accesses corresponding to the first half of this period were used to cluster of web portal users. The other 14,079 accesses were kept to evaluate a prediction analysis, detailed later, which is used as the first step of a recommender system. A preprocessing stage eliminated some descriptors, being 16 the final number of descriptors taken into account.

3.2 Methods

We considered different methods for clustering web users[5], including classical clustering algorithms, such as, CM or FCM, and also the E-M for a Gaussian mixture model and HCA [3]. We also took into account two clustering methods which are based on neural networks, namely, SOM [1] and a network based on the ART (ART2) [4].

ART2 is particularly interesting in the clustering of real data because it is not necessary to asume a given number of clusters in advance, but it is automatically produced by the algorithm once chosen the vigilance parameter. The vigilance parameter measures the desired degree of similarity between a cluster prototype and a pattern in order to consider that the prototype is a proper representation of the pattern. This parameter can show a range of values between 0 and 1, being typical values those ranging between 0,8 and 0,99. The choice of a logical vigilance parameter should lead to a natural number of clusters. Moreover, it is especially interesting to analyze how the number of clusters changes when the vigilance parameter is changed; this is because abrupt changes should help in finding the "natural" number of clusters underlying in a certain data set.

The other algorithms do not show this advantage, and the number of clusters has to be chosen by analyzing the goodness and robustness of the clustering as well as by using some cluster validity parameters, such as, Davies-Bouldin and Dunn indices [5].

3.3 Results

With regard to the artificial data sets, evaluation of the clustering algorithms was done by two kind of measures: we considered whether the number of clusters found by the algorithm was correct, and, also, the accuracy of these clusters. These measures were assessed by the Bhattacharyya distance between the clusters found by the algorithm and the actual ones. The two measures should be assessed together. The best overall behavior was shown by SOM and ART2 while CM performance was quite worse than any other. In fact, CM only presented an acceptable behavior in quite simple data sets (low dimensionality and slight or non-existent overlap among clusters). Besides this, FCM showed a slightly better behavior, and, E-M and HCA, in turn, had a better behavior. As a general conclusion, when the data set to be clustered is supposed to be difficult, or the dimensionality is high or the characteristics are not known "a priori", then SOM or ART2 become the most suitable choice.

Results obtained with artificial data sets may be considered as overoptimistic, except in the case of ART2. This is because all other algorithms use the number of clusters,

[5] Clustering was carried out in descriptors' space due to the high dimensionality of the space defined by services.

which is known in advance, as an input parameter. Therefore, the comparison is not completely fair, and ART2 should be considered as the algorithm providing the best performance.

With regard to real data sets, as quoted above, a preliminary clustering with a reduced data set was carried out in order to analyze the feasibility of extracting useful information from clustering. The aim was to study the obtained clustering in terms of its interpretability. Clusterings provided by HCA, SOM and ART were quite straightforward to interpret, thus showing logical behaviors that were expected to find in this web portal. On the contrary, CM, FCM and E-M produced clusters harder to interpret, what suggested that these three algorithms should not be used to cluster data of this kind. The analysis of intepretability about the clustering achieved was carried out by analyzing users' behavior in collaboration with people in charge of I-XXI. According to the results obtained with artificial data sets and with the preliminary clustering of I-XXI, the conclusion is that the most suitable algorithms to cluster users of this portal are SOM and ART2. Despite HCA does show an acceptable behavior in the preliminary clustering of I-XXI, its use is not recommended for clustering the complete real data set because the dimensionality is considerably high (16 descriptors are taken into account) and the analysis with artificial data sets showed that HCA was not an appropriate choice for clustering of high dimensional data sets. Numerical results about clustering can be found in [6].

4 Recommendation procedure

4.1 Methods and Data

Recommendation procedure consists of two main parts: study of the feasibility of recommendations and the provided recommendation itself. The feasibility of recommendations is studied by means of a prediction analysis. The idea is to check whether the services that would be recommended by the system are actually accessed by users. This approach presents two main advantages. (1) The first one is that it allows an evaluation of the recommender system before its actual implementation. (2) An additional advantage is that this prediction analysis does not take into account the interface of the recommendation; therefore, the success in the prediction only depends on the user profiling carried out by the clustering analysis, and the success of acceptance of recommendations can be interpreted as a lower threshold of the expected success when the recommendations are offered to users; this is because the presentation of attractive items should affect users' behavior positively. The success of recommenders was measured by comparing the percentage of accepted recommendations in a naïve recommender that recommended the most likely service of the portal with collaborative recommenders based on the different clustering algorithms used in this study (both kind of recommenders accomplishing the premise that the recommended services had not yet been accessed). The 14,079 accesses of the data set described in Section 3.1 were used.

In order to take advantage of new users when they interact with the recommender engine, two approaches are proposed (currently, they are still in an early development stage, and therefore, the performance of both methods has not yet been compared). The first approach is based on LVQ. It is a methodology for rapid updating of recommender

systems by re-estimating the underlying clusters in response to user behavior. The LVQ learning algorithm involves two steps. In the first step, an unsupervised learning data clustering method is used to locate several cluster centers without using the class information. In the second step, the class information is used to better locate the cluster centers in order to minimize the number of misclassified cases [1]. In our case study, this second step enables a proper fine-tuning of cluster centers, and in turn, it can improve the recommendations by using new users' behavior. A simple modification is introduced to the standard LVQ's update algorithm: the update keeps with no change if users are correctly classified, i.e., if they do click on the recommendation, but there is no update otherwise. In other words, we do take into account users who accept recommendations, but we ignore users who have not accessed the proposed recommendations; this is because we feel that users who accept a recommendation are likely belonging to the class in which they have been classified; nevertheless, users who do not accept the recommendation may have been well-classified, but perhaps they do not have enough time to consult the recommendation, or maybe they know exactly what they want from the web site; if we consider missclassified these kind of users, we will move the cluster centers away from these users, and thus, we may lose part of the information obtained from users who did accept the recommendation.

The other approach is based on RL. RL is based on learning from interaction to achieve a goal. The learner and decision-maker is called the *agent*. What it interacts with is called the *environment*. Here, our agent's environment consists of the set of web users. In RL, the objective of every agent is to maximize a numerical sum of rewards in time. This reward is a non-obstrusive way of getting feedback from interactions between users and the recommender system [7]. Every recommendation receives a negative reinforcement because; this way, the recommendation agent learns that every useless recommendation has a cost. However, each time a user follows a recommended link, the agent receives a positive reinforcement. Reward is never assigned to a concrete recommendation, but to a whole policy of recommendations. Besides, it is important to point out that recommendation agent's adaptation does not stem from updating a model, which is a restriction of the LVQ approach.

4.2 Results

With regard to the prediction analysis, a collaborative filtering based on clustering yields a much higher Success Rate (SR) than that obtained by a naïve recommender [6]. It should be noted that services based on clustering cannot be recommended for the very first accesses, since there is not enough information to assign users to a certain cluster; instead, the naïve recommender is used for these first accesses. The SR is measured as the percentage of times that users actually click on the recommended object.

Results achieved with artificial data sets and with the reduced real data set showed that SOM and ART2 should be the chosen techniques. Nevertheless, the classical CM was also taken into account in order to have a more appropriate threshold than just a naïve recommender and also to check out that neural clustering techniques are indeed much more accurate than classical CM when dealing with a real data set. SRs obtained by collaborative recommenders based on neural clustering were much higher than that obtained by the naïve recommender, and also by that obtained by the classical CM

clustering. Moreover, the SRs obtained by SOM and ART2 were almost identical. Typically, the percentages achieved by SOM and ART2 were double than those obtained by the other recommenders (typically, SRs were raised from 7.5% using a naïve recommender to 15% using a recommender based on neural clustering). As more accesses were used to cluster, better results were obtained; this was expected, since the information gathered by the clustering algorithms was more extensive. At present, we are still developing the recommender system update, but results obtained with LVQ and RL in other applications make us be optimistic.

5 Conclusions

We have presented three ML approaches in order to improve I-XXI. First, we have presented a preliminary study of web access prediction, which has yielded promising results. Future work will be focused on extending the current model into a fully personalized recommender system improving the clickstream prediction. A methodology for web user profiling and recommendation has also been proposed. The methodology starts with a user model which simulates accesses to a citizen web portal. These accesses are used to benchmark different clustering algorithms. This comparison is then used to determine the most appropriate clustering technique for each kind of web portal. Collaborative filtering techniques based on clustering are used to take advantage of user profiling in a final recommendation stage. A preliminary study of recommendations by means of a prediction analysis has shown excellent results. Our ongoing research is dedicated to test real recommendations, thus improving the recommender system by either updating the collaborative recommender or using RL.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Englewood Cliffs, NJ (1999)
2. Gorse, D., Sheperd, A., Taylor, J.: The new ERA in supervised learning. Neural Networks **10** (1997) 343–352
3. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press, San Diego (1999)
4. Carpenter, G., Grossberg, S.: Pattern Recognition by Self-Organizing Neural Networks. MIT Press (1991)
5. Bezdek, J.C., Pal, N.R.: Some new indexes of cluster validity. IEEE Transactions on Systems, Man and Cybernetics **28** (1998) 301–315
6. Martín-Guerrero, J.D.: Determinación de tendencias en un portal web utilizando técnicas no supervisadas. Aplicación a sistemas de recomendaciones basados en filtrado colaborativo (in Spanish, with abstract in English). PhD thesis, University of Valencia, Spain (2004)
7. Hernandez, F., Gaudioso, E., G. Boticario, J.: A reinforcement learning approach to achieve unobstrusive and interactive recommendation systems for web-based communities. In: Lecture Notes in Computer Science, Proceedings of AH2004, Eindhoven, The Netherlands (2004) 409–412

Log Pre-Processing and Grammatical Inference for *Web Usage Mining*

Thierry MURGUE[1,2]

[1] EURISE, University of Saint-Étienne 23, rue du Dr Paul Michelon
42023 Saint-Étienne cedex 2 – France
[2] RIM, École des Mines de Saint-Étienne 158, Cours Fauriel
42023 Saint-Étienne cedex 2 – France
thierry.murgue@univ-st-etienne.fr

Abstract. In this paper, we propose a WEB USAGE MINING pre-processing method to retrieve missing data from the server log files. Moreover, we propose two levels of evaluation: directly on reconstructed data, but also after a machine learning step by evaluating inferred grammatical models. We conducted some experiments and we showed that our algorithm improves the quality of user data.
Keywords: log pre-processing, web usage mining, grammatical inference, evaluation

1 Introduction

WEB USAGE MINING is a complex process used in order to extract knowledge about users of a web site. It is composed of many steps as described in Fig.1, from selecting relevant data to knowing how users browse on a web site. WEB USAGE MINING was first introduced in 1997 [1] as the "discovery of user access patterns from Web servers". To select, to clean and to format available data are real challenges. The kind of information which is the most used in this context is the Web server log files. Already in 1995, Catledge and Pitkow used logs for the characterization of "Browsing Strategies" [2]. Other types of data can be used in this topic of research, such as client data (mouse gestures, keyboard events,...) or user physical behavior (eye movements, arm gestures,...), but they are really hard to obtain. As a consequence, most work in this research field depends on server log files. Due to some network architectures and features (*cache* and *proxy*), log data are often irrelevant and noisy. Pre-processing is therefore a crucial issue for learning. Machine learning methods for WEB USAGE MINING include frequent sequences learning [3, 4] and more structural models such as Hidden Markov Models (HMMS) [5, 6]. More recently, researchers have worked on grammatical models: n-grams [7], and stochastic automata with classical grammatical inference methods [8].

In this paper, we propose a method for handling the problem of noisy and irrelevant data in log files. We present an algorithm to reconstruct the data and we evaluate it. We use stochastic inference to learn users behaviors and we show that reconstructed data leads to better models.

We will first present the problem we want to deal with. The description of an algorithm to reconstruct the log data in order to obtain reliable ones follows: the method

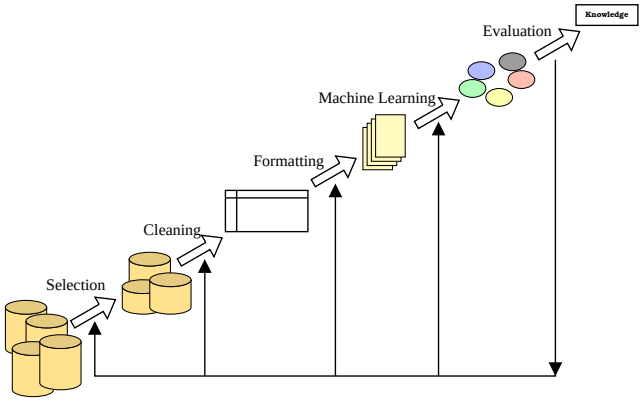


Fig. 1. Web Usage Mining process

depends on some heuristics which try to detect improbable paths on navigation. Two sets of experiments show that we can retrieve some pieces of information which were lost due to caches. Finally, we conclude and propose alternative protocols in order to test log processing methods.

2 Our Problem

In order to mine user behavior, we have to extract from log files each visit of users to the web site: a visit is considered as a set of pages requested in a single semantic goal (without any embedded objects such as pictures and so on). We will see that they are missing and erroneous data in server log files. Before starting to explain our problem, we have to say more about logs.

The World Wide Web Consortium (W3C) recommends to use the *common log file* format to record required fields [9].

Table 1. a common log record

F1	F2	F3	F4	F5	F6	F7
161.3.6.51	-	-	[30/Oct/2001:20:13:27 +0100]	"GET / HTTP/1.1"	403	293

Generally, a log record is composed of (see Table 1):

- `remotehost` field (F1): the address of the machine which makes the request.
- `rfc931` field (F2): the remote login name of the user. Actually, for obvious security reasons, this field is almost always blank.
- `authuser` field (F3): the username as which the user has authenticated himself. This field is only filled when an authentication process is used.
- `date` field (F4): data and time when the server receives the request.

– `request` field (F5): the request line exactly as it came from the client.
– `status` field (F6): a return code informing the client of the request processing status.
– `bytes` field (F7): the content-length of the answer transferred.

Servers record chronologically each hit into the log files: a hit can be a real page request, but also an error or an embedded document request which is not related to the navigation of the user. First of all, we have to filter the logs, by deleting all useless items such as error reporting, non-supported requests (for example, in our preliminary version we did not process any page with parameters as `index.php?id=12&it=la`). The result of filtering raw data is a sequence of logs, which represents only a part of user navigation: actually, many requests never reach the server.

In order to reduce network traffic and to speed up the transfer of data on the Internet, the almost totality of browsers (mozilla-firefox, opera, IE,...) implement a local cache. In [10], caching is defined as "a mechanism that attempts to decrease the time it takes to retrieve a resource by storing a copy of the resource at a closer location". Consequently, when a user requests a page:

1. the cache checks if it has already been viewed by the user (typically when user has hit *back button*);
2. (a) if not, the client sends the request to the server;
   (b) else, the cache produces the answer but the server never receives the request and so can't record it.

This is the first problem due to the network features: it implies that some data is missing in server log files.

Another place where a cache can be, and it is called *global* cache in this context, is on a *proxy* server: a proxy is a machine somewhere in-between client and server; its goal is to increase the security and to make easier the administration of a machines set (a domain) by allowing only connection to external web servers from the proxy. Thus, a machine from the protected domain can request a page only to the proxy:

1. the proxy checks in its own cache, if it has already been viewed by one of the users using the proxy;
2. (a) if not, the proxy forwards the request to the server: in this case, the *from field* recorded into the logs is the proxy address and not the client address;
   (b) else, the cache produces the answer but the server never receives the request and so can't record it.

In this configuration, external servers receive requests only from the proxy, so in the log files, we can't distinguish where the request comes from.

So, to retrieve the exact visits of each user, we have to deal with different kinds of noise: erroneous data, when the from address is a proxy and not the real user's machine and missing data, when the cache instead of the server supplies the answer.

3 Reconstructing data: algorithm WhichSession

In order to retrieve the missing data and correct the wrong data, we suppose that the web site corresponding to the server is known: we model the web site by a graph where

nodes represent pages and links represent hypertext links into the page. The heuristics we use detect inconsistent sequences of pages in the user navigation: the algorithm WhichSession is described below.

Algorithm 1: WhichSession

```
Data: h(0) //page we want to class
foreach k ∈ opened_sessions do
    if h_k(1) → h(0) then
        Store(k, h(0)) /*add h(0) in session k and exits */
min_back_length = ∞;
//the number of backward steps, we have to follow to find a linked page
foreach k ∈ opened_sessions do
    for 2 ≤ i ≤ |h_k| do
        if h_k(i) → h(0) then
            if min_back_length > i then
                min_back_length = i ;
                min_session = k; break;
if min_back_length < ∞ then
    /*we found an inconsistency into log records, we have to include a part of the history
      in order to have a linked path into the web site */
    Add_Missing_Values(min_session, min_back_length);
    Store(min_session,h(0));
else
    /*there's no available page in histories: add a new session */
    l = New_Session();
    Store(l,h(0));
```

The symbol $\rightarrow$ is used to indicate that there is a hypertext link from the first page to second one. Some notations are to be defined: for the situation at the current time, there are some sessions opened and we want to store the current log record $h(0)$ into the right session, the one corresponding to the unique user who requests the page. We have to select the best session, by scrolling through the history of opened sessions (h_k is the history of the session k, $h_k(i)$ is the ith page of h_k), and choosing the one which has the minimum backward steps required to find a link to the current page.

4 Artificial data experiments

4.1 Artificial Data presentation

In order to have reliable data to test our reconstructing method, we generated artificial logs according to possible paths in a web site. We select a page and we compute a path

with some "navigation errors" that could typically have been produced by hitting the *back button* or no taking the shortest path. The site is composed of 105 different pages, the average length of generated paths is 10.9 pages. At this step, we obtain a reference complete set of logs l_0 without erroneous data. Then we simulated some caches from which we obtained l_1 and finally rebuilt the data l_2 from l_1 using algorithm WhichSession.

4.2 Data evaluation

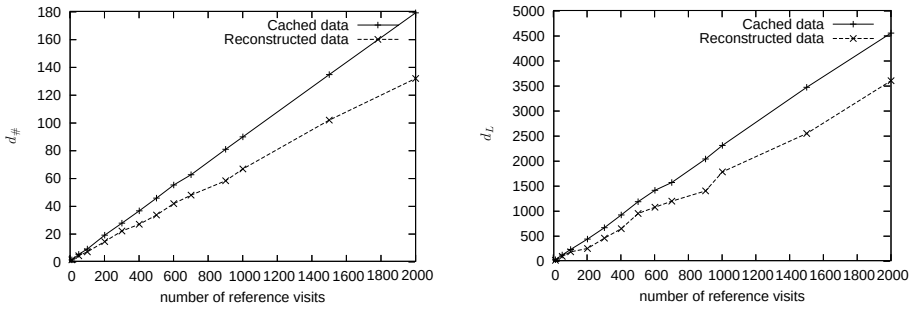
We first carried out a set of experiments on these data by computing two ratings between l_0 and l_1 on one hand, and between l_0 and l_2 on the other hand: one measures the difference between the number of detected visits ($d_\#$), the other the Levenshtein's distance (d_L) [11] for each visit. In order to do that, we have to re-assign logs into their reference visit because of the lack of detected visits.

As an example, if we have a set of three visits (v_i) with two pages (A and B) into the reference artificial data such as:

reference visits l_0	cached ones l_1	reconstructed ones l_2
$v_0 = [A, B, A]$	$v_0 = [A, B, B, A]$	$v_0 = [A, B]$
$v_1 = [B, B, A]$	$v_1 = [B]$	$v_1 = [B, B, A]$
$v_2 = [A, B]$		$v_2 = [A, B]$

One can compute $d_\#(l_0, l_1) = 1$, $d_\#(l_0, l_2) = 0$. By comparing date of logs (it is possible because we have generated the data), we can reorganize l_1 into $v_0 = [A, B]$, $v_1 = [B, A]$ and $v_2 = [A, B]$ and then compute $d_L(l_0, l_1) = 3$, $d_L(l_0, l_2) = 1$.

Results of the experiment are shown in Fig. 2.



(a) $d_\#$ (b) d_L

Fig. 2. $d_\#$ and d_L behaviors

We can see that the two ratings are better when the data are reconstructed by our method. Actually, reconstruction permits to retrieve correct missing data: that helps to detect the end of a visit, and gives better knowledge about the navigation.

This protocol allows testing the method only if we know exactly the initial visits without any kind of noise. With real data, we can't have this *a priori* knowledge, so we decided to learn grammatical models and to test them.

4.3 Models evaluation

Data for this set of experiments are the same as for the preceding one: we generated them. Here, instead of computing ratings on cached and reconstructed visits, we used them to learn grammatical models. We learn stochastic models which can predict the next most probable symbol of a sequence: here sequences represent visits and symbols are pages. We use the MDI [12] algorithm to learn such a model for the cached data and such a model for the reconstructed ones.

We generated also a sample set of ideal visits: for each visit step, we compared the real next page in the visit with the most probable one proposed by the model. We count a success if the pages represent the same one, and a failure otherwise.

Results are shown in Fig. 3.

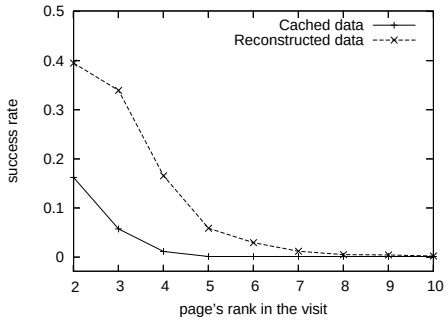


Fig. 3. Success rate in learning task on cached and reconstructed data

As presented in [4], it is a hard problem to predict a page after the few first ones, so usually authors do not try to check after the 7th page and their success rate really depends on the number of site pages. In our case, on different data, we can see that without any reconstruction, after the 5th page, we can not predict anything. But, with our method we learn better models and we can predict with a higher probability the next page. At each fixed position in the visit, we do better prediction, and for a fixed success rate, we can predict at a further position.

5 Real data problem

Experiments on artificial data can help us to test the reconstruction of data, but the main problem is to mine user behavior, so we have to experiment also on real data.

For that, we took data from a real server log file (≈ 137000 logs, 100 pages, mean of 11.4 links per page) [13]. Then, we extracted visits from these data and also from reconstructed ones by our method. Here the protocol is quite the same as for artificial data: we learned one model for each kind of data (raw and reconstructed), and we evaluated models.

The main problem in this context is that we do not have perfect data to use as a sample set. We decided here to test with two different sets: one which is composed of a part of the log file (not used in learning step) for testing the model on raw data, and the other is the reconstruction of this same part by our method for testing model on reconstructed data.

We describe in Fig. 4 the results.

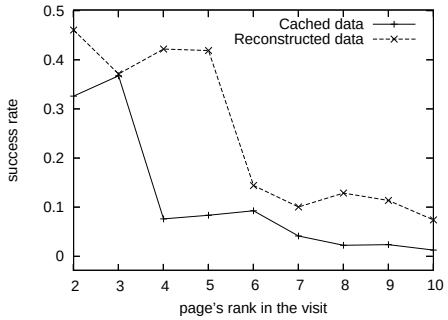


Fig. 4. Success rate in learning task on raw and reconstructed data

When we use reconstruction, results are better. Actually, we can view the two types of data as two different kinds of answers for the problem: learning model on raw data comes to predict the next page requested for the servers, taking into account not only the user behavior but also the disruptive caching features; the model on reconstructed data can be viewed as a true user behavior model without any other kind of perturbation.

6 Conclusion

In this paper, we showed on artificial data that reconstructing log files data can avoid some perturbations due to caching features. With both types of evaluation, ratings on data and prediction evaluation, results are better with reconstruction.

On the real data, there remains a problem: which data can be used to evaluate models? We do not have perfect data to test the quality of the reconstruction: testing on reconstructing sample set introduces a small bias. In an optimal context, we have to have complete data and cached ones in order to test exactly the predictions model.

Results given by grammatical inference are really interesting: they are as good as other types of models shown in state of the art for the usage mining. We want to continue in this field by showing that these models which permit long term dependence are really good for this task.

References

1. Cooley, R., Srivastava, J., Mobasher, B.: WEB Mining: Information and Pattern Discovery on the World Wide Web. In: Proceedings of ICTAI'97. (1997)
2. Catledge, L.D., Pitkow, J.E.: Characterizing Browsing Strategies in the World-Wide Web. Computer Networks and ISDN Systems **27** (1995) 1065–1073
3. Frias-Martinez, E., Karamcheti, V.: A Prediction Model for User Access Sequences. In: WEBKDD Workshop: Web Mining for Usage Patterns and User Profiles. (2002)
4. Géry, M., Haddad, H.: Evaluation of Web Usage Mining Approaches for User's Next Request Prediction. In: Proc. of WIDM'03, New Orleans (2003) 74–81
5. Pitkow, J., Pirolli, P.: Mining Longest Repeating Subsequences to Predict World Wide Web Surfing. In: Proceedings of USITS'99. (1999)
6. Bidel, S., Lemoine, L., Piat, F., Artières, T., Gallinari, P.: Statistical Machine Learning for Tracking Hypermedia User Behaviour. In: MLIRUM - UM Workshop, Pittsburgh (2003)
7. Borges, J., Levene, M.: Data Mining of User Navigation Patterns. In: WEBKDD. (1999) 92–111
8. Karampatziakis, N., Paliouras, G., Pierrakos, D., Stamatopoulos, P.: Navigation Pattern Discovery Using Grammatical Inference. (In: Proc. of ICGI04) 17–56
9. Luotonen, A.: The Common Log File Format (1995) http://www.w3.org/Daemon/User/Config/Logging.html.
10. Pitkow, J.: In Search of Reliable Usage Data on the WWW. In: Proceedings of the Sixth International WWW Conference, Santa-Clara, CA (1997) 451–463
11. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics - Doklady **10** (1966) 707–710 Translated from Doklady Akademii Nauk SSSR, Vol. 163 No. 4 pp. 845-848, August 1965.
12. Thollard, F., Dupont, P.: Entropie relative et algorithmes d'inférence grammaticale probabiliste. In: Conférence sur l'apprentissage automatique, CAP'99, Paris (1999) 115 – 121
13. Eurise: Web site. http://eurise.univ-st-etienne.fr (2002)

Automatically building domain model in hypermedia applications[1]

Hermine Njike, Thierry Artières, Patrick Gallinari, Julien Blanchard, Guillaume Letellier

LIP6, Université Paris 6
8 rue du capitaine Scott, 75015, Paris, France
{Firstname.Lastname}@lip6.fr

Abstract. This paper deals with the automatic building of personalized hypermedia. We build upon ideas developed for educational hypermedia. A standard way to build adaptive educational hypermedia relies on the definition of a domain model and the use of overlay user models. Since much work has been done on learning user models and adapting hypermedia based on such user models, the core problem lies in the automatic definition of a domain model for a static hypermedia. We describe an approach to automatically learn from the hypermedia content such a domain model. This model is a concept hierarchy where concepts are identified by sets of keywords learned from the collection. We propose the use of visualization techniques such as treemaps in order to monitor and analyze efficiently user and domain models.

1. Introduction

Adaptive hypermedia aim at offering personalized hypermedia and websites to a user. It relies on user models that consist in static and dynamic information such as goals, preferences... A domain model may be used that characterizes the whole knowledge accessible in the hypermedia it is used to infer information in the user model [2, 6]. Many works have been done in adaptive hypermedia that one can distinguish according to the nature of the task. Maybe the most well defined problem concerns educational hypermedia and tutorial systems [5, 6, 7]. Although building such systems is still difficult, the task is indeed well identified; in such systems domain models are often manually designed and defined as a set or a graph of the concepts being discussed in the hypermedia. Overlay user models share the same representation as domain models and are used to represent a user knowledge and/or interest in the concept space [5, 6, 9]. These user models are vectors of attributes (e.g. interest) one for each concept in the domain model. These are updated from user navigation logs according to the domain model, a popular way to make inference in these models is to use

[1] This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors'views.

Bayesian Nets [5, 8, 18] since these models allow taking into account relationship between concepts, inferring and propagating information in nodes. A more difficult task that has been less studied up to now, concerns adaptive systems for any single website or hypermedia [1, 13, 18]; the domain model, that is the set of topics or concepts, is often wider and much less explicit, so that the task is much harder. In [18] a very simple approach has been proposed where domain model is derived straightfully from the website structure, the UM 2001 conference website. Concepts of the domain model correspond to pages in the site (with nodes corresponding to *Paper Submission*, *Call For Papers*, etc) and are organized as a hierarchy derived from the website structure. Unfortunately, viewing the domain model as a clone of the site structure may fail when the structure is weak or not so much related with the underlying concepts discussed in the pages of the hypermedia.
We are interested in this paper in developing techniques allowing the automatic building of personalized hypermedia. To do this, we believe that one can take advantage of works done in the educational hypermedia field concerning the learning of overlay user models and the personalization of hypermedia based on overlay user models. In this context, the core problem for the automatic conception of a personalized hypermedia lies in the automatic learning of a relevant domain model. This is not however an easy task. This model involves high level concepts that cannot be easily inferred automatically. We present an approach that allows learning automatically a concept hierarchy from a corpus of documents (e.g. pages of a website). Concepts in this hierarchy are organized according to a generalization / specialization relation and document subsets may be associated to each concept. Such a representation of the hypermedia thematic content allows defining relevant overlay user models. Also, visualization of this representation using treemaps provides an alternative view of the hypermedia by reorganizing its content according to the learned hierarchy. This may be used for easy monitoring of user models or for users to browse this new interface. We first describe our approach, then we discuss how this approach may be used in the context of user modeling for learning automatically domain models.

2. Discovering concepts from a collection of pages

We study now how to learn automatically concept hierarchies from collections of documents (e.g. the pages of a website) according to a generalization/specialization relation. Our aim is to build generic tools to extract simple semantic relations between corpus elements, which can be used to build domain and user models. Our method starts by automatically learning concepts from a corpus, and then learns generalization/specialization relations between these concepts.
Several approaches were developed in information retrieval for the generation of hierarchies. Clustering techniques have often been used to create document hierarchies. However, there is no semantic relation between the nodes at different levels in these hierarchies. As a consequence, these works are practically useless regarding our goal. Recently new types of hierarchies which are automatically built from corpora have been proposed [10, 11, 16]. These are term hierarchies built from generalization/specialization relations automatically discovered between terms in a corpus. Once this term hierarchy is built, it is possible "to project" documents on it, thus pro-

ducing a document hierarchy. We propose to extend these approaches to the discovery of a concept hierarchy where concepts, which are discovered from the corpus, are represented as sets of keywords and not by single terms. Such a representation allows for a richer description than single terms, thus better reflects the different ideas which appear in documents. We detail in the following the main steps of the procedure. For clarity of presentation, we consider in the following that a hypermedia is decomposed in units (e.g. pages of a website), that we will call documents. The corpus, a set of documents, is first preprocessed and segmented into homogeneous paragraphs. The segmentation task consists in identifying, in each document, homogeneous text regions or frontiers corresponding to topic shifts between such regions. Next, all these paragraphs are clustered in order to determine groups of paragraphs related to a similar topic. Each discovered topic is considered then to be a concept of the collection. A by-product of this step is that each cluster (i.e. a concept) is represented as a set of words. Based on a set of concepts, a document may be classified according to the concepts it addresses. Finally, specialization/generalization links are discovered between concepts using a subsumption measure between concepts.

2.1. Pre-processing and document representation

The system's input is a set of documents. All documents are preprocessed as usual in information retrieval tasks; non informative words are removed, all remaining words are lemmatized. Let $V = \{w_j\}_{j\in\{1,\ldots,M\}}$ be the vocabulary of M lemmatized words, $D = \{d_i\}_{i\in\{1,\ldots,N\}}$ be the set of documents in the collection (after preprocessing), and $P=\{p_k\}_{k\in\{1,\ldots,L\}}$ the set of paragraphs of documents in D. Representations of documents and paragraphs are M dimensional vectors. A document d_i is represented as a vector of weighted frequencies (tfidf) for terms in V, $d_i = (tf_i(w_1).idf(w_1),\ldots,tf_i(w_M).idf(w_M))$ where $tf_i(w_j)$ is the frequency of term j in D_i and $idf(w_j) = \log(N / df(w_j))$, where $df(w_j)$ is the number of documents in D containing term w_j. Similarly, a paragraph p_k is represented as a vector: $p_k = (tf_k(w_1).ipf(w_1),\ldots,tf_k(w_M).ipf(w_M))$ where $tf_k(w_j)$ is the frequency of term j in p_k and $ipf(w_j) = log(L/dp(w_j))$, with $pf(w_j)$ the number of paragraphs containing w_j.
The similarity measure between two entities (documents or paragraphs) is the classical cosine between their vector representations used in information retrieval.

2.2. Segmentation step

The segmentation task consists in identifying in a document (i.e. a page), homogeneous text regions or frontiers corresponding to topic shifts between such regions. We used the technique proposed in [15]. This method proceeds by decomposing texts into segments and topics, a segment being a bloc of contiguous text about one subject and a topic being a set of such segments. Here is the sketch of the algorithm, which starts at the paragraph level (Paragraphs are the basic text unit) since authors generally expose one point of view per paragraph. For each document, repeat until convergence:

- Compute similarities between all paragraphs in a document and keep those higher than a given threshold.
- Build a similarity graph and extract triangles. A triangle is a set of three paragraphs with strong similarities, i.e. susceptible to represent a coherent topic.
- For each triangle, build its vector representation which is the average of the three vectors representing the paragraphs of the triangle.
- Merge the triangles whose similarity is higher than a given threshold.

This procedure is used for any document in *D*.

2.3. Clustering topics

Once each document is decomposed into a set of topics, we cluster these topics in order to identify a representative set of concepts for the corpus:

- Build a graph based on similarities between topics identified above using the method by Salton (1996) (i.e. there is an edge between two topics if the similarity is higher than a given threshold).
- Compute the connected components of this graph. For each component, keep only nodes which are connected to at least 75% of the other nodes of the component.
- A component with at least β% of its documents (β has been fixed around 90% in our experiments) in a second component will be merged with the latter.

At last, each remaining component is considered as a concept of the corpus. The concept representation is a set of most significant keywords (e.g. with highest *tfidf* measures). From now on we will identify "concepts" and their sets of keywords.

2.4. Inferring « generalization/specialization » relations between concepts

One main idea of our method is to infer generalization/specialization relations between concepts that are identified by sets of keywords. Quite generally, there exists a "generalization/specialization" relation between entities *C1* and *C2* if *C2* evokes a specificity of *C1*, or is about specifics themes of *C1*. For example *C1* = sport and *C2* = football. Most document hierarchies make use of simple concept representations where a concept is identified with a single keyword. For such a representation, Sanderson (1999) proposed a method for automatically inferring term hierarchies by learning a generalization/ specialization relation between terms; it is based on term subsumption. The idea is that some terms which occur frequently in a collection give significant information about the concepts discussed in the corpus. These terms may define a subject in a general way, whereas others which co-occur with these general terms and are less frequent explain some aspects of the subject. The subsumption measure characterizes a relation of generality/specificity between two terms and is based on asymmetrical terms co-occurrences. It is defined as follows: Term *x* subsumes (i.e. is more general than*)* term *y* if $P(x|y) > th$ *and* $P(y|x) < P(x|y)$, where *th* is a threshold.This means *x* subsumes *y* if documents in which *y* occurs are a subset or nearly a subset of the documents in which *x* occurs. The second rule ($P(y|x) < P(x|y)$) ensures that if both terms occur together more than *t%* of the time, the most frequent term will be chosen as the more general. Probabilities $P(x/y)$ may be approximated through counting $P(x|y) = n(x,y) / n(y)$ where $n(x,y)$ is the number of documents that contain terms *x* and *y*, and $n(y)$ is the number of documents that contain term *y*.

Now recall that a result of the previous step is that each concept is identified by a set of keywords. We extended the term subsumption measure described above to concept subsumption, where each concept is represented by a set of keywords. The method consists in computing conditional probabilities $P(Ci|Cj)$, the probability that a document discussing of concept C_j discusses also of concept C_j. Estimating such probabilities for any pair of concepts allows applying the subsumption definition directly to the concepts. Once the relations of "generalization / specialization" are detected on pairs of concepts, we apply transitivity to build the concept hierarchy.
The main problem is to compute probabilities $P(Ci|Cj)$. It could be estimated with $P(C_i|C_j) = n(C_i, C_j)/n(C_i)$ where $n(C_i, C_j)$ stands for the number of documents dealing with concepts C_i and C_j and $n(C_i)$ stands for the number of documents dealing with concept C_j. This estimation is rather poor. Another way is to approximate posterior probabilities $P(C|d)$, that a document d discusses concept C or not, which is not easy. At this point, the result of the document segmentation step could be used to assign concepts to the documents. If a paragraph in document d belongs to concept C then $P(C|d)$ is non zero. It could be set to a real value, by measuring e.g. the importance of the paragraph in the document. However, this provides a crude estimation of $P(C|d)$. In our system, we propose to estimate $P(C|d)$ via an Estimation / Maximization (EM) algorithm. This algorithm iteratively computes probabilities $P(t|C)$ for all concept C and vocabulary term t, through maximizing the likelihood of the document collection. Assuming a naïve Bayes model for documents, it allows computing $P(d|C)$ and therefore $P(C|d)$ via Bayes rule. It aims at maximizing training data log likelihood:
$Log(L) = Log(P(D|\Theta)) = \sum_d \sum_{t \in d} log(\sum_C P(t|C, \Theta) P(C|\Theta))$
where Θ stands for the model parameters. The EM algorithm alternates estimation of hidden variables $P(C|d)$ and reestimation of model parameter $P(t|C)$. At each step, documents are considered to discuss of concept C if $P(C|d)$ is over a threshold.
Note that with this definition, a concept may have several parents: This corresponds to different meanings of this concept and reflect its polysemia. Also, an important remark concerning this subsumption measure between concepts is that it is suitable in domains where terms are often repeated. If this was not the case, the co-occurrence estimations would not be robust enough to be relevant. However, one could reduce the sensitivity of the technique to corpus variability by using linguistic resources like WordNet to take into account synonymy.

3. Discovering domain model in hypermedia

We applied our approach to the discovery of a domain model, i.e. a concept hierarchy, of a collection of documents, which is a part of the www.looksmart.com site hierarchies. One interest of this corpus is that we can compare, after learning, the discovered hierarchy and the manually designed one. Quantitative evaluation criteria may be defined for estimating generalization / specialization expressiveness of a hierarchy [12], we will mainly show visually our results here since it is more related to our goal. The corpus consists in about 100 documents and 7000 terms about artificial intelligence and is a homogeneous set of documents. This collection has been manually organized in hierarchies of themes. We extracted a heterogeneous sub-hierarchy from this site with documents about different topics. We ran the method on the flat corpus, without any use of the hierarchical information. Compared to the initial Looksmart

hierarchy with five categories, the hierarchy derived by our algorithm is much larger and deeper. Most of the original categories are refined by our algorithm. For example, many sub-categories emerge from the original "Knowledge Representation" category (see Figure 1): ontologies, building ontologies, KDD… and most of the emerging categories are themselves specialized. In the same way, "Philosophy-Morality" is subdivided in many categories like AI definition, Method and stakes, risks … It is clear that such a result could not have been obtained using single keyword concepts.

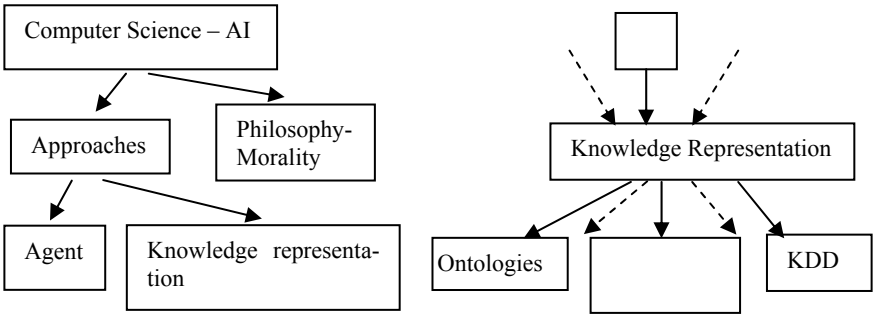


Fig. 1. Sub-hierarchy of the LookSmart corpus used in our experiments (left) and part of the deeper hierarchy discovered using our approach (right).

An interesting feature of this hierarchical organization is that it allows using visualization tools. We considered the use of Treemaps that have been introduced by [17]. The idea of treemaps is to display a tree-like structure in a 2D space where each node is represented by a rectangle whose size or color is determined by a value, it could be the user interest in a concept in our case. Fig. 2 shows a Treemap representing the Looksmart domain model. Each concept is shown as a rectangle with different colour, the hierarchy is shown through inclusion of rectangles. Set of keywords associated to intermediate concepts are also shown.

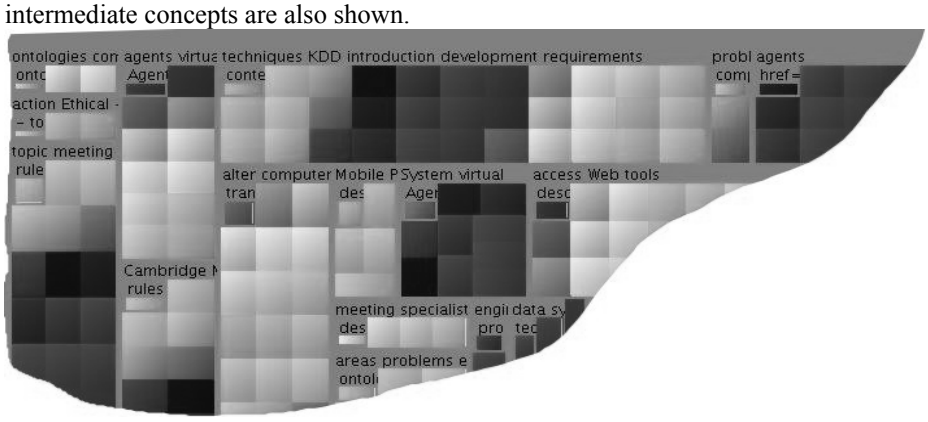


Fig. 2. Part of the Treemap for the Looksmart corpus. Each colored rectangle stands for a concept. Sets of words associated to intermediate concepts only are shown.

4. Using discovered domain model for user modeling

Our method may be applied to build domain models for a hypermedia or website. Once a domain model is learned, user models may be defined as overlay models, sharing the same representation as the domain model. Standard techniques may then be used to learn and update these user models, including Bayesian Nets as proposed in [5, 7].We realized an experiment by running the method on the collection of the pages of the website of a French museum. Fig. 3 shows the resulting domain model as a treemap, with french keywords (paleonthology, sea, press, biodiversity etc).

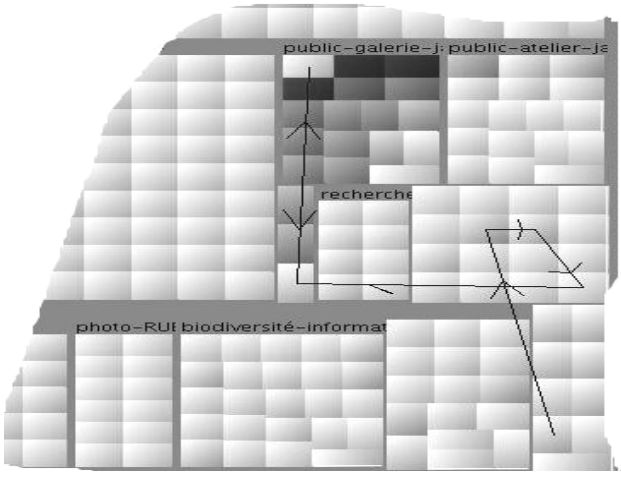


Fig. 3. Part of the Treemap for the website of a French museum where a navigation path has been drawn. The color of rectangles is a function of the similarity between the concepts of the 3 last visited pages and the concepts in the domain model.

To show how such a user model may be used, we have shown a navigation path of a particular user on this treemap, and have defined the colour of a concept (a rectangle) to be a function of the thematic similarity of concepts with the three last pages visited by the user (computed through cosine measure). As may be seen, concepts that are close to the pages recently visited by the user stand close to the current concept. Other information could be visualized. Indeed, treemaps allows redefining easily the rectangles colour and size. Hence, one can browse and investigate a user model by assigning a *knowledge* or *interest* information to the size or colour of the rectangles. This kind of visualization allows having global and synthetic information about a user.

5. Conclusion

We described an approach to automatically learn a domain model from a corpus of hypermedia documents. This approach may be used for instance on the collection of pages of a web site in order to automatically learn a adequate domain model. Based on such a domain model, one can define user models as overlay models. The interest of this approach lies in existing works showing how to learn such user models from logs, and how to perform hypermedia adaptation based on such user models. We also show how efficient visualization techniques such as treemaps may be used to visualize and analyze synthetically both the domain and the user models.

Acknowledgment: The authors would like to thank J.D. Fekete from LRI (Université Paris Sud, France) for helpful discussion about visualization tools and treemaps.

6. References

1. Alfonseca E., Rodriguez P., Modelling users' interests and needs for an adaptive online information system, UM 2003.
2. Brusilovsky P. (1996), Adaptive Hypermedia, an attempt to analyse and generalize, In Multimedia, Hypermedia, and Virtual Reality. Lecture Notes in Computer Science.
3. Brusilovsky P., Adaptive Hypermedia, User Modeling and User-Adapted Interaction, 2001.
4. Cleary C., Bareiss R., 1996, Practical methods for automatically generating typed links. Hypertext '96. Washington DC, USA.
5. Da Silva P., Van Durm V, Duwal E., Olivié H., concepts and documents for adaptive educational hypermedia: a model and a prototype, 2nd workshop on Adaptive Hypertext and Hypermedia, 1998, Pittsburgh, USA.
6. De Bra P., Aerts A., Berden B., De Lange B., Rousseau B., Aha! The adaptive hypermedia architecture, HT'03, United Kingdom.
7. Henze N., Nedjl W., Student modeling in an active learning environment using bayesian networks, UM 1999.
8. Herder E., Van Dijlk B., Personnalized adaptation to device characteristics, International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, 2002.
9. Kavcic A., The role of user models in adaptive hypermedia systems, Electrotechnical Conference, 2000. MELECON 2000.
10. Krishna K., Krishnapuram R., A Clustering Algorithm for Asymmetrically Related Data with Applications to Text Mining. International Conference on Information and Knowledge Management, 2001, Atlanta, Georgia, USA. pp.571-573.
11. Lawrie D., Croft B., Rosenberg A., 2001, Finding Topic Words for Hierarchical Summarization. Proceedings of the 24th annual international ACM SIGIR conference. New Orleans, Louisiana, USA.
12. Njike H. Gallinari P., Leraning generalization/specialization relations between concepts – application for automatic building thematic document hierarchies, RIAO, 2003.
13. Rich E. (1979), User Modeling via Stereotypes, Cognitive Science, 3(4), pp. 329-354.
14. Rojas, Pelechano, Fons Navigational properties and user attributes for modelling adaptive web applications, Engineering the Adaptive Web (EAW'04) Workshop, AH'2004, Eindhoven, The Netherlands.
15. Salton G., Singhal A., Buckley C., Mitra M., 1996, Automatic Text Decomposition Using Text Segments and Text Themes. Hypertext 1996. pp. 53-65
16. Sanderson M., Croft B., 1999, Deriving concept hierarchies from text. In Proceedings ACM SIGIR Conference '99. pp.206-213.

17. Schneiderman B., Tree visualization with tree-maps: 2-d space-filling approach, ACM Transactions on Graphics, Vol. 11, No. 1, January 1992.
18. Schwarzkopf E., An adaptive Web site for the UM2001 conference, Proceedings of the UM2001 Workshop on Machine Learning for User Modeling.
19. Zhu T., Greiner R., Haübl G., Learning a model of a web user's interests, UM'03, pp 65-75.

Activity Modelling using Email and Web Page Classification

Belinda Richards, Judy Kay, Aaron Quigley

School of Information Technologies
University of Sydney
{brichard, judy, aquigley}@it.usyd.edu.au

Abstract. This work explores the modelling of a user's current activity using a single document and a very small collection of classified documents. We describe the WeMAC approach for combining evidence from heterogeneous sources to give a prdict the user's activity. We report evaluation of the WeMAC model using two different document types: emails and web pages; assess its performance on both tiny document sets and larger sets; and assess its performance against a "one bag" approach. We report promising results, with average F1 value of 0.5-0.7.

1 Introduction

A byproduct of the continued growth in the use of computing technologies is that organisations and individuals are generating, gathering and storing data at a rate which is doubling every year [1]. This growth is rapidly outpacing our abilities to handle it. Clearly however, depending on ones' current activities, only parts of one's information set may be relevant at given times. We aim to predict a user's activity based on their currently viewed document, so that applications can deliver relevant information from the user's store of documents. We call our approach WeMAC, and a potential application, the WeMAC Assistant (WeMACA) is illustrated in the scenario below, adapted from [2].

Carla is attending the CHI Conference with presentations related to her research. She and her colleagues have decided to visit different presentations and share their findings at the end of the day. On the first morning, her associate James sends a reminder email detailing the presentations each of them will attend that morning. Carla opens the email using her WeMACA, and on the basis of this, it determines that she is currently "attending the CHI conference". Using this information it displays other documents related to the activity, such as web pages relating to the speakers of the presentations and emails about the conference.

The motivation behind the WeMAC model is to use knowledge from a range of different sources of evidence to a user's activity, from both *implicit (eg. currently viewed document, location)* and *explicit (eg. user input)* sources. Combining both knowledge types, the system reasons about the evidence given to form a conclusion

about the user's current activity, e.g. if the user is currently in her office then from previous monitoring it may conclude that she is engaged in "research".

There has been no work on very tiny data sets of the sizes that we want to study. The most similar work has been in email classification. For example, in predicting the category of a mail item, SwiftFile [3] achieved 50-75% accuracy. It was suggested 50% is of borderline usefulness to users, but 75% is useful. Iems [4] automatically constructed rules to classify mail. Accuracy between 30-67% were achieved on small datasets on one user. On approximately 500 mail items, iems achieved between 41-70% on 2 users' data. Using a bag-of-words and word frequency approach ifile [5] achieves an accuracy score of 86-91% across 4 users. A preliminary study with a small data set for one user ignoring not yet created categories, had an accuracy of 88% on the first 26 items. Some other examples of relevant work involve classifying web pages into pre-defined categories include [6] and [7]. These systems were evaluated on very large datasets, ranging from approximately 4000 [6] to 8000 [7].

We wanted to combine evidence sources from two document types, emails and web pages as shown in Figure 1. The two novel problems that we explore are training on different combinations of document types, and training on tiny datasets. Ideally, the system should be able to classify documents from a variety of different training document types, so that the classification of a new document is possible regardless of the types of documents in the training set. This mirrors the fact that users typically use several different types of documents to support a single activity. This means that it would be valuable to be able to predict relevant documents of different types, based upon classified documents of different types.

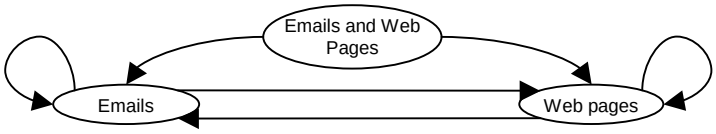


Fig. 1. The combinations of training and testing documents to be explored. The tail of the arrow represents the training document type, and the head refers to the testing document type.

We also wanted to explore the possibility of training on a *tiny* number of documents. This is a difficult task, as most machine learning requires large numbers of examples before it performs well [8]. However, it is an important direction for exploration as it would be valuable to accurately predict a new activity based on just a small number of preclassified documents, so there is quick startup. In the case of supporting user activities, this is particularly important since the relevant document sets may be quite small.

2 WeMAC Model Overview

WeMAC (**We**b page and e**M**ail **A**ccretion for **C**ontext model) deals with two document types: *web pages* and *email*. *Accretion* refers to the method we use to

predict the user's activity. The model is 'for context' as a user's activity is part of their context [9].

The accretion approach to classification involves *resolving* a value from evidence from a number of different sources. The term *resolve* here has a broader meaning than simply the resolution of conflicts between evidence. Rather, to *resolve* means to make a decision based on all available evidence – which may not necessarily be in conflict [10]. An example of the accretion approach is shown in Fig. 2. The values above the lines refer to the *confidence* of the source's prediction, which is the probability that the example belongs to its assigned class. The bold lines are examples of evidence sources considered in the current implementation.

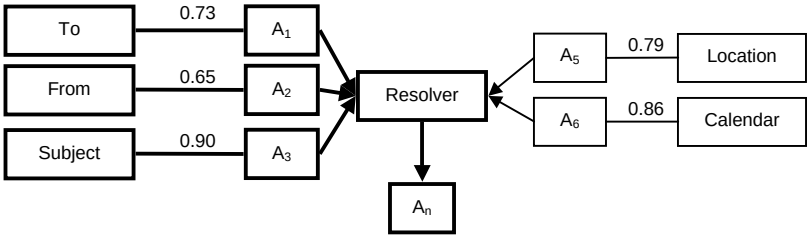


Fig. 2. An example of the accretion based approach to document classification.

We consider three different resolvers:

- *Confidence resolver* uses the evidence source which predicts with the highest confidence to make its final decision.
- *Weighted resolver* is inspired by Multi Attribute Utility Theory [11], and weights the importance of each component's evidence based on its previous performance (we use the average F1 value).
- *Majority resolver* chooses the activity predicted by the most components.

There are 4 main components to the WeMAC architecture: the email and web page preparation, natural language processing, machine learning and accretion components. The first extracts textual information from each email and web page. The components are outlined in Tables 1 and 2. In the situations where emails and web pages are considered in the same classification task, components are collapsed in order to have comparable components which are common to both document types. In initial experiments, we also used feature reduction techniques, specifically stop lists [12] and stemming (the Porter stemmer in the NLTK library). We found, however, that the model performed best on tiny data sets with no feature reduction, which is unsurprising given the small training sets used. Thus our experiments do not use feature reduction.

Table 1. Components of an email

Component	***Description***	***Collapsed Component***
`To`	The information about the sender of the mail item	Heading
`From`	The information about the receiver of the mail item	Heading
`Subject`	The subject line of the mail item	Heading
`Address`	Any URL's within the email. These are identified by those strings which	Address

	start with 'http:/', and are terminated by an end of line or space	
Payload	The main contents of the mail item, after removing any addresses.	Body
All	A combination of the features in all components. This mirrors a traditional bag-of-words approach, which considers all words together. We also use the results from this component as our 'one bag' comparison.	All

TF.IDF scores are then calculated for each word in the training corpus. ARFF [13] files are then created for the training document and each testing document for a suite. Each component of each testing document is then classified using the machine learner, the Naïve Bayes classifier in the Weka Toolkit. We use Naïve Bayes, also used successfully for user-defined email classification (e.g. [5]). The classifications for each component are used as evidence sources for the user's current activity.

Table 2. Components of a web page

Component	*Description*	*Collapsed Component*
H1	The information between 'h1' tags in a web page	Heading
Title	The information between 'title' tags in a web page	Heading
Address	The URL of the web page	Address
Main	Any information in a web page that is not contained in the above components. Note that the actual tag text is ignored (i.e. the text <body> and </body> will not be considered), and text between script and style tags is ignored.	Body
All	As above in **Table X**.	All

Evaluation

In order to test the accuracy of the WeMAC model on small datasets on each of the combinations of document types shown in Fig. 1, we performed cross validation experiments and used a sliding window. With our tiny data sets, we ignored temporality, although it may aid in the classification [14].

Each *static test suite* (set of cross validation experiments) contained the same number of documents from each activity, and if appropriate, each document type. To ensure that equal numbers of documents from each activity were used in the training set, we left out a testing document from each activity and if appropriate, each document type for each experiment. This avoided bias from a larger number of training examples in one class.

A sliding window approach ensured the results were stable across the corpus and not specific to the particular combination of documents chosen for a test suite. Each window contained at least half the documents from the previous window.

To evaluate the performance of the WeMAC model, we varied the resolver, the combination of training and testing document types, and the training set size, and compared the WeMAC accretion approach with a 'one bag' approach (which did not split the document into separate components).

The private nature of email means there is no standard evaluation corpus. Nor has there been work in predicting activities. Thus, we collected data specifically for this classification task.

We report the evaluation on 5 users' data. This small number is comparable to other published work on a user-defined email classification (e.g. [3, 5, 15]). Each user selected their activities from a predefined list: this was done to encourage users to classify their documents according to activity rather than a traditional classification scheme. The list was created based on interviews of research staff and students about activities they do in association with emails and web pages. The activities were: Teaching, Subject Work, Admin, Conference, Seminar, Research and Project.

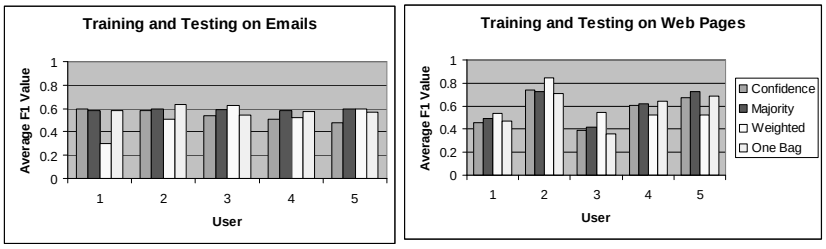


Fig. 3. Comparison of resolvers and the 'one bag' approach for emails and web pages.

All users collected at least 10 emails and 10 web pages for each activity that they selected as applicable to them. Most users found it difficult to find more than 10 web pages per activity and commented that for the activities given, they tended to view only a small set of web pages. However, 2 of the 5 users were able to collect at least 50 emails per activity. This allowed us to compare the results obtained on small data sets with results from training on a somewhat larger number of documents. Four of the five users selected 5 activities, and 1 selected 6.

We performed a careful qualitative analysis of all results obtained to ensure that we fully understood the patterns of performance. The performance value quoted is the average F1 value. This also maps quite closely to the accuracy, as is often reported in the literature. We achieved results comparable to those described above, with accuracy 0.5 to 0.7 for small datasets.

The first evaluation task involved the small data sets, 10 emails and 10 web pages per activity, with 4 documents per static test suite, 3 windows, a total of 12 experiments. For all users, best performance was with the same document type for training and testing, as these documents shared the most textual information.

The pattern of resolver performance tended to differ across users for these experiments, as illustrated in Fig. 3. The Confidence and Majority resolvers tended to perform consistently well across all users. The performance of the Weighted resolver when training and testing on web pages generally depended on the performance of the Address component. If the address component performed consistently well across most classes, as for User 2 and User 3 whose web pages for each activity generally came from similar domains, the Weighted resolver could to use this to its advantage. The Address component was weighted highly by the Weighted resolver due to its good performance, and as it consistently performed well, the Weighted resolver also performed consistently well for these users, and better than the other two resolvers.

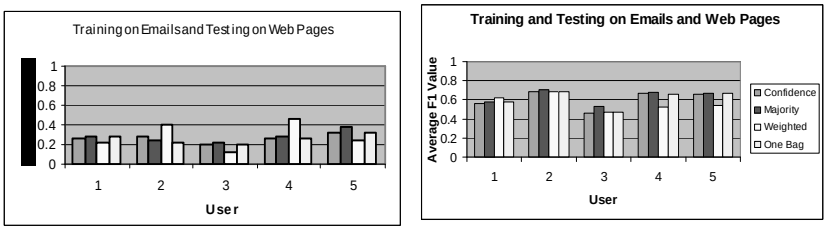


Fig. 4. Comparing the performance of the WeMAC resolvers and the one bag approach.

However, if the Address component did not perform well, then the performance of the Weighted resolver did not tend to perform well, as the performance of the components on each activity tended to change over the corpus.

The effect of unstable component performance on the Weighted resolver was particularly evident for User 1 when training and testing on emails. For this user, the pattern of performance was so variable that weighting each component's input to the decision based on its previous performance caused the Weighted resolver to make more incorrect predictions and perform particularly poorly.

Notably the one bag approach was quite similar for all users. This suggests that both are both appropriate methods for email classification on small datasets.

WeMAC performed poorly when training on one document type and testing on the other. The components performed poorly, as did the resolvers, as shown in Fig. 4. Even so, all resolvers struggled to reach an average F1 value of more than 0.4.

We then explored training and testing on both document types. The performance values were typically similar to the average of the results when training and testing on the same document type. This is the case for both the components and resolvers, and the resolver performance can be seen in Fig. 4. Generally, WeMAC and one bag approach performed almost exactly equally in this case.

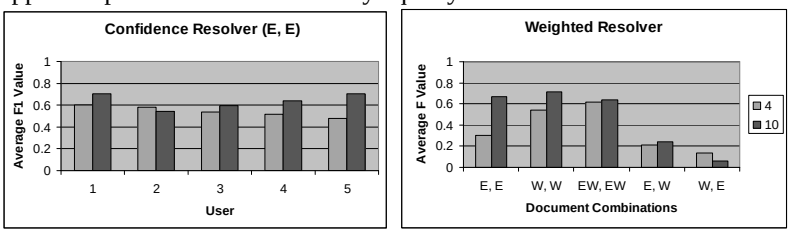


Fig. 5. Performance change for Confidence resolver over all users and Weighted resolver for User 1 (key: e.g. (E,E) = train on emails, test on emails).

To ensure that the WeMAC model's performance on small datasets was not due to features unique to small datasets, we considered 2 additional sets of results. The first used 10 documents per activity and document type for each user over 1 window, a total of 10 sets of experiments. We compare the results obtained on this set with the above results. The second used the larger datasets provided by User 1 and User 4. Overall, as expected, all resolvers and components generally improved or remained

the same as the number of documents increased, as illustrated in Fig. 5, which is a representative example of the performance change.

For some users, the improvement in performance for a particular resolver was quite marked, as was the case for the Weighted resolver for Users 1 and 4. User 1's results are shown in Fig. 5. The substantial performance increase was due to the improvement in performance for the Subject and Payload. Generally, performance increased by approximately 0.1 as the size of the training set increased from 4 to 10. For the even larger data sets, we used the number of documents per activity and the number of windows shown in Table 3.

Table 3. Details the documents used for each experiment

Number of documents per activity	***Number of Windows Per Experiment***	
	User 1 **Total # documents per activity: *100***	**User 4** **Total # documents per activity: *50***
4	30	20
10	15	8
25	5	3
50	3	1

The results for these evaluations are shown in Fig 6. All resolvers clearly improve their performance as the number of training documents increases. The Majority resolver consistently outperformed the other resolvers for both users, followed by the Confidence and then Weighted resolvers. The Weighted resolver performed least well for User 1, and approximately equal to the Confidence resolver for User 4. For the Weighted resolver to perform well, it required a consistent pattern in performance for the components for the classes they predicted well. The components for User 1 did not exhibit this behaviour, as each component's performance in predicting each activity changed over the document set. However, the components for User 4 did eventually tend to exhibit this behaviour.

The importance of resolver choice for different users is especially evident in Fig. 6. For User 1, the Majority and Confidence resolvers performed almost 0.2 above the Weighted resolver. However, all resolvers performed similarly for User 4's data. This illustrates the fact that different methods of combining evidence may be more appropriate for different users.

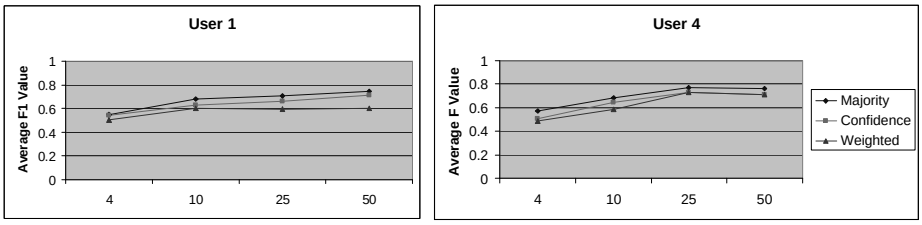


Fig. 6. Rate of learning for the different resolvers on Users' 1 and 4 data

The results on these larger data sets demonstrate that WeMAC can also perform well not only on larger training sets. Thus, we have shown that the WeMAC model is

a feasible general approach to heterogeneous document classification, here for emails and web pages.

Future Work and Conclusion

The development of the WeMAC model grew from the goal of predicting a user's activity from a *tiny number of training documents* and *heterogeneous document types*. We found that even training on just 12 to 15 documents, we were able to achieve average F1 around 0.6. As expected, the performance improved steadily as the training set size increased. As WeMAC was intended to resolve evidence from a number of different sources, it would be interesting to determine whether contextual evidence sources such as location or time improved the performance of the system.

References

1. Middlemiss, J., CIO Challenge: Search Engines (available at *http://www.wallstreetandtech.com/showArticle.jhtml?articleID=21401585)*, in Wall Street and Technology - online. 2004.
2. Dey, A.K., D. Salber, and G.D. Abowd, A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction., 2001(16): p. 97-166.
3. Segal, R.B. and J.O. Kephart. Incremental Learning in SwiftFile. in Proceedings of the 17th International Conference on Machine Learning. 2000.
4. Crawford, E., J. Kay, and E. McCreath. IEMS - The Intelligent Email Sorter. in IEMS - The Intelligent Email Sorter. 2002.
5. Rennie, J.D.M. ifile: An Application of Machine Learning to E-Mail Filtering. in Proceedings of the KDD-2000 Workshop on TextMining. 2000.
6. Asirvatham, A.P. and K.K. Ravi, Web Page Classification based on Document Structure. 2001.
7. Riboni, D. Feature Selection for Web Page Classification. in Proceedings of the Workshop EURASIA-ICT 2002. 2002.
8. Webb, G.I., M.J. Pazzani, and D. Billsus, Machine Learning for User Modeling. User Modeling and User-Adapted Interaction, 2001(11): p. 19-29.
9. Dey, A.K. and G.D. Abowd, Towards a Better Understanding of Context and Context-Awareness. 1999, Georgia Institute of Technology.
10. Kay, J., R.J. Kummerfeld, and P. Lauder. Personis: a server for user models. in Proceedings of Adaptive Hypertext 2002. 2002: Springer.
11. Winterfield, D.V. and W. Edwards, Decision Analysis and Behavioural Research. 1986, Cambridge, England: Cambridge University Press.
12. Available from *http://www.searchengineworld.com/spy/stopwords.htm*.
13. *http://www.cs.waikato.ac.nz/~ml/weka/arff.html*.
14. McCreath, E. and J. Kay. Iems: Helping Users Manage Email. in User Modelling. 2003.
15. Crawford, E., J. Kay, and E. McCreath. Automatic Induction of Rules for E-Mail Classification. in Proceedings of ADCS'2001, Australian Document Computing Symposium. 2001.