International Journal of Security and its Applications
Vol. 2, No. 1, January, 2008

Mining User Models for Effective Adaptation of Context-aware Applications

Shiu Lun Tsang[1]
Distributed Systems Group
Trinity College Dublin
ShiuLun.Tsang@cs.tcd.ie

Siobhán Clarke
Distributed Systems Group
Trinity College Dublin
Siobhan.Clarke@cs.tcd.ie

Abstract

Current context-aware adaptation techniques are limited in their support for user personalisation. Complex codebases, a reliance on developer modification and an inability to automatically learn from user interactions hinder their use for tailoring behaviour to individuals. To address these problems we have devised a personalised, dynamic, run-time approach to adaptation. The approach provides techniques for selecting the relevant information from a user's behaviour history, for mining usage patterns, and for generating, prioritising, and selecting adaptation behaviour. Our evaluation study shows that the proposed mining approach is more accurate than rule-based and neural network methods when compared to actual user choices.

1. Introduction

Advances in mobile computing and sensor devices have led to the emergence of a new class of applications that are context-aware. While there are existing infrastructures that aid the development of context-aware applications, researchers have expressed concerns about their limited support for user issues [2, 4, 6, 14].

Existing context-aware applications commonly adapt to changing context using rules or Machine Learning (ML) techniques. Some also adapt based on historical information. However, each approach has their limitations when behaviour is to be tailored to individuals. Rules must be defined at development time, which means that developers must try to identify "all" possible situations and corresponding actions. Rules are difficult to modify, maintain and scale [21] and explicit developer modification is required to cater for any new knowledge gained from a user's actions. Probabilistic ML methods such as Bayesian networks are constructed to model context dependencies at development time and so suffer similar maintenance and scalability problems to rule-based approaches. Calculating the necessary probabilities is also a tedious, ongoing challenge [23]. The use of neural networks (NN) in context-aware applications has been limited due to their heavy processing requirements and complex black box nature. Existing techniques that store an individual's history are used in conjunction with probabilistic methods and so suffer from the aforementioned problems. Approaches that represent the history of a community of users suffer inaccuracies when there is a large disparity in the preferences of different users. Their accuracy and performance are

[1] Supported by the Irish Research Council for Science, Engineering and Technology (IRCSET)

also affected when historical information is made up of a large number of information types with many possible values. All of the above approaches are also inflexible when adapting to a set of different user goals. Supporting such functionality would require the creation and transition between multiple rule sets, policies, probability tables or networks.

In this paper we propose a personalised, dynamic, run-time approach to adaptation. We contribute to existing work by providing a flexible approach to adaptation that overcomes inaccurate behaviour caused by the disparity in the preferences of users in a group. The approach consists of techniques for: 1) automatically adapting to different (changing) user goals or tasks, 2) storing historical user information which may contain different information types with many possible values, 3) filtering the relevant information from a user's behaviour history, 4) mining usage patterns from a large set of historical information, 5) generating, prioritising, and selecting adaptation behaviour. Tailored actions are achieved by analysing the relevant details of an individual's history. Run-time execution enables the application to dynamically adapt to a user's current goals, requirements, preferences, and context. Online learning is facilitated through the addition of new user interactions and accurate adaptation decisions are achieved even from a small set of historical behaviours. Our evaluation study shows that our approach is more accurate than rule-based and NN methods when compared to actual user choices.

The remaining sections are organised as follows. Section 2 describes the proposed mining approach. Section 3 describes our evaluation study and results. Section 4 documents our related work and Section 5 summarises our approach and findings.

2. A mining approach to adaptation

This section describes our proposed mining approach for personalised adaptation in context-aware applications.

2.1. Overview

The mining approach is based on the premise that users exhibit certain habits in their behaviour that recur when faced with similar situations. An analysis of past user behaviour will therefore provide useful knowledge for predicting current and future user desires and actions. Figure 1 illustrates the steps of the approach.

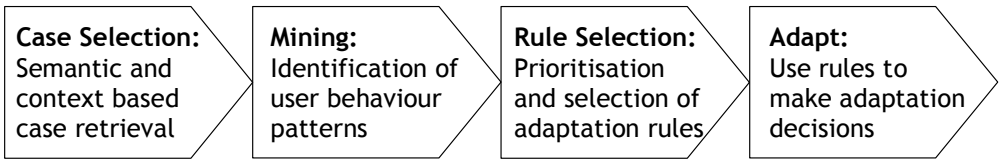

Figure 1: A mining approach to adaptation

A semantic and context based selection method has been devised to ensure that only information relevant to the current decision is selected. Filtering information in this manner provides a flexible technique that enables adaptation to adjust to changing user goals, requirements, preferences, and context. A user model [16, 6] represents user-related information such as past context and behaviour. Our study has shown that individuals use different information with varying levels of importance in adaptation decisions. Consequently, our user model stores information about a single individual and adaptation decisions are based on that individual's information. User behaviour

patterns relevant to the current situation are dynamically mined from retrieved information and adaptation behaviour is prioritised and selected based on the strength of identified patterns. This ensures that actions are individually tailored and that the correct level of importance is assigned to each pattern. Also, the automatic addition of new behaviour and context acquired from user interaction facilitates online learning, enabling the application to evolve with any changing user requirements or preferences.

Our work is conducted as part of the Hermes framework [10, 12]. Hermes is a trails-based framework that provides developer support for common context-aware functions organised in a layered structure. Our work will provide personalisation functions to ensure that the relevant set of information is retrieved from the context management layer, this information is mined for additional knowledge, and mined knowledge is passed to the application layer for adaptation.

2.2. User modelling

Our study has shown that individuals use different information with varying levels of importance in adaptation decisions. Consequently, we have devised a user model that stores information associated with a single user. Adaptation decisions are made based on the information in individual user models. This notion is supported by Kobsa who defines a user model as "*a collection of information and assumptions about individual users [...] which are needed in the adaptation process of systems to individual actions of users*" [15].

Our user model is designed to store user-related information such as user preferences, behaviour history and any relevant contextual information. The model supports data acquisition through a variety of means: from sensors, from user-application interaction, or from any other sources available to the application. Data is stored in the form of cases and each case consists of a set of context tuples. A collection of cases represents user behaviour over time. An example user model case for a restaurant application is as follows:

{preference = chinese, cuisine = chinese, cost = cheap, meal = lunch, day = wednesday}

The *preference* variable represents the user's cuisine preference, *cuisine* and *cost* variables represent properties associated with the chosen behaviour (restaurant in this case), and *meal* and *day* variables represent context information at the time.

2.3. Relevant case selection

In a context-aware environment, a vast amount of information is readily available to an application either from surrounding sensors, networks or users. Large amounts of data can mean heavy processing and make it more difficult to identify useful knowledge [9, 17].

To ensure that only the relevant information is chosen for making adaptation decisions, we have devised a semantic and context based selection technique. We begin by selecting user model cases that are semantically similar to the current user problem. For example, when recommending a restaurant, only cases semantically similar to restaurant visits are selected. Semantic similarity is defined by the applications object hierarchy structure. Take for example a *dining* object with *restaurant* and *café* children objects. When determining restaurants to recommend, we retrieve all cases associated with restaurants, but we can also select cases related to cafés i.e., users who like comfortable restaurants will prefer comfortable cafés. The approach is flexible and can be adjusted to traverse any number of levels or particular branches of the hierarchy. This flexibility also enables applications to

easily adjust to different user goals or requirements. For example, traversing the *shopping* branch of the hierarchy will support adaptation related to shopping tasks and similarly for other object types. Supporting such functionality with existing techniques would require the creation and transition between multiple rule sets, policies, probability tables or networks.

The second step is based on context. From the chosen set of semantically similar cases, we select only those that are similar in context to the current user situation. If the current context, for example, is dinner time on a weekend, then only cases that match this situation are selected, eliminating cases associated with lunch time or weekdays. This again ensures that adaptation decisions are based on the most relevant information, enabling accurate behaviour that is tailored to an individual.

2.4. Mining behaviour patterns

Users exhibit certain habits in their behaviour that recur when faced with similar decisions or situation. An analysis of relevant past user behaviour will therefore provide useful knowledge for predicting user desires and actions. A run time approach ensures that this analysis is performed to reflect the current user goal and context.

Association mining [1] is used as our method for evaluating the behaviour history of a user. In particular the *Apriori* algorithm [1] has been adopted. The function of association mining is the identification of relationships or correlations between items in a data set [9, 17]. Outputs from mining are a set of rules in the form A=>B. For example "*people who like French cuisine also like expensive wine"* may be written *"cuisine=French => wine=expensive*". Mining enables an effective analysis of historical information even if a large number of different information types and values exist. As rules are generated from relevant user model cases, they will reflect the past requirements of an individual, providing a reliable basis from which accurate adaptation decisions are made.

2.5. Adaptation rule selection

While semantic and context based information selection ensures that only relevant user model cases are selected, mining can still yield a large number of potentially conflicting rules. The importance of rules will also vary, similar to how users assign different levels of importance to information when making decisions. Certain rules should therefore carry more weight in the final adaptation decision.

To calculate the weight a rule carries and to ensure that the most appropriate rules are chosen, we rank generated rules according to their *strength*. Association mining provides two measures, *confidence* and *support*, for comparing rules [9]. *Strong* rules have high confidence and high support. The stronger the rule, the more accurate and recurring it is. Rule ranking and selection will therefore favour stronger rules. The highest ranking rule associated with each output type specifies how best to adapt. Take for example the following:

Rule 1: My birthday => Expensive restaurant (confidence 0.8, support 5)

Rule 2: My birthday => Cheap restaurant (confidence 0.4, support 5)

Rule 3: My birthday => Italian cuisine (confidence 0.8, support 2)

Rules 1 and 2 are conflicting, but because Rule 1 is stronger (higher confidence), it will be chosen. Both Rule 1 and Rule 3 will be used in decision making, but Rule 1 will carry more

weight (higher support) i.e., given that a restaurant is not both expensive and Italian, decisions will favour expensive restaurants over Italian ones.

3. Evaluation study and results

This section describes the set-up of our study and the results that were acquired. It also describes how the validity threats posed to our study are addressed.

3.1. Study set-up and implementations

The aim of the study is to determine the effectiveness of a mining approach for personalised adaptation in context-aware applications. We assess its accuracy in predicting user decisions as compared to a rule-based and a NN approach. A restaurant recommender scenario was chosen for our analysis. The study consisted of the following steps: requirements gathering, application development, and evaluation.

Table 1: Input and output variables

Inputs	Meal, Day, Company, Transport, Occasion, Time Constraint
Outputs	Cuisine, Cost, Food Quality, Service/Atmosphere, Distance

A two-part questionnaire was constructed for requirements gathering. The first part captures general restaurant preference information such as cuisine preference and average expenditure per restaurant visit. The second part describes ten different cases. Each case presents a different context situation and details of the type of restaurant the participant would choose in that context are acquired. Table 1 shows the information (inputs) used to construct the different cases and restaurant properties (outputs) that were acquired. The requirements for ten different participants were gathered for our study.

Using the gathered requirements, applications were subsequently implemented. For each of the participants in the requirements gathering stage, three versions of the application were developed, one under each approach. This enables the comparison of the three approaches against actual user choices. Ten developers were used in the study for rule-based development. Each one developed a rule-based version of the application for one of the participants. Rules were implemented in Java, in the form of conditional statements. The NN approach uses case responses gathered in the previous stage as training data. The *EasyNN-plus* tool [19] was used to generate our NN, which consisted of one hidden layer of six nodes. Cases from the previous stage also function as user model cases for the mining approach. The Apriori algorithm implemented in *Weka* [25] was integrated with our approach for mining behaviour patterns.

Our evaluation assesses the accuracy of each approach by comparing their outputs against participant outputs. We compare the outputs for 15 test cases: ten cases presented in the requirements gathering questionnaire, and five other (different) cases that represent new situations that may be encountered in the future. The responses to these new cases were gathered from participants after application development. Developers using the rule-based approach were also questioned post implementation. The *Likert* (five) scale system is used to capture each developer's opinion about a set of statements regarding the difficulties of rule-based system development.

3.2. Findings

This section provides the results of our study. We adopt the following metrics:

- Match Rate: the percentage of system outputs that correspond to the users output.
- Error Rate: the percentage of system outputs that do not correspond to the users output.
- Missing Rate: the percentage of system outputs that do not contain an output value.

Missing values occur in rule-based systems due to the complex structure of rules. Complicated nested statements means that certain blocks of desired code are not reached. Missing values also occur in our mining approach. Depending on the set of user model cases that are selected, certain variables may not be present in the generated rules. NN are designed to always provide the specified output and therefore will never result in missing values. The following example demonstrates how the metrics are calculated:

User choice: {cuisine = japanese, cost = expensive, food quality = excellent, Service = 4 star, distance = far}

System output: {cuisine = japanese, cost = expensive, food quality = good, Service = 4 star, distance = ?}

The output of this sample system would match three of the users outputs, giving a Match Rate of 60%, it contains one error, meaning an Error Rate of 20%, and one missing output (denoted by ?), resulting in a Missing Rate of 20%. Our evaluation performs the same calculation for each case and for each approach. The results are summarised in Table 2.

Table 2: Match rate, error rate, missing rate

Rates	Questionnaire Cases			New Cases		
	Mining	Rules	NN	Mining	Rules	NN
Match	73%	48%	54%	57%	46%	36%
Error	16%	28%	46%	33%	30%	64%
Missing	11%	24%	-	10%	24%	-

3.2.1. Mining v rule-based: Table 2 shows the mining approach to outperform the rule-based approach in most categories. An exception to the trend is seen in the Error Rate metric for new cases; however this is offset by rules having a significantly higher value for the Missing Rate metric in the same category. Table 3 also shows that mining outperforms rules in a higher number of cases.

The lower accuracy of a rule-based approach is attributed to the difficulty developers' face when constructing complex rule-bases. Table 4 shows the results of a post implementation questionnaire given to rule-based system developers. The Likert (five) scale system is used, where 1 corresponds to *Strongly Disagree* and 5 to *Strongly Agree*.

Many current context-aware frameworks support rule-based adaptation (also referred to as event-condition-action model [11]) [3, 8, 13, 22, 5]. However, during our investigation we noted a number of additional limitations with this approach: 1) developers are responsible for explicitly writing the set of required rules, usually to satisfy the requirements of a

stereotypical user. It is difficult and time-consuming to set aside developers to write rules tailored to individuals, 2) rules are defined at development time, thus requiring the developer to specify the situations a user may encounter and the corresponding action to take in each situation. Identifying all possible situation-action permutations a priori is a difficult, if not an impossible, task, especially when there are a large number of variables to consider, 3) a rules approach will not learn or exploit new knowledge gained from a users interaction with the application. Explicit developer modification is required to cater for any new knowledge, 4) rules are generally thought to be difficult to maintain and modify, error prone, and difficult to scale [21].

Table 3: Mining v rule-based and neural network approaches

% Cases with Greater Accuracy	Questionnaire Cases		New Cases	
Mining v Rules	Mining	Rules	Mining	Rules
	74%	9%	48%	19%
Mining v NN	Mining	NN	Mining	NN
	63%	19%	52%	19%

The addition of new information as the user interacts with the system and the dynamic run-time generation of adaptation behaviour means that the mining approach does not suffer similar problems. Adding to the user model enables the application to automatically evolve to any changing user requirements or habits.

Table 4: Rule-based development opinions

Statements	**Likert Rating**
It proved difficult to identify the permutations of rules required	4
The system was overall difficult to design and implement	3.7
The system will cater for new cases	1.5
The system captures the requirements of the user	3.4
The system will produce accurate outputs	2.8

3.2.2. Mining v neural networks: Table 2 shows mining to be more accurate than NN. NN do not output missing values; however this is offset by a significantly higher Error Rate value. Table 3 also shows that mining outperforms NN in a higher number of cases.

During our study, a number of different ML techniques were investigated. We noted several limitations regarding their appropriateness for developing personalised context-aware systems. Probabilistic techniques such as Bayesian networks are constructed to model information dependencies at development time and so suffer similar maintenance and scalability problems to rule-based approaches. They are predominantly focused on addressing uncertainty (as opposed to adaptation) and issues related to the calculation of necessary probabilities remains a tedious, ongoing challenge [23].

A number of additional problems were identified for NN [20, 23]: 1) they are backpropagational networks, which require large amounts of processing power and time to train, 2) a large number of representative cases are required to train the network to a point

where its error rate is sufficiently low. Cases should also be balanced to avoid bias to a particular set of case outcomes, 3) inputs are required to be converted to numeric form. This may cause false ordering of nominal values which reduces the accuracy of the network. Networks also tend not to perform well with nominal values that have a large number of possible values, 4) they are prone to problems of over-learning and over-fitting, 5) their black box nature means that they provide no indication as to why particular outputs were produced. Many studies have cited the need for users to be able to scrutinise systems to determine why certain decisions were made [2, 4, 7].

The mining approach to adaptation overcomes these problems. The methods devised for case and rule selection reduce the processing required for mining rules. The results show that even for a small amount of cases, mining provides accurate results, and because explicit adaptation rules are generated, users can easily query the application as to why particular outcomes were produced.

3.3. Case study validity

This section describes how we addressed the validity threats posed to our evaluation.

A biased comparison of the adaptation approach threatens the internal validity of our study. To address this we compare the application outputs of each approach in a straightforward manner: we compare the same outputs of each approach with actual user outputs.

A possible threat to construct validity is that we chose developers that are of insufficient standard. To address this we chose developers of M.Sc., Ph.D., or Postdoc. level in a computer/engineering related discipline. All are also researchers in the pervasive computing domain.

The small sizes of our systems threaten the external validity of the results of our case study. However, we have determined that the study results will remain true in larger systems. Larger systems contain more variables and values, which will only increase the difficulty in identifying the necessary adaptation rules. More variables will also increase the size of a rule-base leading to added maintenance difficulties. Problems with non-numeric data, small sets of training data and its black box nature will always be present in NN, irrespective of system size.

4. Related work

Recent research in context-aware computing has highlighted the growing need to better integrate user-related properties, including user preferences, knowledge, and behaviour, into application development. Jameson [14] and Bradley [4] both provide models for demonstrating how user properties can be integrated with other context-related functionality. However, both models only consider the current context of the user. Several studies have looked at the use of historic context and behaviour for aiding adaptation. Byun and Cheverst [7] investigate how context history could aid a system in exhibiting intelligent and proactive behaviour. Work however is predominantly focused on addressing the effect of uncertainty in historic information and their use of probabilistic ML techniques means that they suffer similar limitations as those highlighted in Section 3.2. Mohomed et. al [18] investigate the use of a community's past usage semantics for improving adaptation. Their work shows that accurate results can be achieved when there is a high level of similarity between different users. However, we argue that the accuracy of adaptation behaviour diminishes in relation to the dissimilarity within a group, to a point at which utilising the history of an individual would provide more accurate results.

Singh et. al [24] investigate the possibility of using data mining techniques in context-aware systems. They have shown that using context can improve the accuracy and efficacy of data mining results. However, their work does not include any techniques for selecting context (to use in mining) nor for utilising mining results for adaptation. In addition, research remains at an early proof of concept stage and no empirical assessment of the approach (as a standalone technique or as compared with other techniques) has been performed.

5. Conclusions

The paper describes research on a mining approach to adaptation. We describe the various steps and concepts that make up the approach and the study performed for its evaluation. Our findings show that it is more accurate in matching user decisions, provides less errors and missing values, and is more accurate in a higher number of cases, than its rule-based and NN equivalents. It also overcomes additional issues associated with the latter two techniques, making it a promising and effective approach for developing personalised context-aware applications.

6. References

[1] R. Agrawal, H. Mannila, R. Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. Advances in knowledge discovery and data mining, pp.307-328, 1996.

[2] V. Bellotti and K. Edwards. Intelligibility and accountability: Human considerations in context-aware systems. Journal of Human-Computer Interaction, 16:193-212, 2001.

[3] G. Biegel and V. Cahill. A framework for developing mobile, context-aware applications. In IEEE International Conference on Pervasive Computing and Communications, p.361, USA, 2004.

[4] N.A. Bradley and M. D. Dunlop. Toward a multidisciplinary model of context to support context-aware computing. Journal of Human-Computer Interaction, 20(4): pp.403-446, 2005.

[5] P.J. Brown. The stick-e document: A framework for creating context aware applications. In Electronic Publishing, vol.8, pp.259-272, 1996.

[6] H.E. Byun and K. Cheverst. Exploiting user models and context awareness to support personal daily activities. In Workshop on User Modelling for Context-Aware Applications, Germany, 2001.

[7] H.E. Byun and K. Cheverst. Harnessing context to support proactive behaviors. In AI in Mobile Systems, France, 2002.

[8] D. Chen, A. Schmidt, and H.W. Gellesen. An architecture for multi-sensor fusion in mobile environments. In International Conference on Information Fusion, USA, 1999.

[9] M.S. Chen, J. Han, and P.S. Yu. Data mining: An overview from a database perspective. IEEE Transactions on Knowledge and Data Engineering, 8(6):866-883, 1996.

[10] S. Clarke and C. Driver. Context-aware trails. IEEE Computer, 37(8):97-99, 2004.

[11] D.L. De Ipia. An eca rule-matching service for simpler development of reactive applications. Middleware, 2(7), 2001.

[12] C. Driver, E. Linehan, M. Spence, S.L. Tsang, L. Chan, and S. Clarke. Facilitating dynamic schedules for healthcare professionals. In Conference on Pervasive Computing Technologies for Healthcare, Austria, 2006.

[13] K. Henricksen and J. Indulska. A software engineering framework for context-aware pervasive computing. In IEEE International Conference on Pervasive Computing and Communications, p.77, USA, 2004.

[14] A. Jameson. Modeling both the context and the user. Personal Technologies, 5(1):29-33, 2001.

[15] A. Kobsa. Supporting user interfaces for all through user modeling. In International Conference on Human-Computer Interaction, pp.155-157, Japan, 1995.

[16] A. Kobsa. Generic user modeling systems. User Modeling and User-Adapted Interaction, 11(1-2):49-63, 2001.

[17] J.N. Kok and W.A. Kosters. Natural data mining techniques. Current Trends in Theoretical Computer Science, pp.603-613, 2001.

[18] I. Mohomed , J. Chengming Cai , S. Chavoshi , E. de Lara, Context-aware interactive content adaptation, In 4th International Conference on Mobile systems, Applications and Services, pp.42-55, Sweden, 2006

[19] Online: EasyNN-plus. www.easynn.com, 2007.

[20] Online: www.statsoft.com/textbook/stneunet.html, 2007.

[21] J. Prentzas and I. Hatzilygeroudis. Categorizing approaches combining rule-based and case-based reasoning. Expert Systems, 24:97-122(26), 2007.

[22] A. Ranganathan and R.H. Campbell. A middleware for context-aware agents in ubiquitous computing environments. In International Middleware Conference, pp.143-161, Brazil, 2003.

[23] S. Russell and P. Norvig. Artificial Intelligence, A Modern Approach. Prentice Hall, 2nd edition, 2003.

[24] S. Singh, P. Vajirkar, and Y. Lee. Context-based data mining using ontologies. In 22nd International Conference on Conceptual Modeling, pp.405-418, USA, 2003.

[25] I.H. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques, 2nd edition, 2005.

Authors

Shiu Lun Tsang is a Ph.D. student in the Distributed Systems Group at Trinity College Dublin. His current research interests include context-aware adaptation, personalisation, and user modelling. He holds a M.Sc. in Computer Science from Trinity College Dublin. Contact him at Trinity College Dublin, College Green, Dublin 2, Ireland; ShiuLun.Tsang@cs.tcd.ie; www.cs.tcd.ie/ShiuLun.Tsang/.

Siobhán Clarke leads the Distributed Systems Group at Trinity College Dublin. Her research interests include design and programming models for mobile, context-aware systems and aspect-oriented software development. She received her PhD in computer science from Dublin City University. She's a member of the IEEE.Contact her at Trinity Collge Dublin, College Green, Dublin 2, Ireland; Siobhan.Clarke@cs.tcd.ie; www.cs.tcd.ie/Siobhan.Clarke/.