The Challenge of Preparational Behaviours in Preference Learning for Ubiquitous Systems

Sarah Gallacher, Eliza Papadopoulou, Nick K. Taylor, M. Howard Williams
School of Mathematical and Computer Sciences, Heriot-Watt University
Edinburgh, Scotland
{S.Gallacher, E.Papadopoulou, N.K.Taylor, M.H.Williams}@hw.ac.uk

***Abstract*—Many ubiquitous computing systems employ intelligent components that learn how to adapt the user's environment on their behalf, by observing how the user has adapted such environments in the past. Such components employ monitoring and machine learning techniques to capture human behaviours and process them to extract adaptation rules (or user preferences). However, learning preferences from observations of behaviour introduces challenges that are not so compounded in other machine learning problem domains. One key issue is preparational behaviours (or pre-actions) which current preference learning solutions can struggle to handle. This paper uses pre-actions as an example discussion point and raises the question of whether preference learning solutions should take advantage of temporal data from real-world environments to improve performance. The key contribution of this paper is the introduction and analysis of a novel machine learning technique (the DIANNE) that utilises temporal data to handle user behaviour anomalies such as pre-actions.**

Behaviour; Context; Learning; Personalisation; Preferences; Pervasive; Ubiquitous

I. Introduction

The goal of creating ubiquitous/pervasive computing environments has been explored since the early 80's. However, it is only in recent years that advances in technology have begun to impact on everyday environments, transforming them to reflect the integrated and technology rich environments such as those envisaged by Weiser [1]. For example, figures from Kantar Worldpanel ComTech [2] state that in February of 2012 over half the UK population owned a smart phone, a device that is becoming increasingly powerful and integral to our everyday lives. Additionally, with the continued rise of social networking, online gaming, movie downloads and internet television our homes are becoming ever more connected and technology rich. Other trends include smart energy, the internet of things and smart spaces for assisted living, all of which aim to increase the integration of technology into our daily lives.

The challenges of optimising such technology rich environments to meet the current needs of inhabitants have been considered in ubiquitous computing research for many years. The result is that many current ubiquitous computing systems employ intelligent components that can learn how to autonomously and proactively adapt the user's environment on their behalf, by observing how the user has adapted such environments in the past. This process is often referred to as *personalisation*.

A key element of this process is the internal knowledge that accumulates over time via the learning mechanisms. This knowledge dictates what proactive and autonomous behaviours should be performed to optimise an environment for an individual. It can take many forms but a common format is a set of context-dependent rules, often referred to as *user preferences*.

The typical approach adopted by many personalisation systems to learn user preferences is as follows: a behaviour monitoring element monitors user behaviour within the environment, storing it with current user context information to build up a training dataset. Once enough training data is collected, machine learning algorithms are applied to extract user preferences indicating what behaviours the user performs in a given context. An application element can then monitor the user's context in the future and apply the appropriate behaviour(s) as indicated in the preferences.

However, the pre-emptive and cognitive abilities of the human users being monitored often means that the assumptions made in the process described above do not hold. Essentially, there is not always a direct mapping between a behaviour and the context in which that behaviour is performed. It has been observed that sometimes users perform behaviours in one context to prepare for entry into another context. We have termed these preparational behaviours *pre-actions* and in this paper we describe how pre-actions can prove problematic for current preference learning solutions. It is proposed that such solutions could benefit from the exploitation of temporal data from the user's real-world environment, as done by the DIANNE which is introduced in this paper as a novel preference learning solution.

Section 2 gives a brief background to personalisation in ubiquitous environments describing how several other systems learn to proactively adapt environments based on the observation of past user behaviours. Section 3 discusses pre-actions in more detail and raises the question of whether temporal data could benefit preference learning solutions. Section 4 presents the key contribution of this paper - the Dynamic Incremental Associative Neural NEtwork (DIANNE). DIANNE is a neural network based preference learning algorithm that utilises temporal information to handle issues such as pre-actions and provide accurate user preferences. Section 5 presents DIANNE performance results and a comparison with other algorithms. Finally section 6 concludes the paper.

II. Background

Since 2000 the GAIA project [3] has been developing a middleware infrastructure for smart homes and offices which it terms active spaces. The project adopts the typical approach to personalisation, as mentioned above. A training dataset is created from monitored user behaviour and context information and preferences are extracted using a Naive Bayes technique. GAIA agents continuously monitor the user's context and use the learnt preferences to drive automatic and proactive adaptations of the user's home/office environment.

The MavHome project [4] focuses on the learning and prediction of user *tasks* to drive adaptations within a home environment. Behaviour monitoring is performed in a similar way to the GAIA project. Then several machine learning techniques such as sequential pattern discovery and Markov chains are used to identify commonly occurring patterns of behaviour from the stores of monitored behaviour data. An incremental prediction algorithm (Active-LeZi) [5] is used to predict future tasks in real-time (e.g. in a given context, when the user switches on the VCR, they then switch on the TV).

The Adaptive Home (or Neural Network Home) project [6] constructed a prototype pervasive system in an actual residence in 1997. It utilises reinforcement learning and neural network techniques to learn the intentions of inhabitants within the smart home environment. The aim is to balance user requirements and energy conservation. To achieve this the Adaptive Home goes a step beyond other projects by employing learning techniques to build models of future context states for future context prediction (e.g. future occupancy of an area or future hot water usage). User behaviours are then analysed against predicted future context states to pro-actively adapt the home appropriately in terms of future user and energy requirements.

The Synapse system [7] performs environment adaptations under two modes; active and passive. Bayesian Networks are employed to learn preferences dictating the relationships between context states and service usage behaviour. This learnt knowledge is then applied to personalise the user’s environment through service provision. If a preference has a probability above some threshold, personalisation operates in active mode and the service is started automatically. If the preference has a probability below some threshold, personalisation operates in passive mode and the top five potential services are presented to the user for manual selection. This approach aims to minimise incorrect personalisation in uncertain situations while at the same time provide automation when appropriate.

Bayesian networks were also used to learn and represent user preferences in the DAIDALOS project [8] however this project also employed several other preference learning techniques, the key one being Quinlan's C4.5 decision tree learning algorithm [9]. The main benefit with this technique over network based approaches is that decision tree output can be easily mapped into human understandable rules. In DAIDALOS, an IF-THEN-ELSE format was used to store and present preferences.

More recently, projects such as Ubisec [10], MobiLife [11], SPICE [12] and iDorm [13] have supplemented offline monitoring and user preference extraction processes in an attempt to provide systems that are more responsive to preference changes. Real-time update mechanisms temporarily update the user's preference between learning executions if negative feedback is received due to incorrect personalised adaptations.

The solutions above have several commonalities such as how they monitor user behaviour and user context to build their datasets for future preference/task extractions. They all store the current context of the user with a monitored behaviour when the behaviour first occurs at one point in time. No temporal data is gathered relating to the duration that the behaviour endured. Section III illustrates how the above techniques may struggle to handle pre-actions and proposes that the use of temporal data could prove beneficial.

III. Preparational Actions

A pre-action is an action performed by the user in some context in preparation for entrance into a new context. For example, consider the following scenario:

A student is entering a lecture theatre. She mutes her mobile phone in the corridor outside the lecture theatre before entering. Once the lecture is over, the student un-mutes her mobile phone just before she leaves the lecture theatre.

In this short scenario the student performs two pre-actions; muting the phone before entering the lecture theatre and un-muting the phone before leaving the lecture theatre. Based on how the majority of personalisation systems handle behaviour and context data, the typical monitored dataset that would result from this scenario would include the instances:

volume = mute*, location = corridor*
volume = un-mute*, location = lecture theatre*

Since the user could potentially repeat these behaviours every time they enter or exit the lecture theatre, these instances could be repeated multiple times throughout the monitored dataset. Each time, the context that the user is in when they perform a behaviour is stored with that behaviour in the dataset since it is assumed that the context *c* in which the user exhibits some behaviour *b* is the context in which that behaviour is intended to endure and hence *b* should be directly associated to *c*. When machine learning techniques are eventually applied to extract preferences from this dataset it will typically result in the learning of a preference such as:

IF location = corridor
THEN volume = ***mute***
ELSE IF location = lecture theatre
THEN volume = ***un-mute***

This preference will then be used to drive future personalised adaptations. However, this preference is not correct. It is actually the opposite of what the user prefers. In the lecture theatre scenario the user performed behaviour *b* in context *c* as preparation for entry into future context *c'* with the intention that *b* should endure in (and hence be associated to) *c'*. The correct preference is actually:

IF location = corridor

THEN volume = ***un-mute***
ELSE IF location = lecture theatre
THEN volume = ***mute***

Essentially, a repeated or consistent noise has entered into the dataset due to the pre-actions.

A. A Temporal Solution

One may question if additional sensing and inference could provide a solution. With appropriate sensing and inference techniques the system could predict the user's future location (lecture theatre) for association with behaviours (muting the phone). However, even with additional sensing and future context prediction it may still be the case that the user is performing actions to prepare for entry into contexts that are more than one step ahead. Therefore this does not always resolve the issue and could still result in the incorrect association of context and behaviours.

If we reconsider the lecture theatre scenario, the user mutes their mobile phone outside the lecture theatre and then immediately enters the lecture theatre. Therefore, the "mute" state prevails for only a short time period in the context outside the lecture theatre, but prevails for a much longer time period in the context inside the lecture theatre. The temporal duration of the co-occurring behaviour and context provides important information that naturally leads one to conclude that the mute state is more strongly associated to the context inside the lecture theatre where it prevailed for a greater temporal duration.

Consider another scenario. In some context *c* the user sets their screen background colour to blue. This behaviour prevails for several minutes before the user sets their screen background colour to yellow. This behaviour prevails for a number of weeks. Note that the two behaviours only occur once in context *c*. The natural assumption for one to make is that the second action should be more strongly associated to the context due to its longer duration but without this additional temporal information the dataset and the preference learning algorithm will treat both behaviours as equally associated to the context.

Therefore, it is proposed that temporal data could prove beneficial to preference learning systems, enabling them to handle anomalies such as pre-actions. Of course, one could always pose scenarios where the temporal information also introduces noise, for example if the user performs a behaviour and then gets distracted. Any system which exploits temporal information for preference learning should consider this issue and ensure that such temporal noise is dealt with appropriately.

Section IV introduces the key contribution of this paper - the DIANNE preference learning solution which is designed to take advantage of temporal data from the real-world environment.

IV. The Dynamic Incremental Associative Neural Network (DIANNE)

The DIANNE [14] is a Dynamic Incremental Associative Neural NEtwork that learns associations between user context and user behaviours in an incremental, online manner. To date, it has been used as a key preference learning solution in two EU projects: PERSIST [15] and SOCIETIES [16]. The DIANNE is essentially a single layer neural network (although for ease it is described in terms of two layers) with weighted connections between nodes as illustrated in Fig. 1.

The context layer receives updates about the user's current context from some context provider. It acts as a pseudo-representation of the user's current context with the binary nodes in this layer being activated and deactivated to represent which context values are true or false at any time. The behaviour layer receives updates about the user's current behaviours from the services that the user interacts with. Equally, it acts as a pseudo-representation of the user's current behaviours with the binary nodes in this layer being activated and deactivated to represent which behaviours are true or false at any time. Nodes relating to the same context parameter or behaviour are grouped together and mutually exclusive policies are applied. This ensures that conflicting context values or behaviour values cannot be true at the same time. For example, in Fig. 1, if the "Kitchen" location node is true then the "Car" location node cannot also be true. Equally if the "Low" volume node is true then the "High" volume node cannot also be true.

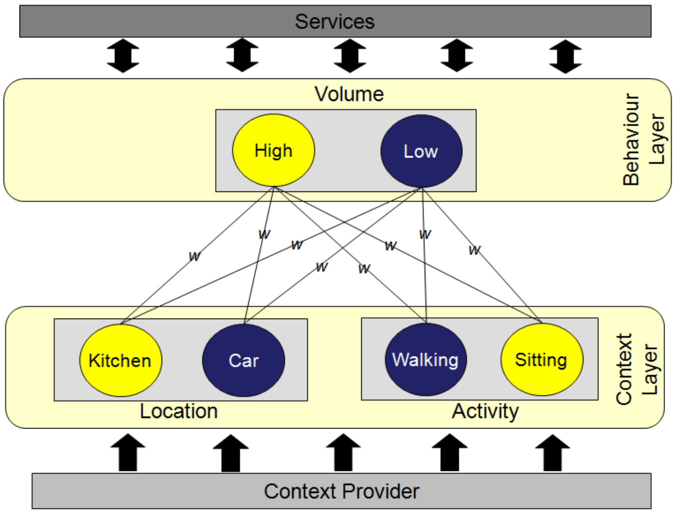


Figure 1. DIANNE Topology.

Context node activations are entirely dependent on context updates from the real world. Their *input potential* to the DIANNE is binary and directly dependent on their activation. Therefore when context node c_i is active, it will provide an input potential of 1 and if c_i is not active, it will provide an input potential of 0.

Behaviour node activations are dependent on both behaviour updates from the real world as well as internal network knowledge. Each behaviour node has an *output potential* value which indicates how strongly the DIANNE believes this node should be true in the current context. The output potential of an outcome node is the sum of its inputs; therefore the output potential *op()* of behaviour node b_j at time *t* is defined as:

$$op(b^t{}_j) = \sigma\left(\sum_{i=0}^{n} w^t{}_{ji} c^t{}_i\right)$$

where c^t_i is the input potential of context node c_i at time t, w^t_{ji} is the weight value between behaviour node b_j and context node c_i at time t and σ is the squashing function that maps the output potential from the possibly very large range of values to a finite range of values between -1 and +1. The output potentials of all behaviour nodes in the same group are compared and in most cases, the node with the highest output potential will be made active in the group in a *winner take all* fashion. However, it may be the case that contradictory behaviour updates are received from the real world. For example, the DIANNE may believe that the volume should be set to "high" in this context however the user has just set the volume to "low". When this occurs the DIANNE implements an online conflict resolution strategy to resolve the conflict in real-time. The weights of the conflicting nodes are updated based on the *Incremental Gradient Descent rule* to promote the output potential of the user performed behaviour and demote the output potential of the DIANNE favoured behaviour.

A. *DIANNE Algorithm and Temporal Learning Policy*

The key elements of the DIANNE learning algorithm are illustrated in Fig. 2. The algorithm is essentially a two step process comprising a layer update process and a learning process. However, this two step algorithm is executed in a continuous loop with a frequency of one second. Hence DIANNE learning occurs on a temporal basis enabling the DIANNE to exploit real-time information such as the temporal duration of user behaviours and context states.

It implements a temporal weight reinforcement policy following the hypothesis that *the time a behaviour endures in some context is just as important as the fact that the behaviour was observable in the context*. Therefore, in the DIANNE, the strength of the connections between context values and behaviours is not only based on the simultaneous occurrence of behaviours and context states but also the period of time that the simultaneous occurrence of behaviours and context states endures. This enables the DIANNE to overcome noise in the observed user behaviours such as that resulting from pre-actions.

During the layer update process, any new context or behaviour updates that have happened since the last algorithm cycle are processed and the context and behaviour nodes are updated accordingly. Then the *Hebbian/anti-Hebbian Learning rule* is applied to update all weights in the network depending on the activity of their connected context and behaviour nodes. If both c_i and b_j are active then weight w_{ji} will be increased; if c_i is active but b_i is not active then w_{ji} will be decreased and if both c_i and b_j are inactive w_{ji} will remain the same. Once all weights have been updated, the new output potentials of all behaviour nodes are calculated and the group nodes with the highest output potentials are identified. If conflicts arise between network knowledge and the real world state they are dealt with at this point as described above.

Reflecting back to the lecture theatre scenario, the DIANNE connection between the "lecture theatre" context value and the "mute" behaviour will be much stronger than the connection between the "lecture theatre" context value and the "unmute" behaviour since the weight on the first connection has been positively incremented for a longer time period.

The use of two learning rules (Hebbian for temporal reinforcements and Incremental Gradient Descent for conflict resolution) is key to dealing with noise introduced by the temporal information itself (i.e. when the user gets distracted after performing some behaviour, hence the behaviour endures in a context for a longer time than it should). If the user sets the volume of some service to "low" and then becomes distracted, the "low" node will be regularly reinforced in this context in line with the Hebbian rule. When the user eventually changes the service volume to "high" this will create a conflict situation and the "high" and "low" node potentials will be more radically updated in line with the Incremental Gradient Descent rule. The "high" node will be promoted and the "low" node will be demoted allowing the "high" node to compete with the "low" node without the "high" node having to endure for a comparable time in the context as the "low" node has.

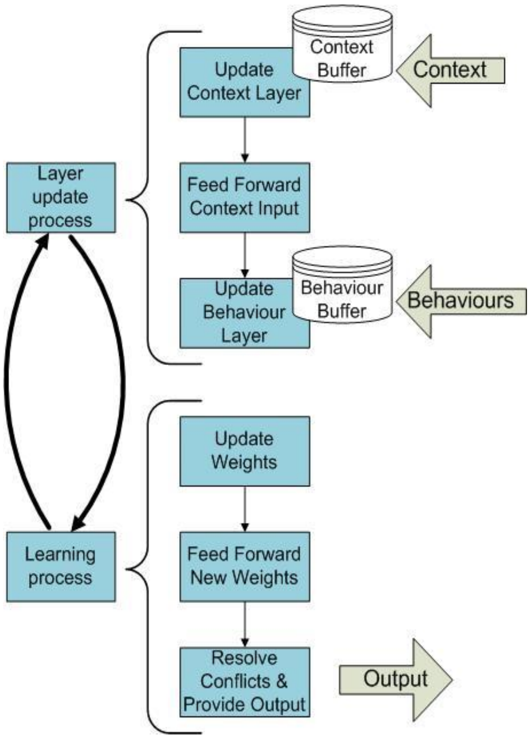


Figure 2. DIANNE algorithm.

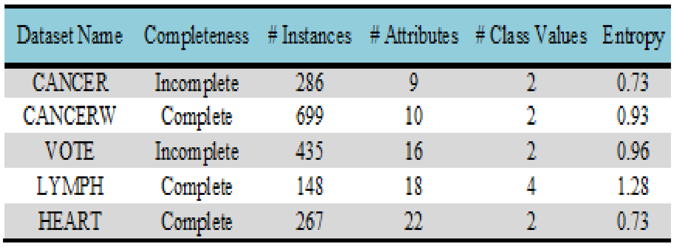
V. PERFORMANCE RESULTS AND COMPARISIONS

The DIANNE has been evaluated in two different ways. Firstly the DIANNE has been applied to benchmark datasets to determine performance and scalability as a machine learning solution. Secondly the DIANNE has been applied in a real-time preference learning situation to assess the benefits of exploiting temporal information in the preference learning process. Both evaluations and their results are described below.

A. *Performance and Scalability Evaluation*

A total of five benchmark datasets were chosen for the performance and scalability evaluation, all of which were sourced from the UCI Machine Learning Repository [17]. The characteristics of the chosen datasets are summarised in Table 1.

TABLE I. SUMMARY OF BENCHMARK DATASETS



Dataset Name	Completeness	# Instances	# Attributes	# Class Values	Entropy
CANCER	Incomplete	286	9	2	0.73
CANCERW	Complete	699	10	2	0.93
VOTE	Incomplete	435	16	2	0.96
LYMPH	Complete	148	18	4	1.28
HEART	Complete	267	22	2	0.73

The datasets were randomly split into 70% training and 30% testing subsets. The training subset was presented to the DIANNE one instance at a time in line with the DIANNE's real-time behaviour. Once all training data was presented the testing data was then also presented to the DIANNE one instance at a time and the DIANNE output was logged and compared against the instance's class value. This process was repeated ten times to give ten percentage accuracies which were then averaged to give one overall percentage accuracy for each dataset.

Fig. 3 presents the results of DIANNE performance on the benchmark datasets. The figure also shows the performances of other well cited algorithms where comparable results (using the same training to testing proportions) were available for the same benchmark datasets.

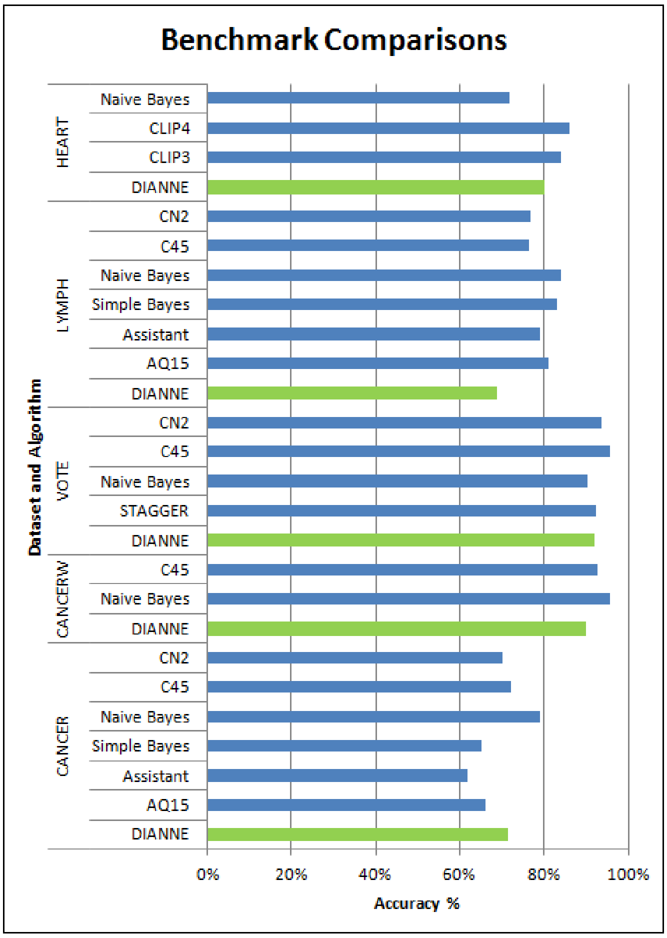


Figure 3. Comparison graph illustrating the accuracies of various algorithms on the datasets.

Notably, the DIANNE achieves accuracy figures that are as good as (and sometimes better than) the other algorithms over the various datasets. Compared to batch algorithms, the DIANNE performs comparably with C45 and outperforms CN2, Simple Bayes and Assistant on the CANCER dataset. The Naive Bayes algorithm is outperformed on the HEART and VOTE datasets. Compared to incremental algorithms, the DIANNE outperforms AQ15 on the CANCER dataset and achieves accuracies comparable to that of the STAGGER algorithm on the VOTE dataset.

B. Temporal Learning Assessment

To assess the benefits of exploiting temporal information in the preference learning process it was necessary to first create appropriate datasets. This was due to the fact that no existing datasets with appropriate temporal information could be sourced. To create such datasets a user trial was developed based on a personalised television experience. A trial environment was created at Heriot-Watt University involving technology such as plasma screens, tablet devices and environmental sensors. A total of 24 individuals participated in the trials. Most participants were postgraduate students.

Each participant was asked to make several visits to plasma screens placed in different locations around a University building. At each screen the participant was asked to choose a channel to watch. They were not given any directions on how they should make their choices and were free to watch channels, switch between channels and change their minds as they pleased. Participants could also exhibit pre-actions as it was possible to select channels before entering the immediate vicinity of a screen.

The behaviour and context of each participant was monitored and the temporal aspects of their behaviours and context was also captured. This resulted in the creation of two datasets for each trial participant. Dataset X contained no temporal information and is typical of the behaviour datasets captured by conventional personalisation systems (such as those mentioned in Section II). Dataset X^T extended dataset X as it also contained temporal information about user behaviours and context (each instance in the dataset was replicated for every second that it remained true). X^T is typical of the datasets expected by the DIANNE preference learning system. The datasets were then used to answer two key questions:

1. How does the performance of DIANNE (when applied to its expected temporal datasets) compare to the performance of a conventional preference learning solution (when applied to typical non-temporal datasets)?
2. Does the performance of a conventional preference learning solution improve when applied to a temporal dataset?

To answer question 1, for each participant, the DIANNE was applied to dataset X^T to give a performance value. Another conventional machine learning algorithm, the C45 decision tree learning algorithm, was applied to dataset X to give a performance value. The C45 algorithm has been used as a key preference learning algorithm in the DAIDALOS and PERSIST projects. To answer question 2, for each participant, the C45 algorithm was applied to dataset X^T to give a performance value. Figure 2. shows the three performance values on the datasets over 24 trials.

In relation to question 1, the graph shows that when applied to their typical datasets, the DIANNE is over three times more accurate than the C45 algorithm when learning the participant's viewing preferences.

In relation to question 2, the graph shows that there is some improvement in the accuracy of the C45 algorithm on the temporal datasets (A) compared to the non-temporal datasets (B), suggesting that additional temporal information is of benefit to preference learning processes enabling more accurate preferences to be learnt.

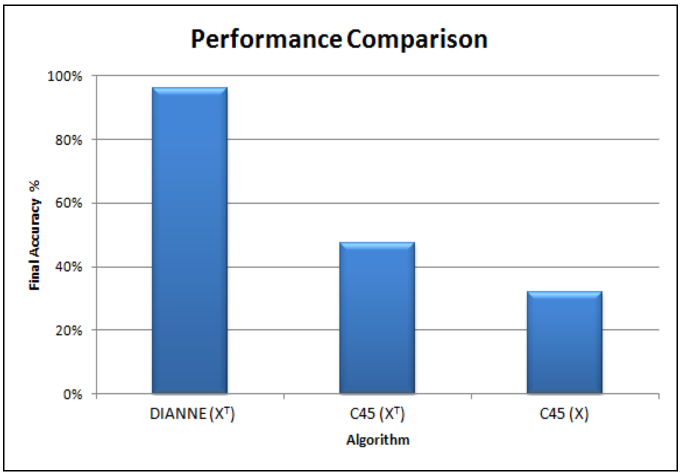


Figure 3. Performance comparisons on temporal and non-temporal data

VI. Conclusion

As our everyday environments become increasingly connected and technology rich, it is important that ubiquitous computing research continues to provide supporting functionalities such as personalisation, to optimise such environments (and the technology within) in line with user needs. A review of related work shows that ubiquitous systems often adopt a similar approach towards a personalisation solution. Typically, user behaviour is monitored and stored with the context in which the behaviour was first exhibited. This store of behaviour and context data is then processed to extract adaptation rules (or preferences) that indicate how environments should be proactively adapted on behalf of the user in the future.

No temporal data about the duration of behaviours is collected. In some cases this omission can lead to incorrect preferences when the dataset contains anomalies such as pre-actions, as illustrated in the lecture theatre scenario. A pre-action is where the user performs a behaviour in one context to prepare for entry to another context where they intend the behaviour to endure.

It is proposed in this paper that temporal information about behaviours and context states should be exploited in preference learning processes as it enables them to better handle anomalies such as pre-actions and provide accurate preferences. The DIANNE is introduced as a preference solution that does exploit temporal information about behaviours and context states. It is essentially a single layer, feed forward neural network that processes inputs incrementally in real-time as they occur in the real world environment. The DIANNE learning algorithm is a two step process that executes in a continuous loop with a frequency of one second. Weight manipulations are performed on each execution allowing the DIANNE to enforce a temporal learning policy where connection strengths are dependent on the period of time that the simultaneous occurrence of behaviours and context states endures. It follows the hypothesis that *the time a behaviour endures in some context is just as important as the fact that the behaviour was observable in the context*.

The DIANNE has been analysed in a two part evaluation. Firstly, it has been applied to several benchmark datasets to evaluate performance and scalability as a machine learning algorithm. The results show that the DIANNE performs comparably with other notable learning algorithms across all tested bench mark datasets. Secondly, the DIANNE has been applied to a real-time preference learning challenge in live user trials where the goal was to learn the context-dependent viewing preferences of trial participants. The results show that in these circumstances the DIANNE provides more accurate preferences than a more conventional preference learning approach. Additionally, the more conventional preference learning approach has also been applied to a dataset that included temporal information about behaviours and context states. The results show an improvement in performance suggesting that temporal information about behaviours and context states is a beneficial addition in preference learning solutions.

Acknowledgment

This work was supported in part by the European Union under the FP7 programme (PERSIST and SOCIETIES projects) which the authors gratefully acknowledge. The authors wish to thank all colleagues in both projects. However, it should be noted that this paper expresses the authors' personal views, which are not necessarily those of the project consortia. Apart from funding the PERSIST and SOCIETIES projects, the European Commission has no responsibility for the content of this paper.

References

[1] M. Weiser, "The Computer for the 21st Century", Scientific America, 1991

[2] Kantar Worldpanel ComTech website, http://www.kantarworldpanel.com/en/index.html#/News/news-list/The-smartest-way-to-communicate-over-half-of-the-GB-population-owns-a-smartphone.

[3] B.D. Ziebart, D. Roth, R.H. Campbell, A.K. Dey, "Learning Automation Policies for Pervasive Computing Environments", Proc. of the Second International Conference on Autonomic Computing (ICAC'05), 2005, pp. 193.

[4] G.M. Youngblood, L.B. Holder, D.J. Cook, "Managing Adaptive Versatile Environments", Pervasive and Mobile Computing, 1(4), 2005, pp. 373 - 403.

[5] G.M. Youngblood, E.O. Heierman, L.B. Holder, D.J. Cook, "Automation Intelligence for the Smart Environment", Proc. of the International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, 2005, pp. 1513 - 1514.

[6] M.C. Mozer, "The Neural Network House: An Environment that Adapts to its Inhabitants", Proc. of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, Menlo Park, California, 1998, pp. 110-114.

[7] H. Si, Y. Kawahara, H. Morikawa, T. Aoyama, "A Stochastic Approach for Creating Context-Aware Services based on Context Histories in

Smart Home", Proc. of Exploiting Context Histories in Smart Environments (ECHISE2005), 2005, pp. 3480 - 3495.

[8] M.H. Williams, N. Taylor, I. Roussaki, P. Robertson, B. Farshchian, K. Doolin, "Developing a Pervasive System for a Mobile Environment", Proc. eChallenges - Exploiting the Knowledge Economy, 2006, pp. 1695 - 1702.

[9] J.R. Quinlan, "C45: Programs for Machine Learning", 1993, Morgan Kaufman.

[10] J. Groppe, W. Mueller, "Profile Management Technology for Smart Customizations in Private Home Applications", Proc. of the Sixteenth International Workshop on Database and Expert Systems Applications (DEXA'05), Copenhagen, Denmark, 2005, pp. 226 - 230.

[11] M. Sutterer, O. Coutand, O. Droegehorn, K. David, "Managing and Delivering Context-Dependent User Preferences in Ubiquitous Computing Environments", Proc. of the 2007 International Symposium on Applications and the Internet Workshops (SAINTW'07), 2007, Article 4, Hiroshima, Japan.

[12] C. Cordier, F. Carrez, H. Van Kranenburg, C. Licciardi, J. Van der Meer, A. Spedalieri, J. P. Le Rouzic, J. Zoric, "Addressing the Challenges of Beyond 3G Service Delivery: the SPICE Service Platform", Proc. of Workshop on Applications and Services in Wireless Networks (ASWN '06), 2006, Kassel, Germany.

[13] H. Hagras, "Embedding Computational Intelligence in Pervasive Spaces", IEEE Pervasive Computing, 6(3), 2007, pp. 85-89.

[14] S. Gallacher, "Learning Preferences for Personalisation in a Pervasive Environment", 2011, PhD Thesis, Heriot-Watt University, UK.

[15] M. Crotty, N. Taylor, H. Williams, K. Frank, I. Roussaki, M. Roddy, "A Pervasive Environment Based on Personal Self-Improving Smart Spaces", Ambient Intelligence 2008, Springer-Verlag, Heidelberg, 2009, pp. 58 - 62.

[16] SOCIETIES Homepage, http://www.ict-societies.eu

[17] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml