Occupant Location Prediction Using Association Rule Mining

Conor Ryan and **Kenneth N. Brown**[1]

Abstract. HVAC systems are significant consumers of energy, however building management systems do not typically operate them in accordance with occupant movements. Due to the delayed response of HVAC systems, prediction of occupant locations is necessary to maximize energy efficiency. In this paper we present an approach to occupant location prediction based on association rule mining, which allows prediction based on historical occupant movements and any available real time information. We show how association rule mining can be adapted for occupant prediction and show the results of applying this approach on simulated and real occupants.

1 INTRODUCTION

Office buildings are significant consumers of energy: buildings typically account for up to 40% of the energy use in industrialised countries [1], and of that, over 70% is consumed in the operation of the building through heating, ventilation, air conditioning (HVAC) and lighting. A large portion of this is consumed under static control regimes, in which heating, cooling and lighting are applied according to fixed schedules, specified when the buildings were designed, regardless of how the buildings are actually used. To improve energy efficiency, the building management system should operate the HVAC systems in response to the actual behavior patterns of the occupants. However, heating and cooling systems have a delayed response, and so to satisfy the needs of the occupants, the management system must predict the occupant behaviour . The prediction system should be accurate at both bulk and individual levels: the total number of occupants of a building or a zone determine the total load on the HVAC system, while knowing the presence and identity of an occupant of an individual office allows us to avoid waste through unnecessary heating or cooling without discomforting the individual.

We believe that in most office buildings, the behaviour of the occupants tends to be regular. The regularity may be based on the time of day, or particular days of the week or times of the year. Behaviour within a day may also be dependent on behaviour earlier in the day, or dependent on the behaviour of other associated individuals. We require a system which is able to recognise these time and feature based patterns across different levels of granularity from observed data. Further, many office users now use electronic calendars to manage their schedules, and information in these calendars may support or override the regular behaviour. The reliability of the calendar data will depend on the individual maintaining it, and so the prediction system needs to be able to learn occupant-specific patterns from the calendars.

We propose the use of association rule mining for learning individual occupant behaviour patterns, using the Apriori algorithm [2]. From the individual patterns, we then aggregate behaviour to produce bulk occupancy predictions. We show how the algorithm can be extended to represent time series, incorporating calendar entries. We then propose a number of transformations of the learning mechanism, pruning itemsets and rules to focus in on useful rules, and extending the generation of itemsets in areas where useful patterns will be found. We evaluate the performance empirically on both simulated and real observed data, and show a 14% increase in accuracy over the base algorithm, reaching up to 79% accuracy in predicting occupant locations on real data and up to 85% accuracy in predicting whether rooms are occupied on simulated data.

The remainder of the paper is organized as follows: Section 2 provides an overview of association rules and the existing work on location prediction. Section 3 details the modifications we make to the mining process. In Section 4 we outline the datasets we use for evaluation and present our results. We conclude the paper in Section 5.

2 RELATED WORK

2.1 Location Prediction

Existing methods for predicting occupant locations include bayesian networks[3], neural networks[4], state predictors[5], context predictors[6], eigenbehaviors [7].

The Bayesian network approach presented in [3] predicts the occupant's next location based on the sequence of their previous locations and the current time of day and day of the week. Based on the current room and the day/time, it also predicts the duration of the occupant's stay in the current room. This results in separate predictions for the occupant's next location and for the time they will move there.

The neural network approach uses a binary codification of the location sequences as input to a neural network. In [4] both local and global predictors are considered. A local predictor is a network which is trained on and predicts a particular occupant, and thus deals only with codified location sequences. The global predictor takes all occupants' location sequences, along with associated occupant codes, as training data, and can make predictions for any occupant.

[1] Cork Constraint Computation Centre, Department of Computer Science, University College Cork, Ireland.

The state predictor approach in [5] uses a two-level context predictor with two-state predictors. This method selects a two-state predictor based on the occupant's sequence of previous locations. Each state within the selected predictor is a prediction; the current state is used as the prediction, and the state may then change depending on whether the prediction was accurate. Being a two-state predictor, each possible location has two corresponding states, so a maximum of two incorrect predictions for any given sequence is necessary to change future predictions, resulting in fast retraining if an occupant changes their behavior.

The second level of this predictor can store the frequencies of the possible next locations for each sequence, instead of state predictors. This makes it equivalent to a markov model approach.

These approaches all predict the occupant's next location, and with the exception of the Bayesian network, only use the occupant's recent locations. Our application requires longer term predictions and we believe there may be more general associations between the occupants' locations at different times which allow for such predictions. Association rule mining is intended to discover general patterns in data and so we propose to investigate whether association rule mining can be used to predict occupant locations.

2.2 Association Rules

Association rule mining was introduced in [2] as an unsupervised approach to finding patterns in large datasets. The original application was discovering patterns in datasets of transactions, where each transaction was a market basket, i.e. a set of purchased items. In that application items were literals, simple strings which are either present or absent in a transaction; however the algorithm can be applied without modification to sets of attribute/value pairs. We chose the Apriori algorithm as it is the most basic association rule mining algorithm and thus simplest to modify.

Let U be a universe of items. A dataset D is a set of instances $I_1 \dots I_n$, where each instance is a set of items from U. An itemset X is a subset of U. The frequency of X, $freq(X)$, is the number of instances I in D for which $X \subseteq I$, while the support is:

$$supp(x) = freq(X)/|D| \quad (1)$$

An association rule is an implication of the form $X \Rightarrow Y$ where X and Y are itemsets such that $X \cap Y = \emptyset$. This rule states that each instance which contains X tends to contain Y. The support of the rule is $supp(X \cup Y)$. The confidence of the rule is how often it is correct as a fraction of how often it applies:

$$conf(X \Rightarrow Y) = supp(X \cup Y)/supp(X) \quad (2)$$

The purpose of an association rule mining algorithm is to find the set of rules which are above user-specific thresholds of confidence and support. The first step is to find all itemsets which are 'frequent' according to the support threshold. Association rules are then generated from these itemsets, and any rules which fall below the user-specified minimum confidence are discarded. Confidence is used to measure the reliability of a rule in terms of how often it is correct according to the training data. Finding the frequent itemsets is the more difficult step, as the desired itemsets must be found among the total $2^{|U|} - 1$ itemsets which can be generated.

Apriori uses breadth first search to find all frequent itemsets. First all itemsets of size 1 are enumerated. Itemsets whose support falls below the support threshold (infrequent itemsets) are removed, as any superset of an infrequent itemset will also be infrequent. Candidate itemsets of size 2 are then generated by combining all frequent itemsets of size 1, and infrequent itemsets of size 2 are removed. This process continues, finding frequent itemsets of size n by generating candidates from the itemsets of size n-1 and removing infrequent itemsets, until an n where no frequent itemsets exist is reached.

Once the frequent itemsets have been found, for each frequent itemset X all rules of the form $Y \Longrightarrow X - Y$ where $Y \subset X$ and $Y \neq \emptyset$ are generated, and those which do not obey the confidence threshold are discarded.

3 ADAPTING ASSOCIATION RULE MINING FOR OCCUPANT PREDICTION

The first task in applying association rule mining is to determine the format of the dataset. We define an instance to be a single day for a single occupant, recording for each time slot the location of the occupant. It also includes a set of scheduled locations, specifying where the occupant's calendar stated they would be. Finally, each instance records which occupant and day of the week it applies to. Thus the set of attributes in our dataset is $A = \{d, o, l_i \dots l_j, s_i \dots s_j\}$, where d is the day, o is the occupant, l_n is the occupant's location at time slot n, and s_n is the location the occupant was scheduled to be in at time n. Our objective then is to find rules which predict the value of an attribute in $l_i \dots l_j$ based on the other attributes. In order to be able to compare confidences meaningfully, we restrict our attention to rules which predict single attributes.

Although this format is all that is needed to run Apriori, it is unlikely to produce usable results. The items in our dataset have semantics which are critical for the eventual application, but Apriori by default treats them all as equivalent.

The location attributes $l_i \dots l_j$ represent an ordered list of time/location pairs which it is our objective to predict. However, Apriori has no concept of the importance of or ordering over these items, so it will produce rules which run counter to the order, i.e. rules which use later locations to predict earlier locations, and which make useless predictions, e.g. predicting timetable entries.

A further important attribute distinction is that $l_i \dots l_j$ and $s_i \dots s_j$ are actual location data, whereas d and o are data labelling the location data, i.e. meta-data. Due to this their values are in a sense fixed. For example, in an instance which describes occupant A's movements on a Monday, d and o are fixed at Monday and A respectively, whereas all the other attributes can, in principle, take any value in their domain. This affects the meaning of the support metric as the maximum support for any itemset which includes d or o will be less than 1. Since support is used to determine which itemsets are considered frequent, patterns which occur frequently for certain days and/or agents will be rated as less frequent due to the inclusion of other days and agents in the dataset.

A problem with regard to the content of the data is that the many common patterns tend to be the least interesting, while we require low frequency patterns to be found in order to make predictions in unusual circumstances. Consider for example an occupant who has a 90% chance of being in their office in any

timeslot from 9am to 5pm. In this case, any pattern of the form "in at N implies in at M" where N and M are between 9-5 will have support of at least 80%, thus all such patterns will be found. But there is no real correlation there; all these patterns could be summarized simply as "the occupant is likely to be in". At the extreme opposite end, we have days when the occupant does not turn up at all, due to illness or other reasons – a very obvious pattern which would be represented by rules such as "out at 9,10,11 implies out at 12". Such rules could have confidence close to 100% if the occupant tends to be in in the morning, but if absences are rare the itemset behind the rule will have such low support it won't even be a candidate. Since enumerating every itemset is not feasible, we wish to eliminate the common uninteresting ones and focus on the less common but interesting ones.

3.1 Candidate/Rule Pruning

As mentioned above, standard Apriori has no concept of the relationships between the items in an instance which exist in occupancy data. Due to this it will by default generate some useless rules. The important features are that the location attributes $l_i \dots l_j$ represent an ordered list and that they are the only attributes we wish to predict. As an itemset which does not contain any of these attributes cannot produce a rule which predicts any of them, we eliminate itemsets which do not contain some subset of $l_i \dots l_j$ during candidate elimination. This prunes areas of the itemset lattice which could not provide useful predictions.

With regard to rule generation, we only wish to predict the future based on the past (i.e. rules which obey the ordering of $l_i \dots l_j$), and we only wish to predict a single location at a time in order to allow meaningful comparison of the rules at rule selection time. Thus our rule generation is as follows: for every itemset $\{l_i \dots l_j, x_i \dots x_j\}$, where l is a location item and x is any other type of item, l_j is the consequent and all other items are the antecedent.

3.2 Support Modification

In 2.1 we provided the typical definition of support, the proportion of the instances which contain the itemset/rule. To deal with the reduction in support for itemsets which contain metadata items, we redefine support as follows:

$$supp(X) = freq(X)/max(freq(X)) \quad (3)$$

For market basket items, which can in principle occur in every instance, this is the same definition. In the case of our metadata attribute/value pairs however, this definition results in a different value which is normalized such that the maximum value of supp(X) is always 1 for comparison to other support values.

Using this modified support threshold in Apriori allows it to find itemsets when have a lower support due to their metadata attributes. However this greatly increases the area of the itemset lattice which is explored for any given support threshold. Thus, in order to conserve memory, we mine each possible combination in a separate pass. For every combination of metadata attributes/values C, we initialize Apriori with all itemsets of size $|C| + 1$ which are a superset of C, instead of standard 1-itemsets. This allows the generation of every itemset which contains that metadata combination in a separate pass.

3.3 Windowing

Some important patterns have such low support that trying to find them by simply lowering the support threshold would result in a combinatorial explosion. Instead we will use the structure of the data to target them specifically. An example of such a pattern is a full day of absence: a very obvious pattern, but one which occurs so infrequently that it won't be learned. As our location attributes form an ordered list we can define subsets of them which are consecutive, temporal windows over the location data. By mining these subsets individually, we can reduce the size of the space of itemsets while still discovering the itemsets which describe consecutive elements of the low support patterns.

We define a window as: $Win(n,m) =< d, o, l_{n\dots n+m}, s_{i\dots j} >$ where i and j denote the first and last timeslots, and n and m denote the beginning and length of the window respectively. In the windowing phase, we search within every window of the chosen length. This approach ignores patterns which span times which do not fit within a window. We choose to focus on patterns which occur in consecutive time slots as predicting occupant locations based on their most recent movements has been shown to work by the other approaches discussed in 2.2.

For distinct patterns windowing is sufficient to find rules which will make the correct predictions should the pattern recur. Taking the example of an occupant who is absent all day, within each window we will learn that consecutive hours of absence imply absence in the next hour. Taken in combination, these rules will state that at any hour of the day, consecutive absence implies continued absence, although we are still not learning sequences in the same sense as the approaches in 2.2, as the individual rules are still tied to specific time slots. These rules are added to the rules mined from the complete instances.

3.4 Rule Selection

Once the rules are generated we need a mechanism to choose a rule to make a prediction. When a prediction is required, values for any subset of the possible attributes can be supplied in the form of an itemset V. A target for the prediction l_t is also given. We search the generated rules for all rules $X \Rightarrow Y$ where $X \subseteq V$ and $Y = \{l_t\}$. From these we select the rule with the highest confidence to make the prediction[2].

4 EXPERIMENTAL EVALUATION

To test our approach we use three different datasets: (i) data obtained from a simulator which generates occupancy patterns, (ii) data recorded by occupants of the 4C lab in UCC, and (iii) data from the Augsburg Indoor Location Tracking Benchmarks [8].

4.1 Occupancy Simulator

We have developed an occupancy simulator loosely based on the work of [9]. We model the activities which the occupants engage in during the day to determine their location. The model can also be viewed as a markov chain, where each state is a set of generated events and each transition adds a new event to the set.

[2] Confidence provided the highest accuracy over several evaluated metrics, though the difference was marginal

We have a set of agents A for whom we wish to generate locations over a period of time. To determine these locations we have a set of tasks T which can be assigned to agents. Each task has a set of attributes, enumerated below, which determine the agent's location for a period of time. To assign tasks to agents we have a set of roles R, each of which contains a set of tasks. Each agent then has a set of roles which apply to them. A role is chosen for each agent from their respective set at the beginning of each day based on a probability distribution over their possible roles.

Roles are defined as $R_i =< A_i, F_i, D_i, m_i, P_i, S_i, V_i >$ where A_i and $F_i$are sets of possible arrival and finish times outside which the agent is absent, D_i is a set of possible durations for the role, m_i is a flag indicating whether the agent may attend meetings in this role, and P_i, S_i, V_i are sets of tasks of different types.

There are three different types of task in order to represent different kinds of activities which can occur. These types differ in how the task's start time is selected and the order in which the tasks are evaluated:

Primary tasks are evaluated first: $P_i =< T_i, D_i, L_i, p_i >$ where T_i is a set of possible start times, D_i is a set of possible durations, L_i is a set of possible locations, and p_i is the base probability of the task occurring. Values are chosen from T_i, D_i and L_i based on associated probability distributions. A primary task will not occur if the agent is not available at the time chosen.

Secondary tasks are evaluated once all primary tasks are done: $S_i =< T_i, D_i, L_i, p_i >$ where the attributes are as above, except that times are selected from T_i in a set order until a free time slot is found. A secondary task will not occur if the agent is not free at any of the possible times.

For every empty slot which is left unassigned at the end of the process, a tertiary task is select to fill the slot: $V_i =< l_i >$.

Tertiary tasks are selected based on an associated probability distribution in the role. They have fixed duration and can appear multiple times, and so their only property is a location.

Primary tasks represent activities which must occur at a fixed time if they occur and take priority over other tasks, while secondary and tertiary tasks represent less important and miscellaneous activities which are scheduled around other tasks.

We also model meetings, which are essentially tasks involving multiple agents. Because they involve multiple agents, each meeting has a set of relevant agents rather than each agent having a set of meetings. We have two types of meetings:

Primary meetings are evaluated before all other tasks and meetings: $PM_i =< T_i, D_i, L_i, p_i, R_i, O_i >$. Primary meetings have the same attributes as primary tasks. In addition, they have a set of required agents R_i, whom all must be available for the meeting to occur, and a set of optional agents O_i, who will attend if possible.

Secondary meetings are evaluated after primary meetings and primary tasks: $SM_i =< T_i, D_i, L_i, p_i, A_i, m_i >$. Secondary meetings have the same properties as secondary tasks. In addition, they have a set of agents A_i a minimum number of available agents required for the meeting to occur m_i.

The model used to generate the data used in this evaluation includes 8 agents. 3 of these are lecturers, the remaining 5 are students. The lecturers each have a one person office, which they leave to give lectures and labs at fixed times with high probability. The students share an open plan office, which they leave to have one-on-one meetings with their supervisor (one of the lecturers). All agents also go to lunch, choosing whether to remain in the building or leave based on individual probabilities, attend a weekly group meeting which is dependent on certain agents being present to run it, and have a chance of missing an entire day due to illness. To simulate how a system using this approach would actually function, we use a contiguous dataset which is split such that there is 2 months of training data with the following 3 weeks as test data.

4.2 UCC Data

To gather data to test our approach, five occupants of the 4C lab in University College Cork including the authors manually recorded their movements over a period of 1-2 months using google calendar. Each occupant recorded their location by room code if within a campus building, or marked themselves as 'away' if off campus. The data was recorded from 8am to 6pm with half-hour granularity, with any occupancy of significantly shorter duration than 30 minutes filtered out. The occupants also recorded their timetables for the time period, which recorded the locations they were scheduled to be in in the same format as the record of their actual movements. 20 locations were frequented by the occupants including the 'away' location. The test set for this evaluation was the most recent two weeks of data for each occupant, while the training set was all the preceding data each occupant had recorded, which covered between 3 and 7 weeks.

4.3 Augsburg Dataset

This external dataset contains data on 4 occupants for 1-2 weeks in summer and 1-2 months in fall. The format of the dataset is a series of timestamped locations for each occupant. In order to be able to apply Apriori to the data, we converted it to the same timeslot format as our gathered data. An occupant's location in each timeslot is the location they spent the majority of that timeslot in according to the original data. Following this conversion there are 7 locations frequented by the occupants including 'away'.

4.4 Experiments

We generate rules from each training set using a minimum support and confidence of 0.2 and 0.5 respectively. During windowing we use a window size of 6 slots and a minimum support of 0.05. We evaluate the predictions of the association rules for each test set by predicting the location of each occupant at each time slot in the test set and recording the following statistics:

- Overall Accuracy: The percentage of predictions made which were correct
- Exact Occupancy: The percentage of room/time combinations for which the correct occupancy was exactly predicted
- Binary Occupancy: The percentage of room/time combinations for which the room was correctly predicted to be occupied or not

Occupancy level prediction accuracy is only available on the simulated dataset, as the collected dataset does not feature shared rooms, rendering occupant location prediction and occupancy level prediction essentially the same.

We test the association rules on their ability to predict with and without the timetable data available, and their ability to predict next-hour and next-day. The former determines whether the values of $s_i \ldots s_j$ are available when predicting, and is marked 'no

Timetable' if they are not. The latter determines whether $l_i \ldots l_{n-1}$ are available, where n is the time slot being predicted, 'Next Hour' if this information is available, and 'Next Day' if it is not.

4.5 Results

4.5.1 Algorithm Modification

Table 1. Algorithm Evaluation Results (UCC Dataset)

	No Support Mod / NoWindowing	No Windowing	All
Overall Accuracy	65%	74%	79%

Table 1 shows accuracy gains due to the extra areas of the itemset lattice explored due to Windowing and Support Modification. Support Modification adds the most accuracy, as it allows searching for more occupant-specific patterns. Windowing provides a smaller boost by allowing us to find patterns which describe rarer events.

The predictor highlights any instances where less than 50% of time slots were predicted correctly as problem instances. The addition of Support Modification and Windowing reduce the number of these problem instances by approx. 66% on all datasets.

4.5.2 Generated Data

Table 2. Generated Data Evaluation Results

Prediction	Overall	Exact	Binary
Next-Hour	69%	67%	85%
Next-Day	66%	66%	83%
Next Hour (No Timetable)	69%	66%	85%
Next Day (No Timetable)	64%	64%	81%

Table 2 shows the prediction accuracy for the agents in the generated data. These agents have a high probability of attending their timetabled tasks and narrow windows within which they arrive at and leave the building. Due to this their overall behavior is quite predictable, resulting in only a 5% drop in accuracy when predicting next-day locations without timetable data compared to next-hour predictions. The variations in their behavior, such as missing tasks or entire days, are purely random and thus unpredictable, reducing the accuracy in all tests.

The exact and binary occupancy level accuracy values vary due to the open plan office which the students share. The exact occupancy is the same as or slightly lower than the overall accuracy as the occupancy for the open plan office can be predicted incorrectly if any one student is predicted incorrectly. The binary occupancy accuracy is significantly higher due to the high probability of the open plan office being occupied by at least one student coupled with the high probability of correctly predicting the presence of at least one student.

4.5.3 UCC Data

Table 3. UCC Data Evaluation Results

	Next Hour	Next Day	Next Hour (no Timetable)	Next Day (no Timetable)
Overall Accuracy	79%	66%	79%	65%

Table 3 shows higher overall accuracy on the UCC data than on the generated data, which may indicate that the regular habits of individual people are less random than the probability model in our simulator. The drop in accuracy for next-day predictions compared to next-hour predictions shows that more intelligent predictions of the real occupants' movements later in the day can be made based on their movements earlier on, if that information is available.

Figure 1 shows the overall accuracy across the day. The next day and next hour predictions are of equal accuracy at the beginning and end of the day and at lunch time, as these are the times at which people's movements are most reliable.

Outside these times, data on the occupant's movements that day are needed to make accurate predictions. The next-day predictions drop in accuracy at 14:00. This is due to the system being unable to make a prediction for 14:00 without information on earlier movements, indicating that relevant rules based solely on historical movements at that time were below the confidence threshold.

Figure 2 shows per-occupant overall accuracy for next-hour and next-day predictions. In all cases next-day predictions are equal or worse than next-hour predictions, but the degree varies between the agents. One occupant is predictable enough that the overall accuracy is equal, though this does not necessarily mean that exactly the same predictions were made in both tests.

The availability of timetable data makes no difference to the accuracy of the predictions in either case. In general the timetable entries in this dataset are weekly meetings. Without the presence of irregularly scheduled meetings no extra useful correlations can be found. Furthermore meetings can be cancelled, relocated or rescheduled without sufficient notice to update the timetable, or the occupant can be absent, all of which will reduce the reliability of the timetable-related rules.

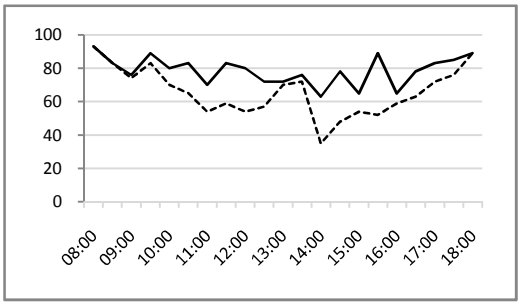

Figure 1. Prediction accuracy across the day for next-hour (solid) and next-day (dashed) predictions

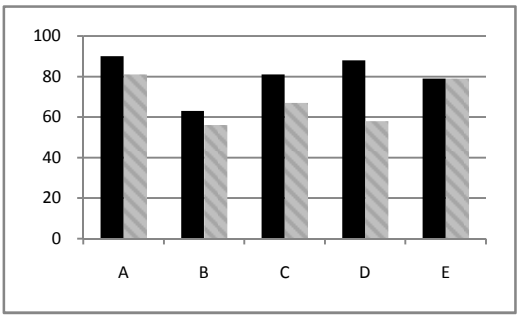

Figure 2. Prediction accuracy by occupant for next-hour (solid) and next-day (dashed) predictions

A recurring pattern in the selection of rules is the use of timetable entries to predict movements before the meeting, and the

use of those predicted movements to predict attendance at the meeting. This does not help accuracy on next-day predictions, as in these cases the meetings are always scheduled every week and thus do not allow discrimination between different patterns of movement. However, this does show that in cases where the meeting may or may not be scheduled, the timetable data could improve accuracy for times before the scheduled event, as well as for the event itself.

4.5.4 Augsburg Dataset

Table 4. External Data Evaluation Results

	Two Season	Fall
Overall Accuracy	40%	56%

Table 4 shows overall accuracy on two evaluations on this data. The 'Two Season' evaluation trains on the summer data and tests on the fall data. Our prediction accuracy is low while accuracy levels of 70-80% are achieved in [10]. We believe there are two reasons for this. First, the occupant patterns are predictable sequences but at irregular times of day and our approach cannot learn a sequence of movements independently of the time at which it occurs. If an occupant repeats the same sequence of movements at different times, other approaches will treat this as reinforcement of a single sequence, whereas our approach will attempt to learn rules representing multiple separate sequences. Second, the evaluation in [10] only predicts the destination of an occupant when they move rather than predicting their location at each time slot.

The 'Two Season' evaluation limits the training data to a maximum of two weeks for each occupant, therefore we also evaluated a split of the fall data where the final week is used for testing, leaving approximately a month of preceding data per occupant for training. The increased training set results in a significant increase in accuracy, however our approach still has difficulty predicting these occupants.

5 CONCLUSIONS AND FUTURE WORK

In this paper we presented an approach for applying association rule mining to the problem of predicting future occupant locations. We implemented our approach using a modification of a standard association rule mining algorithm, and presented experimental results which show that our modification of the algorithm can predict actual occupant movements with a high degree of accuracy, but is dependent on the types of patterns in the occupancy data.

In comparison to standard approaches, association rule mining has some benefits and drawbacks. While most other approaches predict an occupant's next location from a sequence representing the occupant's recent movements, our aim is to predict for any time slot using whatever information is available, whether it be the occupant's recent movements the same day, or simply their historical patterns. This is successful on two of the datasets, but our approach's inability to learn time-independent sequences means we fall short of the existing approaches on the third dataset. We intend to perform a deeper investigation of how our approach compares to existing approaches on our datasets.

There is existing work on extending Apriori to add new functionality, including sequence mining and mining with taxonomies. As part of our future work we intend to integrate these features into our modified Apriori. Sequence mining will allow it to perform better on datasets similar to the Augsburg dataset, as well as improving performance on the other datasets, by finding time-independent patterns. Taxonomies would allow the specification of a hierarchy over the possible locations an occupant may occupy. For example, we may generalize all locations to a simple in or out of office value for each occupant, and then find that in some cases where we cannot predict an occupant's exact location with high confidence, we can be highly confident that they will not be in their office.

We also wish to consider inter-instance mining. In many cases an occupant's movements will depend on other occupants, for example a meeting not occurring due to someone being absent. By learning patterns between occupants we could recognize these cases and improve our prediction accuracy.

The eventual goal is to integrate this approach with occupant localization systems such as [11], and predictive control systems such as [12]. Using occupant localization data, a system based on our approach could provide the predictions necessary for more energy efficient building control.

REFERENCES

[1] World Business Council for Sustainable Development, *Energy Efficiency in Buildings: Facts and Trends – Full Report*, (2008).

[2] R. Agrawal and R. Srikant, *Fast Algorithms for Mining Association Rules in Large Databases*, Proceedings of the 20th International Conference on Very Large Data Bases, 487-499, (1994).

[3] J. Petzold, A. Pietzowski, F. Bagci, W. Trumler and T. Ungerer, *Prediction of Indoor Movements Using Bayesian Networks*, LoCA 2005, LNCS 3479, 211–222, (2005).

[4] L. Vintan, A. Gellert, J. Petzold, and T. Ungerer, *Person Movement Prediction Using Neural Networks*, Technical Report, Institute of Computer Science, University of Augsburg, (2004).

[5] J. Petzold, F. Bagci, W. Trumler, T. Ungerer, and L. Vintan, *Global State Context Prediction Techniques Applied to a Smart Office Building*, The Communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, CA, USA, (2004).

[6] C. Voigtmann, Sian Lun Lau, K. David, *A Collaborative Context Prediction Technique*, Vehicular Technology Conference, (2011)

[7] N. Eagle and A. S. Pentland, *Eigenbehaviors: Identifying structure in routine*, Behavioral Ecology and Sociobiology, vol. 63, no. 7, 1057–1066, (2009).

[8] Jan Petzold, *Augsburg Indoor Location Tracking Benchmarks*, Context Database, Institute of Pervasive Computing, University of Linz, Austria. http://www.soft.uni-linz.ac.at/Research/Context Database/index.php, (2005).

[9] G. Zimmermann, *Modeling and Simulation of Individual User Behavior for Building Performance Predictions*, Proceedings of the Summer Computer Simulation Conference (SCSC), 913–920, San Diego, California, (2007).

[10] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer. *Comparison of Different Methods for Next Location Prediction*, EuroPar 2006, LNCS 4128, 909–918, (2006).

[11] W. Najib, M. Klepal and S.B. Wibowo, *MapUme: Scalable middleware for location aware computing applications*, International Conf. on Indoor Positioning and Indoor Navigation (IPIN), (2011).

[12] A. E.D. Mady, G. Provan, C. Ryan and & K. N. Brown, *Stochastic Model Predictive Controller for the Integration of Building Use and Temperature Regulation*, AAAI (2011).

This work is funded by Science Foundation Ireland as part of the ITOBO project under grant No. 07.SRC.I1170.