Routine Learning: Analyzing Your Whereabouts

Susanna Pirttikangas, Jukka Riekki, Juha Röning

Intelligent Systems Group, Department of Electrical and Information Engineering
University of Oulu, Finland
Firstname.Surname@ee.oulu.fi

Abstract

A routine is a temporal context sequence that occurs often. In routine learning, already recognized contexts are utilized in modeling the user behavior. The methodology is presented via a use case scenario. Data collected from various ubiquitous sensors are used in recognizing and defining contexts, and association rules determine the routines. The focus is on testing the suitability of the Apriori algorithm for this application area. Several useful routines were derived from the user data, and the results show that data mining can be utilized in pervasive computing.

Keywords: Pervasive computing, association rules, machine learning, context-awareness, locationing

1. Introduction

This paper is for reporting and analyzing the connection of data mining algorithms and routine learning in a mobile environment. A routine is a temporal (context) sequence that occurs often. Routine learning has been studied in the frameworks of building intelligent or smart homes [1], [2], [3], in assisting patients with Alzheimer’s [4], [5], and studying PC users’ behavior [6].

In data mining [7], data analysis is generally done after the data has been collected. In a mobile environment, it is necessary to consider also the situations, where data is analyzed in real-time and the results of the analysis are sent to the user straight away in order to make things easier for her. Also, usually the data mining algorithms available are designed keeping in mind the fact, that a human analyses the results. When ubiquitous computing is concerned, this is not always the case. The system has to be automatic, and there should be minor interaction with the user. All in all, the problem is to specify what user actions and contexts are really important and meaningful.

We have generated a use case Know Your Whereabouts [8] that contains a vision of an intelligent mobile device, which records the actions and locations of the user, and identifies the user’s routines based on sensory information. The work started with recognizing important places from location data and associating simple actions to these locations. Context recognition is certainly an important factor in the system, but in this paper, we label the contexts by hand (guidelines are taken from real data) in order to test routine learning algorithms. In [9], [10], [11], and [12] there are algorithms for recognizing the contexts of a mobile user, carrying similar sensors we have used. Later on, we are going to user these algorithms for recognizing the contexts from our versatile data set.

Here, routine learning is considered as associating different contexts into each other. These associations are derived as IF-THEN rules, called association rules. The Apriori algorithm [13] is tested for deriving these rules from a data set, collected in an office environment. The results show that it is possible to utilize data mining in a mobile environment.

2. Know your whereabouts

Know Your Whereabouts [8] is a scenario on routine learning. The test settings for routine learning were designed based on this application scenario. In this section, the basic principles for enabling routine learning are described.

The scenario is the following.

Marie buys a new mobile device. She is not very familiar with all the services the device has to offer so she mainly uses it as a phone. Marie also has an electronic calendar. She takes the phone along everywhere she goes.

After a week, the device asks Marie if she wants to get familiar with all the possibilities of the device. If Marie helped the device to place her important locations on the map, she could start to use the location-sensitive services.
Marie cannot believe her eyes when the device asks whether she would like the device to go automatically into a silent mode when she enters the meeting room in her office. This is something she has been waiting for!

Marie will have a meeting in half an hour at her office and she is still at home. The device reminds her that it is time to go.

Marie goes to a meeting at the office. The device is silent.

The user’s important locations can be determined based on the location data. After the important places have been identified, the user is asked to name them. In this way, each place gets a higher and more abstract meaning attached to it. The device guides the user to give generally known names for locations. When “home” is used to label the user’s home, all services that understand the concept of home can start to utilize this information, for example. Also, the user can ignore some places if she does not think they are important.

The important places could be the office, the meeting room 1, the meeting room 2, the coffee room, etc. The mobile device also records the changes of profile (time stamps included). These could be silent, loud, beep, normal mode, etc. An imaginary data matrix of user behaviors is presented in Table 1.

In Table 1, the user contexts are expressed as variables (Loc for location, and Mode for phone profile), and each context is coded as a Boolean variable. If the user (or device) is in a context identified by a variable, the value of the corresponding variable is set to 1. Otherwise, the value of the variable is zero.

<table>
<thead>
<tr>
<th>Time</th>
<th>Loc M1</th>
<th>Loc M2</th>
<th>Loc Off</th>
<th>Mode Silent</th>
<th>Mode Loud</th>
<th>Mode Beep</th>
<th>Walking</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

From this data matrix, it is needed to determine which contexts occur together, in other words, which variables are associated. Another interesting problem is to predict, what the user is about to do next. Thus, the events that occur in (temporal) sequences need to be identified. In the next section, a method for finding associations from a data matrix of the kind shown above is described.

3. Association rules

Association rules were initially developed for analyzing market-basket data where rules like “A customer who buys beer and sausage will also buy diapers with probability 0.85.” The Boolean data matrix in Table 1 suggests what routine learning could be like. By following the use case and the data matrix in Table 1, rules (or associations) like LocMeet1 ⇒ ModeSilent or LocOff ⇒ ModeBeep could be recognized. These rules are called association rules, and in the following, definitions and methodology about finding these association rules are described.

An association rule is an expression \( X \Rightarrow Y \), where \( X \) and \( Y \) are sets of items. Given a database \( D \) of transactions, where each transaction \( T \in D \) is a set of items, \( X \Rightarrow Y \) expresses that whenever a transaction \( T \) contains \( X \) then \( T \) probably contains \( Y \) also. The probability is called the rule confidence and it is defined as the percentage of transactions containing \( Y \) in addition to \( X \) with regard to the overall number of transactions containing \( X \). In other words, the rule confidence is the conditional probability \( P(Y \mid X) \).
In this paper, different values for minimum confidence and support were tested, in order to achieve association rules for routine learning.

4. Test settings

The most common algorithm in discovering association rules is the Apriori algorithm developed by Agrawal et al. [13], [14]. The basic goal is to find all combinations of items that have certain user-specified minimum support (\textit{minsups}) and confidence (\textit{minconf}). The combinations that have transaction support above minimum support are called \textit{large} itemsets and all other combinations \textit{small} itemsets. The large itemsets are used to generate desired rules. That is if, say \textit{ABCD} and \textit{AB} are large itemset, then we can determine if the rule \textit{AB} \Rightarrow \textit{CD} holds by computing the ratio \textit{r}=\textit{S(ABCD)}/\textit{S(AB)}. Only if \textit{r}>\textit{minconf}, then holds the rule. The rule will have minimum support because \textit{ABCD} is large.

The pseudo code for Apriori algorithm is described below (from [13]).

1. \textbf{L}_{1} = \{large itemsets\};
2. \textbf{for} \{k = 2; \textbf{L}_{k-1} \neq \emptyset ; k + + \} \textbf{do begin}
3. \textbf{C}_{k} = \text{apriori-gen} (\textbf{L}_{k-1}); // New candidates
4. \textbf{forall} transactions \textit{t} \in \textbf{T} \textbf{do begin}
5. \textbf{C}_{t} = \text{subset} (\textbf{C}_{k}, t ); // Candidates contained in \textit{t}
6. \textbf{forall} candidates \textit{c} \in \textbf{C}_{t} \textbf{do}
7. \textit{c}.\text{count} + + ;
8. \textbf{end}
9. \textbf{L}_{k} = \{ \textit{c} \in \textbf{C}_{k} \mid \textit{c}.\text{count} \geq \textit{minsups} \}
10. \textbf{end}
11. \textbf{Answer} = \bigcup_{k} \textbf{L}_{k}

The data collected consisted of three different scenarios. One scenario was about 20 minutes long. The first scenario was repeated 10 times, the second and the third 5 times. The sampling frequency for the sensor box is 10Hz and for the Ekahau 0.5Hz.

Data was collected by going to different locations in the research group's office space. One of the three scenarios is described below:

1. The testee sits by a desk in location 1 for about six minutes. The device is on the table (illumination sensor upwards).
2. The testee gets up and walks to the elevators (location 4), and orders the elevator.
3. When the elevator arrives, the testee steps into the elevator, and goes down to the first floor in the building.
4. When arriving downstairs, the testee gets out of the elevator and walks to the main doors (location 5).
5. At the doors, the testee steps outside, and walks to the door beside a corridor (location 6).
6. The testee walks back to location 1.

The data was tagged every time a context changed. This way, the recognition of different contexts can be verified. At this point, the contexts were labeled by hand for further processing with the APriori algorithm, however. This means, that \textit{apriori} information from the tags were used in determining the contexts, and no actual context recognition occurred. Later on, we are going to use the data for context recognition.

The important locations were derived from the location data. The methodology of [8] was used here. The important location is considered as a circle, the radius covering an area of one room in our research laboratory. If the user stays in one location for more than five minutes, and on two different occasions during the measurement period, then it is considered to be important.

The known contexts labeled (and the sensor measuring the situation) were:
- walking, walking downstairs, walking upstairs, sitting, in an elevator (acceleration signals)
- inside, outside (illumination)
- in the office, in a library, in a lobby, in a stairway, in a restaurant (Ekahau)
Also, a context variable describing the profile status of a PDA (mobile phone) was integrated to the data matrix. It was assumed that in each location a suitable profile was used. The profiles are general, silent, outdoors, and meeting.

A binary data matrix was generated for the user behavior and contexts. The context variables were arranged according to the sensorbox data, so the transaction database consisted of over 6000 lines. In other words, this transaction file has 16 columns for the contexts and for each context there is either 0 or 1 indicating, whether the user is in this particular context. The Apriori algorithm was implemented with java.

5. Results and discussion

The routine learning starts by identifying important places from location data. In [8], we have verified that it is possible to associate simple contexts or actions to these locations using the (simplified) methodology of the Apriori algorithm. We built a prototype for an architecture, which aids in building interactive mobile services that adapt to the user’s situation. In our previous experiments, we had a data set consisting of over 60 hours of location data for three different users. In our prototype, a PDA could be profiled into general, meeting, silent or outdoors modes. The routines identified for one testee were (26 hours of data collection):

Location:TS380 ⇒ Mode:General, Location:TS335 ⇒ Mode:Meeting, Location:LabX ⇒ Mode:Silent, Location:Robolab ⇒ Mode:Silent, Location:CoffeeRoom ⇒ Mode:General, Location:Stairway ⇒ Mode:General, and Location:Medialab ⇒ Mode:Meeting.

For another testee, data were collected for about 22 hours. In this test, eight important locations were correctly recognized, and four routines were detected. From the third test, which lasted for a total of about 12 hours, three important locations were recognized and two routines detected. The amount of data in the database grows considerably as several users join the network. Each user provides merely 2.4 M of location data per each 24 hours. In the first prototype, the database is flushed of the actual location and profile from time to time.

In this paper, we wanted to test a more complex setting and extend the number of contexts. In the first prototype, the important locations and PDA profiles were considered. Therefore, there are only two variables to consider and one needs to calculate only the confidences values for these two variables, one for each important location. In this paper, there are several other contexts (walking, sitting etc.) to consider, so the Apriori algorithm is utilized with its full extent. The binary matrix containing the context variables was processed with the Apriori algorithm and different support and confidence values were tested. The processing time of our data set was less than one second. All the combinations for support values of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6 and confidence values of 0.6, 0.7, 0.8, 0.9 and 1 were tested. First, the algorithm produces the frequent k-itemsets, and then the rules. The most interesting rules were generated with a low support value and high confidence. This is natural, since it is important to know if some contexts occur together in most of the cases. If the user visits a meeting room only twice during the measurement period and always switches the profile into a silent mode, for example. Below, the rules for the support value of 0.1 and confidence value of 0.8 are shown.

Walking ⇒ general, inside 0.94
In the Office ⇒ general, 1
In the Office ⇒ sitting, inside, general, 1
In the library ⇒ sitting, inside, silent, 1
In the lobby ⇒ sitting, inside, meeting, 1
Silent ⇒ sitting, inside, in library, 1

The rules are more diversified than in our first prototype, since the user contexts sitting, walking, etc. have been considered also. When in the office, the user is probably sitting. she is inside, and the profile of the PDA is on a general mode, etc.

The number of rules was 97, but the rules shown above are selected so, that the location or the profile context is the main factor for the rule. The other rules are basically combinations of these rules. It seems that the methodology can be used in routine learning, as long as the number of contexts is not very high.

In Figure 2, the important locations derived from the location data collected are shown. There were three recognized places. Also, the routines identified with the Apriori algorithm, are shown. When these rules have been presented to the user, she can select and automate some of the actions. For example, the user can verify, that when in the library, the profile should be switched into a silent mode. Also, the contrast of the PDA display can be set higher or the ringing volume of a mobile phone set louder when outdoors, for example.
The rule inside does not seem very important here, but it is because the data was mainly collected indoors. If different scenarios are considered the meaning of indoors and outdoors would be more evident.

**Figure 2. Important locations and routines identified**

In this paper, it was assumed that the contexts used here can be reliably recognized. In our future work, the main problem will be the fuzziness of the situations. We probably cannot make a binary assumption for each of these contexts. The location contexts are easily verified with map information, but the walking, running, sitting etc. contexts, derived from the acceleration signals might not be so straightforward.

Furthermore, longer periods of data collection will be done. It is worth studying, how the rules already identified can be utilized when new data arrives, since the database must be flushed from redundant context data to avoid overflows.

**Acknowledgments**

This work was funded by TEKES and Academy of Finland. The authors would like to thank all the personnel in CAPNET program, and the participating companies.

**7. References**


