

# Unsupervised Activity Recognition using Temporal Data Mining

Mohamed Tarik Moutacalli  
*Departement of Informatics*  
*UQAC*  
*Chicoutimi, Canada*  
*mohamed-tarik.moutacalli@uqac.ca*

Abdenour Bouzouane  
*Departement of Informatics*  
*UQAC*  
*Chicoutimi, Canada*  
*Abdenour\_Bouzouane@uqac.ca*

Bruno Bouchard  
*Departement of Informatics*  
*UQAC*  
*Chicoutimi, Canada*  
*Bruno\_Bouchard@uqac.ca*

**Abstract**—Excellent results have been obtained from data mining techniques in many areas. This article presents one such technique, in the context of activity recognition in a smart home. We use sequential pattern mining to analyze the history of information transmitted by the sensors, discovering thereby the frequent activities of the home occupant. Then each of the activities is temporally segmented, in order to facilitate the recognition of activities already started or even ones that are about to start. Our tests revealed that this segmentation diminished the activity search time by more than 70%, and helped predict some activities before detecting any action.

**Keywords**-Activity Recognition; Temporal Data Mining; Temporal Segmentation; Smart Homes.

## I. INTRODUCTION

The use of data mining (DM) has recently become more widespread and fashionable. The best definition of DM for us is the one suggested by Frawley in 1992 [1]: "the nontrivial extraction of implicit, previously unknown, and potentially useful information from databases". In particular, spatio-temporal DM [2] deals with the specific complexity of databases containing information about time or space. Temporal DM techniques are used in fields so different and so numerous that we cannot list them all: security, the stock market, electronic commerce, health, the web, etc. What is remarkable is the rarity of research which exploits these techniques in a field as large and as favorable to them as activity recognition in a smart home. The smart home, part of the recent school of thought derived from ambient intelligence [3], refers to a trend that discreetly embeds miniaturized electronic devices (sensors) in everyday objects, in order to provide real time assistance to the home occupant, based on information recently sent by the sensors or already stored in the database. The smart home can be considered as a warehouse storing a variety of data from different sensors. The large volume of data makes DM techniques the most suitable for the analysis and retrieval of knowledge, i.e., recognition of an activity (e.g., preparing meals, washing dishes, etc.), whether it has already started or is about to start.

The complexity of daily activity recognition in a smart home is due to the large number of activities that an

occupant can perform. This complexity first causes a problem in creating model activities, an essential step in the process of activity recognition, where we find not only all the activities that an occupant habitually performs, but also the various actions that compose them. For example, a model of the activity preparing coffee can be composed of the actions: take cup, pour coffee, add milk and add sugar. It should also be noted that a model activity depends on the person observed, and thus it is impossible to use the same model activities for different home occupants. In our example, the activity preparing coffee for another occupant may be composed of the actions: take cup and pour coffee. For these two reasons we have included in our approach an unsupervised method for creating model activities.

The large number of activities causes a more serious problem in searching for an activity. Normally, to assist the occupant in real time, we should quickly find the required activity among all the model activities. Reducing the number of activities seems to be a natural way to speed up this search. For this purpose, we thought of using the DM technique of temporal segmentation [14]. The general idea is to create a set of time intervals for each activity, covering the periods when the occupant usually starts the activity. For example, the activity taking a shower might be segmented into two intervals: from 8:30 A.M. to 10:00 A.M. and from 7:00 P.M. to 10:00 P.M. This way, in searching for activities, we will not go through all the model activities, but only those with an interval that contains the current time. For example, if the current time is 1:00 pm, the activity taking a shower will not be considered.

Temporal segmentation proved to be a very efficient solution; moreover, it dealt with some occupant errors (due to cognitive deficiency) that no other approach has dealt with up to now. The different steps of this approach will be detailed in the next section, and the following section will be reserved for tests. Before concluding, we devote a section to related work.

## II. ACTIVITY PATTERN MINING

The first step in the process of activity recognition is the creation of model activities. The recognition agent observing the smart home occupant should have a list of all the activities that this occupant usually performs, as well as their component actions, in order to choose which activity the occupant is actually performing. Because of the large number of possibilities and the fact that model activities are unique to each person, we decided to use an unsupervised method for creating these models. In other words, the observing agent has to learn the model activities by analyzing the historical sensor data. Alireza et al. [4] have already dealt with this problem, but, as they saved all the sensor states or values all the time, activity recognition belonged more to the field of motif discovery [5]. Motif discovery is usually used in bioinformatics to speed up the detection of motifs in biosequences (e.g., DNA). So for them, an activity is a sequence that is repeated over time specifying the states or values of all sensors. Our approach, however, only saves the times and the names of sensors that have changed state or greatly changed in value, in order to reduce the size of our data warehouse without losing significant information. We interpret a small value change in a sensor as a noise and therefore ignore it. Thus, an activity is composed only of sensors that have significantly changed during the relevant time frame. Our data warehouse looks like TABLE 1:

TABLE 1. Data warehouse of active sensors per day.

Day	Sensors
1	5 2 1 4 3 2 4 6 8 3
2	6 2 1 4 7 9 2 6 1 6 9 4
3	9 8 3 7 6 1 6 9 7 1 2 1 4

For the record, the time when each sensor changed significantly has been removed from Table 1 for easy reading. For example, Day 1 is normally represented as follows: 5 (8:50) 2 (9:15) 1 (9:16) 4 (9:18) 3 (9:50) ...

As shown in Table 1, the data warehouse looks like those used in the field of market basket analysis [6], which makes sequential pattern mining [7] the most appropriate technique for detecting activities. Several algorithms have been proposed for this technique; but, to quickly find the closed frequent patterns, we chose the BIDE algorithm proposed by Wang and Han [8]. A pattern is frequent if

the number of its appearances is not less than a given minimum frequency, and it is closed if it is not included in any other pattern. For example, if the activity 2 1 4 is reported, we do not report the sub-sequences such as 1 4 if they appear the same number of times as the first sequence. The BIDE idea consists of enumerating the complete set of frequent sequences, then choosing the closed ones. We start by creating a sequence tree with a root  $\emptyset$ . A node N at level L in the tree can be recursively extended by adding one item. Then, by removing the infrequent sequences in the sequence tree, like the node 3, the remaining nodes form a frequent sequence tree which contains the complete set of frequent sequences. Figure 1 shows the frequent sequence tree built from Table 1 with a minimum frequency equal to 3:

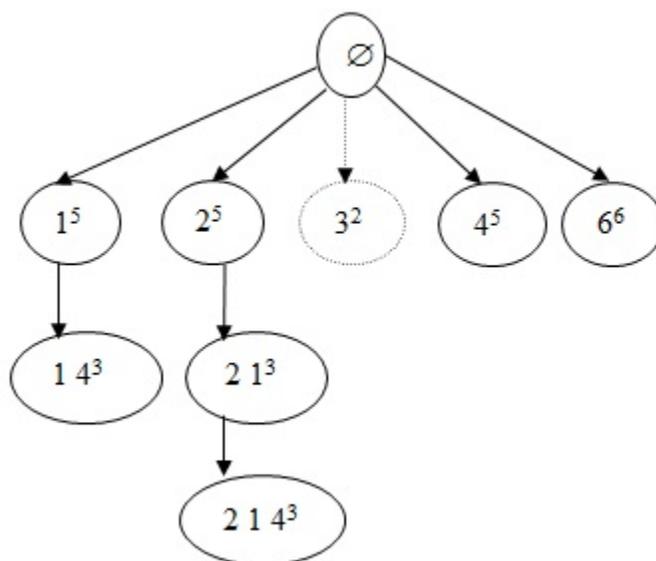


Figure 1. Sequences tree.

The customized version of this algorithm that we used allows us, at each execution, to specify the value of some very important parameters. The first indicates the minimum frequency of the activity relative to the total number of days. For example, if this value is set to 0.9 and the total number of days is ten, the activity must appear at least once in nine of the ten days. The other parameters specify the number of errors allowed for an activity and the number of errors allowed between two successive sensors. For example, if the activity is 2 1 4, then the sequence 2 6 1 4 can be interpreted as the same activity with one error that occurred between the first and the second sensor.

### III. TEMPORAL SEGMENTATION

The BIDE algorithm finds all the activity models, but even allocating to each model all of its starting times, we will not have a clear and useful idea of the occupants activity habits. For this reason, we need to transform all the starting times of each model activity into time intervals which cover the periods when the occupant usually starts the activity. At any moment, then, we can use the current system time to determine the activities that he usually performs at this time, by choosing activities that have an interval containing the current time. This selection allows us to reduce the number of possibilities. Instead of searching through all the activity models for an activity started at a given time, we search only among those that the occupant usually performs at that time. Figure 2 shows the result of segmenting two activities ( x and • ) into two intervals each. For example, if the current time is T0, the second activity will not be selected.

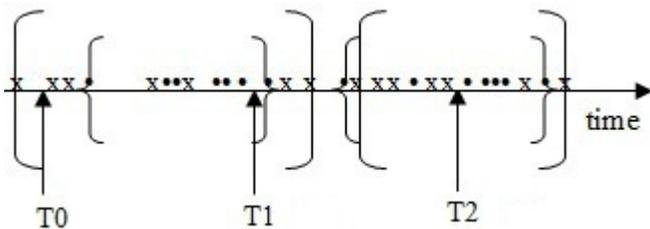


Figure 2. Example of activities temporals intervals.

The number of intervals for each activity is very important information greatly affecting the results of the segmentation. These numbers can be calculated by an algorithm such as C-means [9] for segmentation, but we found that each number is simply the maximum number of times the activity appears in a single day: it can be calculated when we visit our warehouse to get the activities starting times.

Knowing the number of intervals into which the activity is segmented, we could use the K-means algorithm [10] where k will be equal to that number. However, as we have to sort these start times during the activity search process, we decided to develop a simpler and faster algorithm, giving the same results as the K-means. The difference between the two algorithms is that the K-means creates intervals by assigning to them the times closest to their centers, while our algorithm (see Algorithm 1 below), looping k-1 times, looks for the two successive points farthest from one another, in order to create two separate intervals.

**Input** : a table of likely activities ActivProb <sup>1</sup>  
**Output** : a non-empty list of intervals for each activity in ActivProb

```

1: For each activity Activ in ActivProb
2:   i=1
3:   While i < Activ.NbrIntvl
4:     For each Time in Activ.TimeBeg And
       Time ∈ Table
5:       M = Max (NextTime - Time)
6:       Add IndiceM to Table
7:       Add 1 to i
8:   Sort_Asc (Table )
9:   Add 0 to ActivProb.Intvl
10:  For each T in Table
11:    Add T to ActivProb.Intvl
12:    Add T+1 to ActivProb.Intvl
13:  Add Activ.TimeBeg.size -1 to ActivProb.Intvl

```

<sup>1</sup> each activity is composed of :  
 NbrIntvl : the number of intervals  
 TimeBeg : sorted table of beginning time  
 Intvl : table of intervals ...

Algorithm 1. Creating activities temporals intervals.

Once the intervals are created, a final check is necessary to ensure efficiency by eliminating the gaps where the activity is very infrequent and dividing any interval where two successive times are very far from each other. This last step is detailed in Algorithm 2 below.

**Input** : ActivProb.Intvl  
**Output** : ActivProb.Intvl

```

1: For each Activ in ActivProb
2:   For each Intvl in Activ.Intvl
3:     If Intvl.NbrTime < (frequency/ Activ.NbrIntvl)
4:       Delete (Intvl )
5:     Else
6:       For each Time in Intvl
7:         ( NextTime - Time ) > MaxDist
8:         Add 1 to Activ.NbrIntvl
9:         End = Intvl.end
10:        Intvl.end = Time
11:        Insert NewIntvl
12:        NewIntvl.beg = Time + 1
13:        NewIntvl.end = End

```

Algorithm 2. Improving results of activities temporals intervals.

#### IV. ACTIVITY RECOGNITION SYSTEM

Temporal segmentation not only reduces the number of possibilities, but also can be used to calculate an initial priority for each activity. These priorities can tell us which activity is most likely to be initiated by the occupant, even before he acts. We have here a response to initiation errors committed by patients with Alzheimers disease that has not yet been described elsewhere in the literature, to our knowledge. Initiation errors are defined in the study by Baum et al. [11] as errors that prevent the patient from starting an activity.

The calculus of initial priorities is based on two observations:

1: The closer the current time to the end of an activity interval, the greater the likelihood of the activity. For example, going back to Figure 2 and assuming that the current time is T1; then, the activity 2 (•) is more likely than activity 1 (x). The formula we use for this observation is:  $P1 = \frac{Tc - Tb}{D}$ , where Tc is the current time, Tb and D are respectively the start time and the duration of the interval.

2: The higher the frequency before the current time, within an activity interval, the greater the likelihood of the activity. In other words, the occupant was more likely to perform this activity before the current time. For example, if Tc = T2 in Figure 2, then activity 1 (x) is more likely than activity 2 (•). The formula we use for this observation is:  $P2 = \frac{Nbc}{Nt}$ , where Nbc and Nt are the number of activities before the current time and the total number, within the interval.

Finally, the initial priority of the activity is:  $Pr = P1 + P2$ .

Our activity recognition system assigns to each likely activity its initial priority and proposes the activity that has the greatest priority as the most likely. Once the occupant acts, we must consider his action in our decision, so these values must be updated: we add 0.5 to the priority of each likely activity involving a sensor that has just come into use, and another 0.5 if the action detected is in the right place. For example, if we have two activities: 2 1 4 and 4 2 5 7, and if we detect that the first sensor that has been activated is the sensor 4, then the priority of the first activity will be augmented by 0.5 and the second by 1.0. After each action the priority of each likely activity is updated and the activity with the greatest priority is always proposed as the most likely.

#### V. VALIDATION

To test our approach, we used the database in van Kasteren et al. [12]. These data were recorded by observing an occupant for 28 days, using 14 sensors which detected

seven distinct activities: 1: 'leave house'; 2: 'use toilet'; 3: 'take shower'; 4: 'go to bed'; 5: 'prepare breakfast'; 6: 'prepare dinner'; 7: 'get drink'.

Concerning the creation of model activities, the results of our tests were a little different. In fact, just five of the seven activities were discovered. The first reason for this difference is that the occupant was used to perform the activity leave house just after activity 3, take shower, which meant that our algorithm recognized the two activities as one. This difference does not really matter because the occupant will be assisted in both activities and the next action can be predicted, but the use of other temporal information, such as the average duration of an activity or the maximum time between two successive sensors, may make such assistance more efficient. The fourth activity, going to bed, was not detected because it is carried out around midnight, which usually divides its actions between two days. This is a problem, not just for BIDE, but for all similar algorithms. In fact, BIDE created more than 71% of the model activities, but it assisted the occupant in more than 85% of his activities.

To test the temporal segmentation, we used just twenty days for the segmentation and we devoted the other days to tests. Results were more than satisfactory. As shown in Figure 3, which shows the activity intervals, the number of likely activities was reduced from 30 to 70% depending on the time of day.

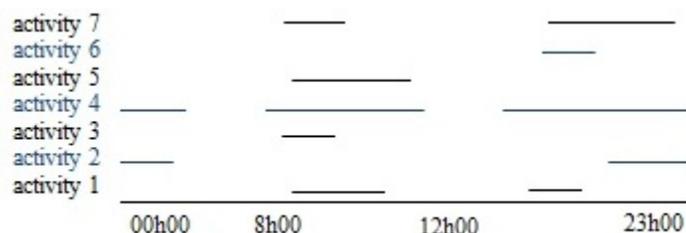


Figure 3. Representation of activities temporal intervals.

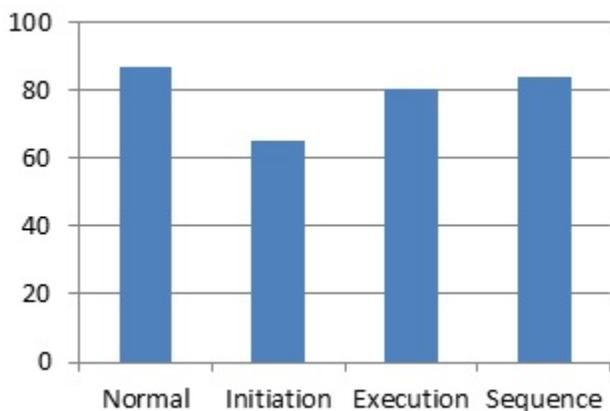
It should be noted that the number of activities that a person normally performs daily is well above the seven activities that were used in these experiments. This would necessarily increase the percentage of reduction of the number of likely activities. However, we must note another important problem in this segmentation. The perception of a single day as a time line gives the result that two times very close to each other seem to be very far. For example, 11:59 p.m. and 0:01 a.m. are considered to be almost 24 hours, not two minutes, apart. If we need to create an interval, then it will be 0:01 until 11:59 p.m., because the start time is always less than the end time, while the interval should be from 11:59 p.m. until 0:01 a.m. The difference between the two is enormous and will greatly affect our segmentation.

That is why the final step of segmentation, the audit, is very important because it will divide such an interval which took almost all day into two small intervals, giving a very similar result to that intended.

For the occupant of the smart home, who in our case is a person with Alzheimer’s disease, we have introduced some errors in the tests to simulate errors likely to be committed by the occupant, which are defined by Baum [11] as:  
 Execution errors: the patient forgets or adds actions that have nothing to do with the activity.  
 Sequence errors: the patient performs in a disorderly manner the various stages of the activity.

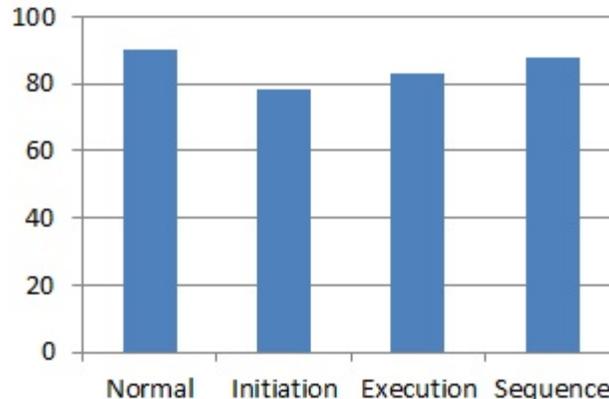
Graph 1 shows the percentage of detected activities. The column names have the following meanings:

- Normal: all the actions that compose the activity were done.
- Initiation: no action has been done.
- Execution: some irrelevant actions were added or deleted.
- Sequence: actions were not done in order.



Graph 1. Percentage of detected activities.

In general, the results were very satisfactory. In the first column, the activities that were not detected were started at an unusual time by the occupant. We are sure that the number of these activities would decrease if we observe the patient for a longer period. The third and fourth columns show that our system has responded well to execution and sequence errors. The decrease in activities recognized, in comparison with the first column, is explained by the similarity of certain activities with respect to starting times and component actions. In the second column, the relatively low percentage of activities recognized is mainly due to periods when the segmentation has reduced the number of likely activities by only 30%. The results of the second column are improved when we remove the interval of an activity that just has been recognized, so that it would not be considered again in the same period. Graph 2 presents the new results obtained.



Graph 2. Percentage of detected activities after removing the interval of an activity that just has been recognized.

## VI. RELATED WORK

Our approach explores the new possibilities offered by the emergence of techniques of data mining and exploits them in the field of activity recognition. It is mainly based on temporal segmentation, which plays a role in the feasibility of this approach and greatly affects the results. Temporal segmentation has been used in various studies of activity recognition, but never, to our knowledge, as described in this paper. In Spriggs et al. [14], for example, cameras are used to observe the patient. Then temporal segmentation divides the movement of the observed patient into actions, in order to understand and create movement patterns that will help recognize the activity. In other words, they try to break an activity into time intervals, where each interval represents an action. These actions will therefore consist of several frames of a video sequence, and they will be classified in order to create model movements that will be used for the future detection of activities. The results of this approach, according to Spriggs et al., were difficult to assess, but we can imagine the heavy processing of video and its impact on an application that needs to respond in real time. In addition, the use of cameras is still subject to debate because it does not preserve the privacy of the occupant.

Harvey et al. [15] have used segmentation to analyze the history of the work of developers. They see time as a sequence of points. Segmentation is to combine several consecutive points in order to create a segment. The goal is to reduce the number of points to a smaller number of segments. The only problem with this technique is that the segments do not perfectly represent the input data. The segments are created by trying to minimize errors, while in our approach the time intervals summarize all the input data.

Jakkula and Cook [13] used time to find relations between events. They defined 13 relations, like event 1 after

event 2 or event 1 finished event 2, etc. These relations helped them to predict an event when the usually previous event happened. They were then able to make many other interesting deductions, like: an event must end because it is followed by another event that just finished. The problem with Jakkula and Cooks work is that they apply their algorithm directly on the data warehouse, so that all deductions are about events and not activities. As an event may be included in many activities, we will have many relations for the same event, and we will be confused as to which relation we should apply. We still think that this algorithm is very interesting and may improve our activity recognition system. In fact, we could use it after our first step of activity pattern mining, when the relations are between activities and not events. After that, they may be converted as probabilities or priorities that can help calculate the most likely activity in our activity recognition system.

## VII. CONCLUSION AND FUTURE WORKS

In this study, we noticed the huge benefits of the use of data mining techniques in the area of activity recognition in smart homes for cognitive assistance to Alzheimer's patients. We have seen how these techniques were able to respond to different problems in this area. First we used them to create unsupervised and personalized model activities. Then they helped us reduce the number of likely activities by using the current time to eliminate the unlikely ones. They also allowed us to solve the problem of initiation errors of Alzheimer's patients by suggesting an activity to the occupant if the current time exceeds the end of an activity interval without the system detecting that the activity was performed. Finally, the complex problem of intersecting activities can be solved: if the occupant is used to perform several activities at once, upon detecting an activity, all the intersecting activities will be detected as a single activity, and our approach will therefore be able to assist him.

However, the success of this approach depends on a good temporal segmentation. If, for example, the occupant tries to perform an activity at a time outside its intervals, this approach will be unable to recognize the activity even if it is contained in the model activities.

Several improvements can be made to this approach, such as trying to have an average duration of the activity in order to recognize if the occupant has problems finishing it. We may even have a maximum time between two successive sensors in order to know when to trigger aid to the occupant. These improvements and others will be included in our next article.

## REFERENCES

- [1] J. William Frawley, G. Piatetsky-Shapiro, and J. Christopher Matheus, "Knowledge Discovery in Databases: An Overview", *Ai Magazine*, 1992.

- [2] F. John Roddick and M. Spiliopoulou, "A bibliography of temporal, spatial and spatio-temporal data mining research", *ACM SIGKDD Explorations Newsletter*, Volume 1 Issue 1, June 1999.
- [3] Fariba Sadri, "Ambient intelligence: A survey", *ACM Computing Surveys (CSUR)*, Volume 43 Issue 4, October 2011.
- [4] A. Vahdatpour, N. Amini, and M. Sarrafzadeh, "Toward Un-supervised Activity Discovery Using Multi Motif Dimensional Detection in Time Series", *IJCAI'09 Proceedings of the 21st international joint conference on Artificial intelligence*, 2009.
- [5] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan, "Detecting time series motifs under uniform scaling", *KDD '07 Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.
- [6] R. Michael Wick and J. Paul Wagner, "Using market basket analysis to integrate and motivate topics in discrete structures", *SIGCSE '06 Proceedings of the 37th SIGCSE technical symposium on Computer science education*, 2006.
- [7] R. Nizar Mabroukeh and C. I. Ezeife, "A taxonomy of sequential pattern mining algorithms", *ACM Computing Surveys (CSUR)*, Volume 43 Issue 1, November 2010 .
- [8] J. Wang and J. Han, "BIDE: Efficient Mining of Frequent Closed Sequences", 2007.
- [9] S R. Kannan and R. Pandiyarajan, "New modified fuzzy C-means for determination of proper structure in dataset", 2009.
- [10] A. Vattani, "k-means requires exponentially many iterations even in the plane", 2009.
- [11] C. Baum, D. F. Edwards, and N.Morrow Howell, "Identification and Measurement of productive behaviors in senile dementia of the Alzheimer type", 1993.
- [12] T. van Kasteren, A. Noulas, G. Englebienne and B. Krse, "Accurate Activity Recognition in has Home Setting", 2008.
- [13] R. Vikramaditya Jakkula and J. Diane Cook, "Mining Sensor Data in Smart Environment for Temporal Activity Prediction", 2007.
- [14] H. Ekaterina Spriggs, F. de la Torre, and M. Heber, "Temporal Segmentation and Activity Classification from First-person Sensing", *IEEE Workshop on Egocentric Vision, CVPR 2009*, June 2009.
- [15] H. Siy, P. Chundi, and M. Subramaniam, "Summarizing Developer Work History Using Time Series Segmentation", May 2008.