A New Illness Recognition Framework Using Frequent Temporal Pattern Mining

Zahra Hajihashemi
Computer Science Department,
University of Missouri
Columbia, MO 65201 USA
Zhmr5@mail.missouri.edu

Mihail Popescu
Health Management and Informatics Department,
University of Missouri
Columbia, MO 65201 USA
PopescuM@missouri.edu

Ubicomp '14 Adjunct, September 13 - 17, 2014, Seattle, WA, USA

http://dx.doi.org/10.1145/2638728.2638805

Abstract
Living alone in their own residence, older adults are at-risk for late assessment of physical or cognitive changes due to many factors such as their impression that such changes are simply a normal part of aging or their reluctance to admit to a problem. Sensors networks have emerged in the last decade as a possible solution to older adult health monitoring and early illness recognition. Typical early illness recognition approaches are either concentrated on the detection of a given set of activities such as a fall or walks, or on the detection of anomalies such as too many bathroom visits. In this paper we propose a new illness recognition framework, MFA, based on detecting a missing frequent activity from the daily routine. MFA is implemented using a frequent temporal pattern detection algorithm and demonstrated on a pilot dataset collected in TigerPlace, an aging in place community from Columbia, Missouri.

Author Keywords
Wireless sensor networks; Early illness recognition;

ACM Classification Keywords
G.3 Probability And Statistics (Time series Analysis);
H.2.8 Database Applications (Data Mining);

Mining frequent patterns from raw motion sensor data has difficulties such as multiple periodic behaviors, vast variations of a given periodic pattern due to different ways of doing the same thing, incomplete observations due to uneven sampling, interleaved action patterns in the real environment (such as walking while brushing teeth), and having large portion of missing data due to sensors malfunction.

Introduction

Older adults in the US prefer to live independently, despite the onset of conditions such as frailty and dementia. Elderly living alone are particularly at-risk for late assessment of physical or cognitive changes due to many factors: their impression that such changes are simply a normal part of aging; their reluctance to admit to a problem; their fear of being institutionalized; and even the failure of physicians to fully assess their function due to the belief that no intervention is possible [1].

Sensor networks are a promising solution for health monitoring of older adults [2]. Typical early illness recognition approaches are either concentrated on the detection of a given set of activities such as a fall or walks [3], or on the detection of anomalies such as too many bathroom visits [4]. The trajectory of typical functional decline in elderly is shown in Figure 1 (solid line) [5]. The curve has quasi-plateaus followed by sharp step-downs (solid line). The step-down are due loss of functional ability such as ability to shower, ability to dress, etc. Some step-downs are temporary (that is why we used the term "quasi-plateau" above) such as the ability to shower after having a leg injury, before they become permanent. Our goal with sensor-based health assessment is to stop the decline through early recognition and/or prediction of health problems in advance (dotted line). In this paper, we propose a new illness recognition framework based on the detection of a missing frequent activity (MFA) (as opposed to detecting an abnormal activity).

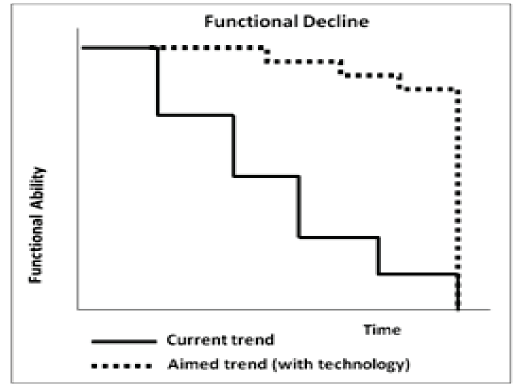


Figure 1. Trajectory of typical functional decline and the goal with early illness recognition.

Under this framework, at the end of the day, we examine if all frequent patterns found in previous two weeks are still present. If some patterns are found missing several days in a rows, we assume that it is due to an early onset of a decline and we send an alarm to caregiver. To detect the loss of an existent ability, we need to employ a frequent pattern detection method that is independent of the types and number of activities performed by the resident. Many techniques for detecting frequent patterns in time series have been reported in literature such as a combination of suffix tree and Markov model [6], a random projection algorithm [7], a rule based approach using likelihood criterion [8], FP-tree based approach using extended prefix-tree structure [9], fuzzy association rule mining approach [10]. Other frequent pattern detection work is related to activity recognition. For example, in [11] Bayesian networks are used to detect and predict the action time for 11 human activities using motion sensor data, while in [16] Latent Dirichlet Allocation is used for activity recognition using Kinect. However, most of these methods assume a certain number (fixed) of known activities. Moreover, in many cases, the training algorithm process requires data labeling, which is not feasible in a real environment. An implicit activity recognition approach based on sensor sequence similarity that combined sensor and electronic medical records (EMR) to provide early illness recognition was explored in [14]. One of the most used frequent pattern detection algorithm in behavior sciences is based on T-Patterns [12].

The frequent T-Pattern pattern (FTP) algorithm was first introduced by Magnusson in [12] to extract frequent human behaviors and improved in [13] for artificial sensor data. T-patterns are represented by symbols and time stamps. FTP algorithm finds possible relationships between pairs of symbols by building trees of temporal dependencies. The existent methods [12, 13] consider all possible combination of symbols to form patterns. This approach is potentially slow in sensor networks application for real time monitoring, especially for large number of sensors and long time line. In this paper, we adapt the FTP algorithm introduced in [12], for detection of frequent patterns in monitoring sensor networks and explore its suitability for MFA on a TigerPlace pilot data set. In the next section, we briefly describe our experimental setup and pilot dataset.

System Architecture and Dataset

With the University of Missouri IRB approval, we deployed sensor networks in 47 apartments from TigerPlace, an aging in place community from Columbia, Missouri [5]. In this paper, we validate our FTP algorithm using labeled bathroom visits in 10 days of sensor data. Our sequences contain 23 motion sensors deployed throughout the apartments. The pilot dataset used in this paper sample consists of three residents (see Table1) with different bathroom habits.

Resident ID	#1	#2	#3
Age	88	99	90
Gender	Male	Male	Female
Urinary Problem	Non	Retention	Incontinence
Has Walker	No	Yes	Yes
Diuretic Medication	Yes	No	No

Table 1. TigerPlace resident characteristics.

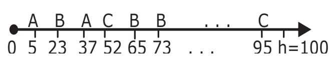
Table 2 shows an example of a bathroom visit as recorded by our in-home sensor network system. In this example, on December 5, 2005, around 12:30AM Resident #3 was in the bathroom for about 3 minutes. Table 3 shows the total number of known bathroom visits from the apartments of the three residents for the period of 10 consecutive days used in this study. Bathroom visits range in duration from few seconds to about 16 minutes.

Resident ID	Sensor	Time stamp
3	Bathroom	2005-05-12 12:34:38
3	Bathroom	2005-05-12 12:36:52
3	Bathroom	2005-05-12 12:37:04
3	Bathroom	2005-05-12 12:37:11
3	Closet	2005-05-12 12:37:26

Table 2. Example of sensor firings for a bathroom visit.

Resident ID	Number of bathroom visits	Number of shower visits
1	74	3
2	82	4
3	69	1

Table 3. TigerPlace pilot dataset.



Sensor	Time stamp
A	5, 37, …
B	23, 65, 73, …
C	52, … , 95

Figure 2. An example of occurrence tables.

Methods

In T-Pattern approach [12], Magnusson proposed the notion of a Critical Interval (CI) to find strongly correlated events (sensor firings). In this approach, all possible combinations of events are considered. The number of CI searches for a single pass of the T-pattern algorithm is $O(n^2h^2)$, where n is the number of event types (or sensors) and h is the event horizon (sequence length). Our goal is to modify this method such that it is linear in h, $O(n^2h)$, and be suitable for real time processing (we would like to find the pattern in real time as opposed to at the end of the day).

1. Data Representation

We define a time series S as a set of n couples $(s_i\ t_i)$, i.e. $S=\{(s_1\ t_1), \ldots, (s_n\ t_n)\}$. Each couple $(s_i\ t_i)$, $1 \leq i \leq n$, has two components: a sensor signal s_i that belongs to a symbols set Σ and a time stamp t_i which represents the time t_i when s_i was recorded. The alphabet Σ is a set of identifiers we use to represent multi-dimensional sensor time series. In our experiments, Σ comprises of three symbols as B (for bathroom sensor), C (for closet sensor), and A (for Shower sensor).

2. T-Pattern Representation

The T-Pattern is defined, recursively, as an ordered m triples, i.e. $T=\{(s_1\ [d_1,d_2]_1\ s_2), \ldots, (s_i\ [d_1,d_2]_i\ s_{i+1}), \ldots, (s_{m-1}\ [d_1,d_2]_m\ s_m)\}$. Each triples $(s_i\ [d_1,d_2]_i\ s_{i+1})$, $1 \leq i \leq m$, has three components: two sensor signals s_i and s_{i+1} that belongs to the symbols set Σ, and an interval $[d_1,d_2]_i$ which reflects different average values of occurences of the same pattern within the observation timeline. In this presentation the term $(s_i\ [d_1,d_2]_i\ s_{i+1})$ reads: when sensor s_i triggers between d_1 and d_2 time units later sensor s_{i+1} triggers.

3. Critical Intervals

For every frequent T-Patterns there is a CI that represents the relationship between the distributions of the elements of the T-Pattern, i.e. sensors [12]. For example, for a T-Pattern (s_i, s_j), CI $[d_1,d_2]$ indicates that if s_i occures at time t then there is an interval $[t+d_1,t+d_2]$ that tends to contain at least one occurrence of s_j.

4. Finding Frequent T-Pattern using CI

Inspired from KMP string search algorithm [15], we propose an efficient algorithm to find T-Patterns without the need exaoustive search of all sensor combinations. For the description of the original algorithm the reader is refered to [1]. Our proposed method has three main steps as described in the following.

First Step: Building Occurrence table

Considering the training observation interval $[0,h]$, the occurrence table is a data structure that indexes the firing time of each sensor in the observation interval. Assume we have n sensors, then the occurrence table have n rows each for one sensor. It takes just one pass through the whole dataset to build this matrix. So the time complexity of this step is *O(h)*. Figure 2 shows this step in an example.

SECOND STEP: FINDING COUPLE T-PATTERNS

Having the occurrence table, we search for 2-TPatterns (patterns of size 2, such as pattern "CB"). To find a 2-TPattern (s_i s_j), the algorithm starts at row *i* in the occurrence table. Then for each occurrence of s_i at time t_p it seeks for the first occurrence of s_j at time t_q such that $t_q>t_p$, where $1<p<N_i$ and N_i is the number of occurrences of s_i in the observation time, and $1<q<N_j$ and N_j is the number of occurrences of s_j in the observation time interval. Having two occurrences of the 2-TPattern (s_i s_j) at $[t_p,t_q]_1$ and $[t_p,t_q]_2$ the related CI is formed using the size of each occurrence that is $[\Delta d_1,\Delta d_2]$ where Δd_1 is the size of $[t_p,t_q]_1$ in time units. After finding all CIs, we sort them increasingly based on the size of CI. Then we perform a *p*-test to verify the significance of the CI. The *p*-value of CI $[d_1,d_2]$ is calculated using Eq.1.

$$p = \frac{(1-\sum_{k=0}^{N_{i,j}-1} binomial\left(N_{i,j},k,P\left(\sim s_j\right)\right))}{N_{i,j}} \quad (1)$$

Where $N_{i,j}$ is the number of occurrences of pattern (s_i s_j), and $P(\sim s_j)$ is the probability of absence of sensor s_j, $P(\sim s_j) = 1-(frequency(s_j)/h)$. Intuitively, if the CI of size Δd_1 is not significant, then the CI of size Δd_2 where $\Delta d_2>\Delta d_1$ is not significant either. The time complexity of this step in the worst case is *O(nh)* and in the best case is $O(n^2)$ where *n* is the number of sensors.

THIRD STEP: FINDING *K*-TPATTERNS

In this step in an iterative process we find T-Patterns of size *k*, called as *k*-TPatterns. For a given *(k-1)*-TPattern ($2>k$, $k \in \mathbb{N}$) with significant CI $[d_1,d_2]$, the next *k*-TPattern is built by looking up in the occurrence table for the first sensor event at time t_q where $t_q>t_p$ and t_p is when the last event when the *(k-1)*-TPattern happened. This step has the same time complexity as the second step. Figure 3 shows the pseudo code of Frequent T-Patterns (FTP) Algorithm.

Experimental Results

Dataset and Experiments Setup

To evaluate FTP performance in finding frequent activity we used the 3 resident pilot dataset where the bathroom visits were labeled. We note that labeling is not necessary in our MFA framework: if the activity is frequent, we don't need to know its name.

Frequent T-Patterns Algorithm (FTPA)

Build occurrence table;
K=2;
Find couple T-Patterns with their significant CIs;
While(termination condition){
    Find K-Tuple T-patterns form
        (k-1)-Tuple T-Patterns set;
   k= k+1;
}
Return the set of T-patterns with their CIs;

Figure 3. Frequent T-Patterns (FTP) pseudo code.

First, we run our FTP algorithm on a training sensor dataset to extract all T-Patterns of size *k* ($2 \le k \le d$, where there is no significant CI for pattern of size d+1) with their significant CIs and save them in a dictionary of T-Patterns. Then for a given test sensor sequence, we apply a top-down approach such that we find the longest T-Pattern first that matches with one of T-patterns in the dictionary. If the interval of pattern of size *k* does not belong to the CI from the dictionary, then we look for T-Pattern of size (*k*-1), where *k*>=3 (see Figure *4*).

Frequent pattern detection using FTPA
Step1. Build dictionary of T-Patterns using FTPA and sensor data. K=d; Step2. If k>=2, then look for pattern of size k in the given sensor sequence such that their CI matches. Step 3. If nothing found, then k=k-1, and repeat Step2.

Figure 4. Finding frequent patterns using FTPA algorithm.

Evaluation metrics
To evaluate the performance of the proposed method we use *sensitivity* and *specificity* as defined bellow.

$$sensitivity = \frac{TP}{TP+FN} \quad (2)$$

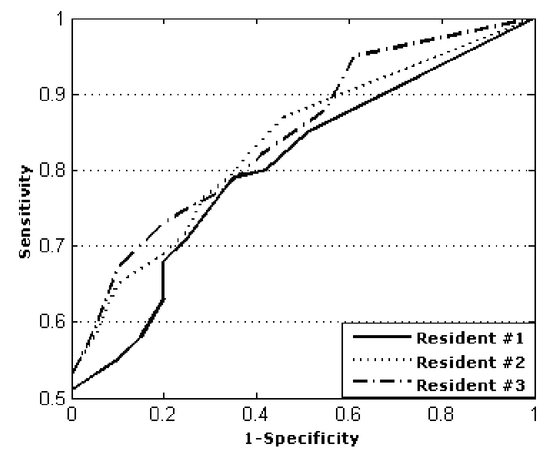
$$specificity = \frac{TN}{TN+FP} \quad (3)$$

In Eq.1 and Eq.2 *TP* is the number of true positives, *TN* is the number of true negatives, *FP* is the number of false positives, and *FN* is the number of false negatives.

Bathroom Activity Recognition Using T-Patterns
Table 4 presents the result of activity recognition using a 10 cross-fold validation approach*.* Figure 5 shows the results in a ROC curve as well as the area under the curve (AUC) separately for each of the residents. The average recognition performance is 0.83. Resident #2 and #3 use a walker that affects their speed reflected in the sensor hits. Therefore, the extraction process has a lower sensitivity, compared to Resident #1.

Resident ID	Sensitivity	Specificity
#1	0.85	0.76
#2	0.83	0.69
#3	0.83	0.71

Table 4. Frequent pattern detection perfromance.



AUC #1	AUC #2	AUC #3
0.80	0.83	0.84

Figure 5 . Roc Curves and area under ROC curve for frequent pattern detection using FTPA.

Resident #1 is the youngest and most active, which allows him to have a less rigid routine, with bathroom visits that are more varied in time. Also he is the only one taking a diuretic medication which in turn may affect the number of bathroom visits. Both Residents

#2 and #3 use walkers that may force them to have more regimented and planned bathroom trips.

Conclusions

We described a new early illness recognition framework, MFA, based on frequent pattern analysis. Since MFA is based on a frequent pattern analysis method, we explored a modified version of the T-Pattern algorithm, FTP as a possible candidate. On a pilot dataset, our modified T-Pattern algorithm had an average recognition rate of bathroom visits of about 0.83 with about 0.3 false alarm rate, which makes it a reasonable candidate for our MFA framework.

However, the complete implementation of the MFA framework will be performed in future work. Aside of FTP algorithm, it will include the activity similarity measure developed in [14] together with a relational clustering technique.

Acknowledgements

This project has been funded by a NSF SHWB grant, award #IIS-1115956 and the John A. Hartford Foundation's NHCGNE Award Program.

References

[1] Hayes, TL., et al. An unobtrusive in-home monitoring system for detection of key motor changes preceding cognitive decline. *IEEE EMBS*, 2004, 2480-2483.

[2] Mack, D., et al. A passive and portable system for monitoring heart rate and detecting sleep apnea and arousals: Preliminary validation. *D2H2*, 2006, 51-54.

[3] Li, Y., et al. "Efficient source separation algorithms for acoustic fall detection using a Microsoft Kinect", *IEEE TBE*, vol. 61, no. 3, 2014, 745-755.

[4] Popescu, M., Mahnot, A. Early Illness Recognition in Older Adults Using In-Home Monitoring Sensors and Multiple Instance Learning, *Methods Inf. Med.*, 2012, 359-67.

[5] Rantz, MJ., et al. "A Technology and Nursing Collaboration to Help Older Adults Age in Place", *Nursing Outlook*, vol. 53, no. 1, 2005, 40-45.

[6] Keogh, E., et al. Finding surprising patterns in a time series database in linear time and space. *SIGKDD*, 2002, 550-556.

[7] Kalpakis, et al. Distance measures for effective clustering of ARIMA time-series. *IEEE ICDM*, 2001, 273-280.

[8] Tompa, M. Finding motifs using random projections. *Journal of Comput Biol.* 2002,225-242 .

[9] Guralink, V., Srivastava, J. Event detection from time series data. *ACM SIGKDD*, 1999, 33-42.

[10] Han, J., Pei, j., et al. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Dicov*, 2004, 53-87.

[11] Nazerfard, E., Cook, D. Using Bayesian Networks for Daily Activity Prediction, *AAAI*, 2013, 32-38.

[12] Magnusson, M. Discovering hidden time patterns in behavior: T-pattern and their detection. *Behavior Research Methods,* 2000, Vol. 32, 93-110.

[13] Salah, A., Pauwels, E., et al. T-Pattern Revisited: Mining for Temporal Patterns in Sensor Data. *Sensors (Basel)*, 2010, 7496-7513.

[14] Hajihashemi, Z., Popescu, M. An Early Illness Recognition Framework Using a Temporal Smith Waterman algorithm and NLP, *AMIA*, 2013, 548–557.

[15] Knuth, D., et al. Fast pattern matching in strings. *SIAM Journal on Computing*, 1977, 323–350.

[16] Zang H, Parker L, "4-Dimensional Local Spatio-Temporal Features for Human Activity Recognition", Proc. of IEEE International Conference on Intelligent Robots and Systems, San Francisco, CA, 2011.