Discovering Rules of Subtle Deficits Indicating Mild Cognitive Impairment Using Inductive Logic Programming

Keisuke Abe[1](✉), Niken Prasasti Martono[1], Takehiko Yamaguchi[2], Hayato Ohwada[1], and Tania Giovannetti[3]

[1] Departement of Industrial Administration, Tokyo University of Science, Katsushika, Japan
7416603@ed.tus.ac.jp, niken.prasasti@sbm-itb.ac.id, ohwada@rs.tus.ac.jp

[2] Department of Applied Electronic, Tokyo University of Science, Katsushika, Japan
tk-ymgch@te.noda.tus.ac.jp

[3] Department of Psychology, Temple University, Philadelphia, USA
tgio@temple.edu

Abstract. Recently, Japan has been experiencing a declining birthrate and an increasingly aging population; as a result, the number of dementia patients is increasing. Current medical science has no way to treat dementia completely after onset. Therefore, it is necessary to detect mild cognitive impairment (MCI) in the early stage just before dementia develops. It is clear that MCI patients who exhibit subtle deficits in daily living behavior (in this study, micro-errors (MEs)) have declining cognitive function associated with cognitive impairment. Virtual reality (VR) technology has been actively utilized in rehabilitation and therapy, and here we use an application known as Virtual Kitchen (VK). In this work, we analyze how ME happens. We use finger movement data and subtask information from VK. Our methodology proposes a combination of inductive logic programming (ILP) and the sliding window algorithm. Because ILP can extract expressive rules but is susceptible to noise and memory hog, it is difficult to use sensor data directly for learning. Sliding window is used as its ability to reduce the amount of data while holding the shape of original time series data. From preliminary experiments, we obtained some rules of ME occurrence that are related to differences in speed, time interval, and subtask. We obtained results that explain how ME occurrence is generally related to subtask and finger speed. In the future, we will use more positive samples and conduct more experiments to obtain better and more accurate results.

Keywords: Cognitive impairment · Virtual reality · Time series · Data mining

1 Introduction

Recently, Japan has been experiencing a declining birthrate and an increasingly aging population; as a result, the number of dementia patients is increasing. Current medical science has no way to treat dementia completely after onset. Therefore, it is necessary

S. Yamamoto (Ed.): HIMI 2017, Part II, LNCS 10274, pp. 331–340, 2017.
DOI: 10.1007/978-3-319-58524-6_27

to detect mild cognitive impairment (MCI) in the early stage just before dementia develops. If treatment or prevention can be done early in the MCI stage, there is a possibility that dementia will not develop. However, MCI is diagnosed subjectively by a doctor. Recent quantitative research indicates that MCI patients who exhibit subtle deficits in daily living behavior (in this study, micro-errors (MEs)) have declining cognitive function associated with cognitive impairment. Sara et al. demonstrated that there was a high possibility that participants experiencing a large number of MEs had lower cognitive function [1].

In recent years, virtual reality (VR) technology has been actively utilized in rehabilitation and therapy, and VR applications corresponding to human physical and cognitive functions have been developed [2]. In this study, we use an application known as Virtual Kitchen (VK) [3] to analyze how ME occurs. VK is used as an application for participants to perform daily living tasks, such as making breakfast, in virtual space [4]. VK is equipped with a leap motion sensor, and we are able to record finger movements to obtain useful information.

This study utilizes finger movement data and subtask information from VK to analyze how ME occurs. First, we performed data smoothing to reduce sensor noise. Second, we segmented speed data according to subtask completion time. Third, we employed the sliding window algorithm to use Inductive Logic Programing (ILP). Finally, we used ILP to visualize rules and to use qualitative and quantitative data.

2 Finger Movement and VK Data

2.1 Raw Data

The VK system is equipped with a leap motion sensor to collect finger movements and provide two-dimensional coordinate data. This sensor records data at 0.01 per second and contains much noise; thus, data preprocessing is necessary. We performed data smoothing for each dimension using a moving average filter to make data trends easier to understand. Figure 1 plots the results of moving average filter using a span of 50 points. We then calculated speed data from the preprocessed data.

We obtained finger movement data and VK information data that included the state of the finger when touching the screen of the VK application. Thirteen healthy young adults, the pilot participants, prepared breakfast as everyday action tasks in the VK.

2.2 VK Data

Breakfast tasks analyzed in this work consist of preparing toast and making coffee (Table 1). Toast preparation subtasks include putting the toast in the toaster, switching on the toaster, putting the toast on the plate, taking the butter and jelly with knife, and spreading the butter and jelly on the toast. Coffee making subtasks include opening the coffee lid, scooping the coffee, putting the coffee in the mug, opening the sugar lid, scooping sugar with a spoon, stirring the coffee, putting in milk, and stirring the coffee. Figure 2 depicts the configuration of the VK system and the leap motion devices.

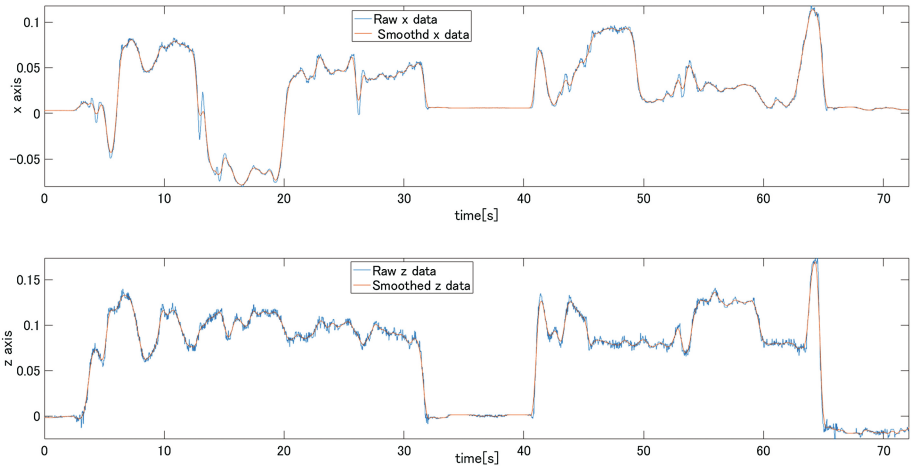

Fig. 1. Result of moving average filter

Table 1. Subtasks of toast and coffee preparation

Variable	Description	Variable	Description
Toast 1	Putting the toast in the toaster	Coffee 1	Opening the coffee lid
Toast 2	Switching on the toaster	Coffee 2	Scooping coffee
Toast 3	Putting the toast on the plate	Coffee 3	Putting coffee into mug
Toast 4	Taking the butter with knife	Coffee 4	Opening the sugar lid
Toast 5	Spread the butter on toast	Coffee 5	Scooping sugar with a spoon
Toast 6	Taking the jelly with knife	Coffee 6	Stirring the coffee
Toast 7	Spread the jelly on toast	Coffee 7	Putting in milk
		Coffee 8	Stirring the coffee

2.3 Segmentation by Subtask

In this study, we segmented completion time by subtask and used segmented data as sample data for the data-mining method. The time frame in Fig. 3 represents finger speed until completion of the breakfast task. For example, the leftmost frame separates the speed of the fingers until finishing subtask toast 1. The yellow segments indicate that ME was observed during the subtask.

We found 45 positive examples that represent ME events in the subtasks and 168 negative examples of events other than ME.

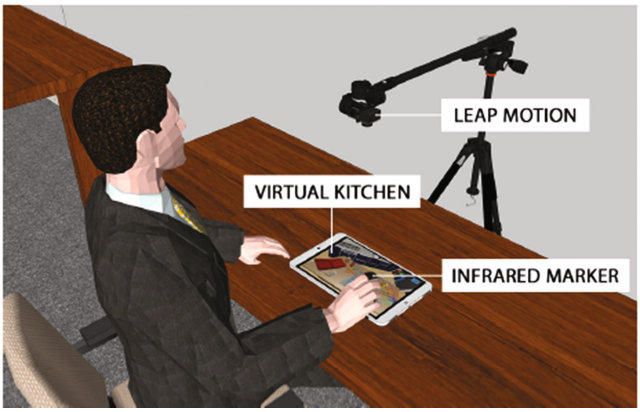

Fig. 2. System configuration of VK application [3]

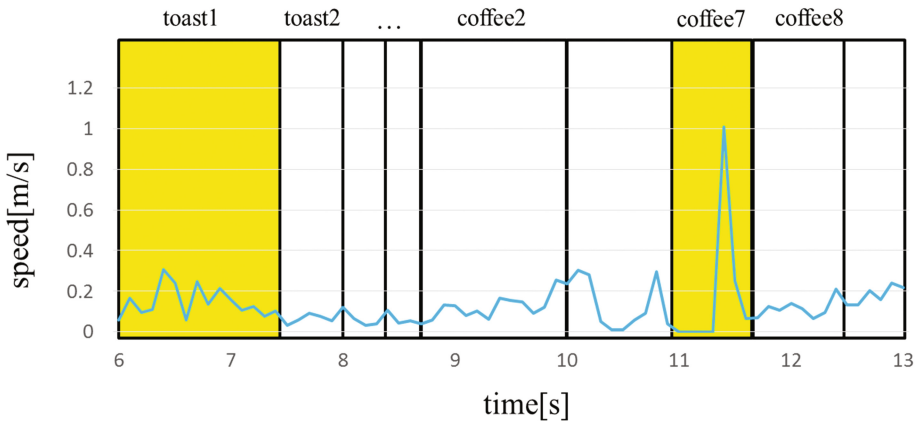

Fig. 3. Segmentation

3 Method

Our methodology proposes a combination of inductive logic programming (ILP) and the sliding window algorithm. Because ILP can extract expressive rules using background knowledge but ILP is susceptible to noise and memory hog, it is difficult to use sensor data directly for ILP. Sliding window is used to reduce the amount of data retaining shape of original time series data. The details are discussed in the next section. To use ILP, we must define some background knowledge that is obtained from finger speed and subtask information: speed, difference in speed, time intervals, the current subtask, the previous subtask, and the next subtask. In the final process, we use ILP to extract rules covering positive data. We used the parallel ILP system known as GKS [5, 6].

3.1 Sliding Window Algorithm

The sliding window algorithm is a piecewise linear approximation algorithm that is often used for machine learning and data mining for preprocessing sensor data [7]. In this study, it is used as it has the ability to reduce the amount of data while holding the shape of original time series data. We used the sliding window algorithm with interpolation against each segmented data [7]. Figure 4 plots the result of the sliding window algorithm with interpolation to reduce noise.

In this study, sliding windows are used on the original finger speed data in an effort to produce the smallest number of time points that will be used in ILP background knowledge.

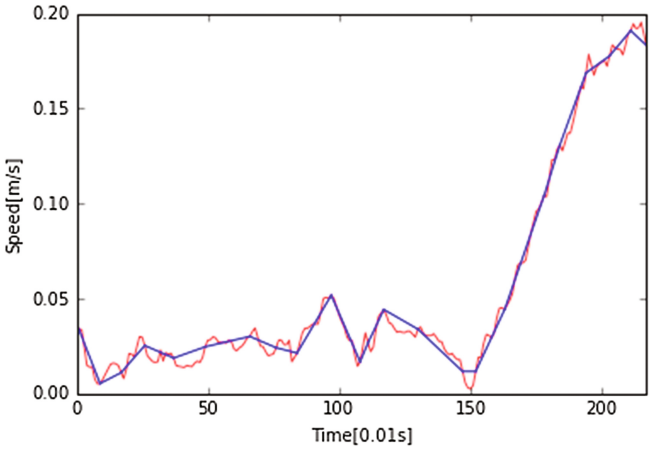

Fig. 4. Example of sliding window result with interpolation

3.2 ILP

ILP is a machine learning method that uses logic as the input. This method can extract rules explaining positive samples but not negative samples, and is a method to express time series data using flexible background knowledge. We defined two categories of predicate: movement and subtask. Table 2 lists predicates of ILP background knowledge. The movement category represents the quantitative data related to finger speed. The speed predicate represents the speed at a certain point in time. The diff_span predicate has two constants that exhibit difference between speed at a certain time point and speed at the adjacent time point, and time intervals between them. The subtask category includes predicates that explain qualitative data. We defined current subtask, previous subtask, and next subtask. By combining finger movement and subtask information, we can determine trends in type of subtask and type of movement when ME occurs.

ILP cannot deal with a continuous value; thus, we must transform data into discrete values. Table 3 lists descriptions of data transformation. The + symbol denotes an

Table 2. Predicates and mode declarations in background knowledge

Type	Predicate
Movement value	speed (+time, #value)
	diff_span (+time, -time, #value, #value)
Subtask	current_subtask (#value)
	previous_subtask (#value)
	next_subtask (#value)

Table 3. Descriptions of transformed values

#value	Range	Definition
speed	value < 0.02 [m/s]	low
	0.02 [m/s] ≦ value < 0.06 [m/s]	middle
	0.06 [m/s] ≦ value	high
span	value < 0.2 [sec]	short
	0.2 [sec] ≦ value < 0.5 [sec]	middle
	0.5 [sec] ≦ value	long
diff	Categorize the positive values of speed difference (increasing) into three levels	high_up
		middle_up
		low_up
	Categorize the positive values of speed difference (decreasing) into three levels	high_down
		middle_down
		low_down

input variable, the – symbol indicates an output variable, and the # symbol denotes a constant. We transformed speed into high, middle, and low. We transformed the difference value into three stages so as to be equal frequency, depending on whether it was positive or negative. We transformed time intervals into long, middle and short.

For example, in diff_span, we express change of speed as

```
speed(1,low)
diff_span(1,2,high_up,short)
diff_span(2,3,low_down,middle).
```

The upper expression indicates that time series point 1 is low speed. The middle expression indicates that from time series point 1 to time series point 2 there is a high speed increase at short time intervals. The second expression indicates that from time series point 2 to time series point 3 there is a speed decrease at medium time intervals. We can express the time series change as described above.

4 Results

As a result, we obtained rules using the ILP learning result. "{T, F}" means the number of positive examples and the number of negative examples explained by the rule. The obtained results from ILP are as follows.

Rule 1:

```
{5,0}+pos(A) :- speed(A,B,low), diff_span(A,B,C,high_up,
short), diff_span(A,D,B,middle_down,short),
diff_span(A,E,D,high_up,middle)
```

ME occurs when there is a high speed increase in a short time interval from the middle point, a moderate decrease occurs at a short time interval, and speed rises greatly in short time intervals.

Rule 2:

```
{7,2} +pos(A) :- speed(A,B,middle), diff_span(A,B,C, mid-
dle_up,short), diff_span(A,D,B,middle_down, middle),
diff_span(A,E,D,high_up,middle)
```

ME occurs when there is a moderate speed decrease at a middle time interval from the middle point, speed increases moderately in a short time interval, and then it increases greatly in a middle time interval.

Rule 3:

```
{6,2} +pos(A) :- diff_span(A,B,C,middle_down,short),
previous_subtask(A,toast2), diff_span(A,C,D,middle_up,
short)
```

ME occurs when there is a moderate speed decrease at a short time interval from the previous time series point; then it increase moderately at short time interval after performing subtask toast 2.

Rule 4:

```
{5,2} +pos(A) :- speed(A,B,middle), next_subtask(A,
toast3), diff_span(A,B,C,high_up,middle),
diff_span(A,C,D,middle_down, middle)
```

ME occurs when there is a high speed increase at a short time interval from the middle point; then speed decreases moderately at a middle time interval before performing subtask toast 3.

Rule 5:

```
{5,1} +pos(A):- speed(A,B,high), current_subtask(A, cof-
fee2),diff_span(A,B,C,middle_down,middle),
diff_span(A,D,B,high_up,short)
```

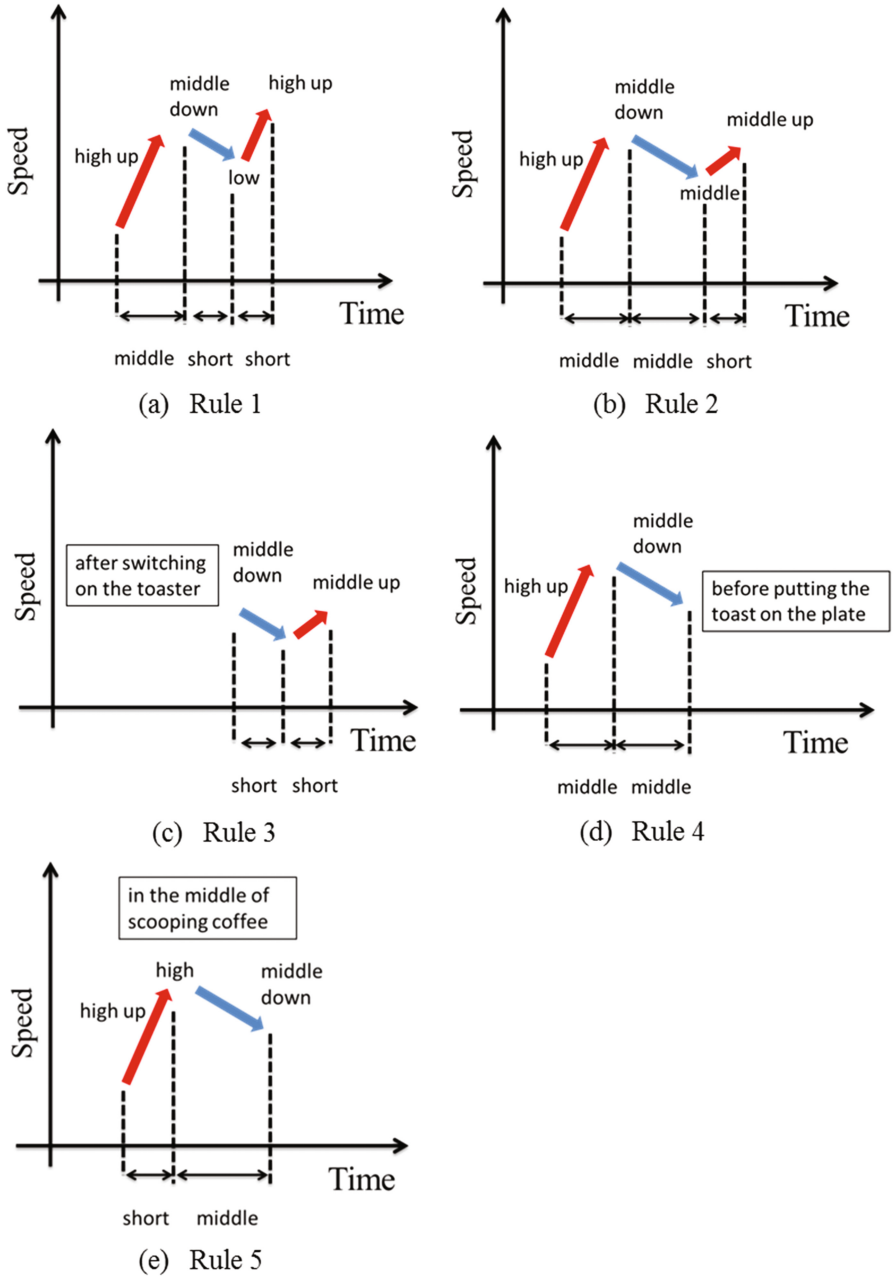

Fig. 5. Result of rules

ME occurs when there is a high speed increase at a short time interval, and then speed decreases moderately from the low point in a short time interval in the middle of subtask coffee 2.

Figure 5 presents these results in a time frame.

5 Discussion

From the preliminary experiments, we obtained some rules related to speed change and time interval. Rule 1 and Rule 2 indicate that ME tends to occur when speed rapidly increases and decreases in a short time interval. Also, regarding subtasks, Rule 3 and Rule 4 indicate that ME occurred after toast 2 (when the participant changed from toast task to coffee task) and before toast 3 (when the participant changed from coffee task to toast task). These results indicate that ME occurs during changing of subtasks, when there is much cognitive load. Rule 5 indicates that ME occurred in the middle of performing coffee 3.

Some MEs occur when there is no change in finger speed. Therefore, MEs can be roughly divided into two groups: those that involve speed change and those that involve no speed change. In the group that involves speed change, speed tends to increase and decrease in a short time interval, and ME occurs at the time of task switching.

However, in this study, the number of positive examples and the number negative example are unbalanced, because the amount of positive data was small and only a few samples were covered by the rule. In the future, we will work with more samples to obtain better results.

6 Conclusion and Future Work

This study used ILP to extract rules for subtle deficits occurring during performance of cognitive tasks in a VR environment. We arranged raw data and transformed data into discrete values for use in ILP. We defined background knowledge such as speed, difference, time interval, and subtask; then we used ILP to learn rules that define the occurence of ME. We obtained rules regarding finger movement and subtask in relation to ME occurence. ME tends to occur when movement changes in a short time interval (less than 0.5 s) and when the participant switches from one task to another. We obtained some preliminary results that explain how ME is generally related to subtask and finger speed. However, we had only a small amount of positive data in this pilot experiment. In the future, we will use more positive samples and conduct more experiments to obtain better and more accurate results.

References

1. Seligman, S.C., Giovannetti, T., Sestito, J., Libon, D.J.: A new approach to the characterization of subtle errors in everyday action: implications for mild cognitive impairment. Clin. Neuropsychologist **28**(1), 97–115 (2014). https://doi.org/10.1080/13854046.2013.852624

2. Faria, A.L., Andrade, A., Soares, L., i Badia, S.B., Langhorne, P., Bernhardt, J., Hsieh, C.: Benefits of virtual reality based cognitive rehabilitation through simulated activities of daily living: a randomized controlled trial with stroke patients. J. NeuroEng. Rehabil. **13**(1), 96 (2016). https://doi.org/10.1186/s12984-016-0204-z
3. Foloppe, D.A., Richard, P., Yamaguchi, T., Etcharry-Bouyx, F., Allain, P.: The potential of virtual reality-based training to enhance the functional autonomy of Alzheimer's disease patients in cooking activities: a single case study. Neuropsychol. Rehabil. 1–25 (2015). https://doi.org/10.1080/09602011.2015.1094394. November 2011
4. Martono, N.P., Yamaguchi, T., Ohwada, H.: Utilizing finger movement data to cluster patients with everyday action impairment. In: 2016 IEEE 14th International Conference on Cognitive Inlormatics & Cognitive Computing, pp. 459–464 (2016). https://doi.org/10.13140/RG.2.1.2084.5684. (August)
5. Mizoguchi, F., Ohwada, H.: Constrained relative least general generalization for inducing constraint logic programs. New Gener. Comput. **13**(3–4), 335–368 (1995). https://doi.org/10.1007/BF03037230
6. Nishiyama, H., Ohwada, H.: Module … Module Worker. In: CEUR Workshop Proceedings of the Late Breaking Papers of the 25th International Conference on Inductive Logic Programming, pp. 86–94 (2015)
7. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An online algorithm for segmenting time series. In: Proceedings of the 2001 IEEE International Conference on Data Mining, pp. 289–296 (2001). https://doi.org/10.1109/ICDM.2001.989531