

# 並行に実行されるタスクを考慮した タスク検証システム

中島丞<sup>†1</sup> 大村廉<sup>†2</sup>

**概要:** 実環境で行われる業務や活動(タスク)は、効率化や安全性の確保のため、その遂行手順がある程度決められていたり、推奨される手順が存在するものが多い。システムが人間の動きを監督し正確な手順の教示や不正な行動の警告を行うことはタスク遂行の効率性、安全性の向上に有益である。しかし、既存研究ではユーザの行動を認識してその行動に対する適応的なサービスを目的とするものが多く、ユーザの行動の適切さを検証することを目的としたものはあまり多くない。特に、同時に複数のタスクを並行して実行する可能性のある状況において、それぞれのタスクの適切さを検証する研究は十分に行なわれていない。そこで、本研究では実環境においてユーザの行動がタスク遂行の上で適切なものであるかをリアルタイムに検証するシステムを構築する。本稿における実環境とは、多様なタスクが発生し得ると共にユーザがそれらを並行して進める可能性のある環境を示す。提案手法では、タスクをユーザの行動で遷移する状態系列として定義し、その遷移を3種類に分類した。また、ユーザが並行して複数のタスクを実行する場合に対応するため、実行される可能性のあるタスクを複数同時に評価し、遷移間で優先順位を設定することでタスク検証を行った。また、このままでは多様なタスクが発生し得る場合、評価するタスク数の増加によって計算量の増加や検証精度が低下してしまうため、それぞれのタスクの実行尤度を算出し、検証を行うタスクモデルを選出する手法を導入した。評価実験として一般家庭における家事を提案手法で検証し、最大で0.517の適合率と0.767の再現率を示した。

## Verification Methods for Concurrent Multiple Tasks

TASUKU NAKAJIMA<sup>†1</sup> REN OHMURA<sup>†2</sup>

**Abstract:** Many of real tasks has decided or recommended execution procedure in order to ensure the efficiency and safety. Validating human activity and instructing correct behavior are effective for users to accomplish their task efficiently and safety. However, many existing studies focus on adaptive services based on simple recognized activities, and few studies focus on verifying appropriateness of user's current activity sequence, especially in a situation where a user conduct multiple tasks concurrently. Thus, this study proposes a real-time verification method that verifies user's activity sequence. The proposed method can deal with a situation where many tasks are possibly conducted by a user concurrently. In our study, a single task is defined as an ordinal state machine whose state transitions are caused by a user's activity, and the transitions are categorized into three types. On the verification, tasks possibly conducted are evaluated simultaneously with priorities assigned on the transition type. In this method, increasing a number of evaluated tasks gets significantly higher computational load and lower verification accuracy, and a selection filter that decreases the number of verification target tasks by selecting the tasks based on their occurrence likelihood. In order to evaluate proposed method, experiments were conducted with daily life activities. The results showed that proposed task models achieved at most 0.517 precision and 0.767 recall.

### 1.はじめに

近年ではスマートフォンに代表されるようなセンシング能力を持つ小型スマートデバイスが発達し、ユーザが身につけたデバイスによりユーザの生活をサポートするようなウェアラブル・コンピューティングシステムの研究開発が多く行われている。その中で各種センサデータに基づき人の行動を自動認識する行動認識技術は重要な技術の一つであり、数多くの研究がおこなわれている [1-3]。行動認識を使ってサービスを提供する研究として、行動ログの取得 [4, 5], 調理進行の補助 [6], 高齢者の生活サポート [7, 8], ヘルスケア [9, 10], スキルアセスメント [11, 12], などが提案されている。

これらの既存研究は行動認識の結果を受理して支援を

行うことに注力されていた。しかし、現実には人間は業務の実行手順を間違えたり、危険な行動を取ったりしてしまうものである。既存研究ではユーザが誤った行動をとったとしても、それを指摘し、是正することに着目した研究はあまりなかった。より実用的なサポートを行なうためには、ユーザの行動が行なっている行動がその時行なっているタスク（登校の準備や味噌汁を調理といった、目的を達成するために行われる一連の行動系列）に適したものであるかどうかを検証できなければならない。例えば、看護師の業務では薬品の取り扱いや器具の操作を間違えることは大きな危険へと発展しやすく、システムがそれらを指摘することで未然に防ぐことができれば医療サービスの安全性の向上に貢献できる。このように、業務（タスク）遂行の正確性や効率、安全性を向上させるためには、ユーザの誤りを

<sup>†1</sup> 豊橋技術科学大学大学院  
Graduate School of Toyohashi University of Technology

<sup>†2</sup> 豊橋技術科学大学

指摘することが必要となる。また、実環境ではユーザは複数の業務を抱え込み、それらを並行して進めることも多い。実用的なシステムを実現するためには、多様な業務内容が並行して進行している環境下であっても行動の検証ができるものでなければならない。タスクを認識するための手法はいくつか研究されているが [3-5]、タスクの遂行を検証する方法についての研究はあまり行われていない。

そこで、本研究では日常生活や業務などのサポートを目的とし、実環境においてユーザの行動がタスク遂行の上で正しいかどうかをリアルタイムに検証する手法を提案する。本稿において想定する実環境とは、多様なタスクが発生する可能性があり、ユーザは複数のタスクを並列して進められる環境を想定する。本研究では、提案手法を実現するにあたり、1)タスクを行動認識器のラベルを用いた状態系列として定義すること、2)並行して複数のタスクを進められる環境下で検証を行えるシステムを構築すること、3)検証精度の向上および計算量の増大を抑えるために検証対象となるタスクを動的に選択すること、の3点を行った。また、本手法の有効性を確認するため、一般家庭にて実際の家事の遂行を検証した。

以下、本論文では、2章において関連研究の紹介と実用的なタスク検証システムの要素について明確化し、3章において提案手法の設計について上記の3項目をその意図と共に述べる。4章では評価実験とそれによる提案手法の性能評価について述べ、5章では本論文をまとめる。6章にて今後の課題と方向性について述べる。

## 2. 関連研究

本研究が目指すタスク検証システムは、ユーザの行動をリアルタイムで認識し、タスクを遂行する上でのその正誤を検証できるものである。システムはユーザがタスクを達成することができるように、行動を誤った場合にそれを指摘できなければならない。特に、実環境においてユーザが行なうタスクは単一ではなく、複数のタスクを抱えこみ、それらを適宜切り替えながら作業することが多い[4, 6, 13]。単一のタスクのみを想定したシステムでは、複数のタスクが並行して行われる場合に対応できない [14]。本研究では、適切なサポートを提供するため、そのように並行して行なわれるタスクの検証を考慮する。

行動のモニタリングは行動認識を用いる主要な研究の一つである [4, 5, 7, 13, 15]。例えば、体に付けた3つの加速度センサと RFID タグから 18 種の行動を 95%の精度で識別する研究がある [16]。この研究では、識別された行動に基づいてカロリー消費を計算し、ヘルスケアに役立てている。また、スマートフォンに内蔵された加速度センサ、マイクを用いて3種の行動を識別してユーザの生活を追跡する研究がある [17]。この研究では行動ログを分析し、より

良いライフサイクルの提示を行っている。また、ウェアラブルマルチセンサボードから 651 の特徴量を収集し、HMM を利用した特徴量選択によって 10 種の行動を 95%の精度で識別する研究がある [18]。[18]では、最終的に行動認識システムのスマートホームや民族誌学での利用を目指している。このようにウェアラブルセンサによる高精度な行動認識が多く研究されており、その実用性も増してきている。

ユーザの行動を検証する上ではユーザのタスクを識別することは有益である [5, 8, 19]。粒度の細かい情報を基にして粒度の粗いタスクの推定へと繋げる手法はいくつか研究されている [1, 3, 4, 8, 20]。例えば、[4]では加速度センサから点滴混注や点滴作成などを識別し、点滴業務というタスクを把握する手法が提案されている。[19]では3種類のセンサ、加速度センサ、モーションセンサ、スマートスイッチを用いて、高齢者の日常生活行動が事前定義されたルーチンワークに一致するかを確認する手法が提案されている。しかし、これらの研究ではユーザが一度に遂行するタスクは常に一つずつだと仮定されており、ユーザが複数のタスクを適宜切り替えながら作業するような場合に対応することができない。

一方で、複数のタスクが並行して実行されることを考慮した研究もある。[5]ではタスクマネジメントの補助を目的とし、目の動きからユーザの知覚的負荷や認知的負荷を推定し、それを基にタスクの切り替えを識別する手法が提案されている。[6]ではAR マーカーや画像処理を用い、複数の品目の料理を並列に調理する場合に、調理中の料理を判別する手法が提案されている。しかし、これらの手法は実行中のタスクを推定するのみであり、ユーザの行動の誤りは考慮していない。そのため、ユーザがタスクの遂行を誤った場合には、タスクの識別が不正確になる。タスクの識別が不正確な場合、行動の検証精度にも影響する。さらに、これらの手法は予め定義した複数のタスクを網羅的に評価することで実行中のタスクを推定している。しかし、複数のタスクを検証する場合には各タスク間で検証結果が競合する可能性を考慮しなければならない。例えば、スクランブルエッグの調理中に、タスクを切り替えて別の品目の調理を進行しようとしているとする。スクランブルエッグの調理が卵を溶いている段階において、ユーザが別のタスクに移り、別の内容の作業をしたとしても大きな問題とはならない。しかし、スクランブルエッグの調理において、卵をフライパンへ注いだ段階ならば、卵を素早くかき混ぜなければならず、別のタスクを進行しようとする行動は制止されるべきである。このように複数のタスクを並行して進める場合には、実施中のタスクの進行具合や、その内容(実施中の内容とこれから実施する内容)の兼ね合いによってその妥当性を判断しなければならない。

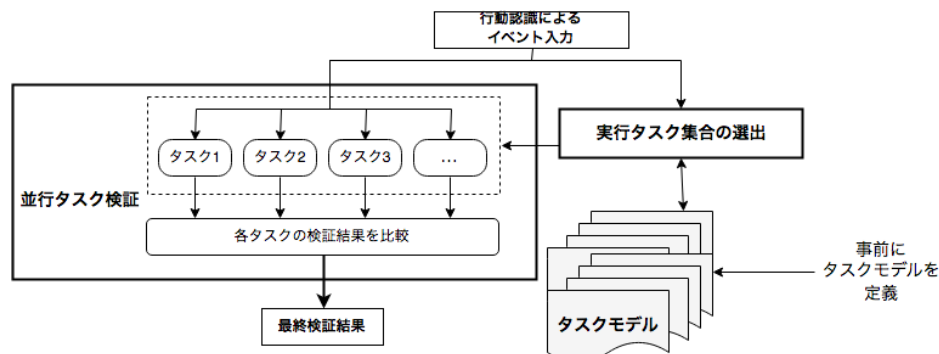


図 1 提案手法の概要

また、モニタリング手法の多くは後日記録されたデータを本人、または専門家が確認し、目的に合わせて分析することが前提とされていた [4, 5, 7, 8, 15, 19]。また、認識された行動の善し悪しを評価するスキルアセスメント(技術品質査定)システムは、特定のタスクを定量的に評価することでユーザの技術の改善を促すことを目的としている [11, 12]。しかし、これらの研究ではリアルタイムでの支援を目的としておらず、ユーザが推奨されない行動や危険な行動を取ったとしても、リアルタイムにそれを指摘したり防いだりといった対応をとることができない。例えば、馬場馬術のスキルアセスメントでは [11]、落馬の危険を伴うような誤った行動をとったとしても、それを回避するよう働きかけることはできない。

### 3. 提案手法

本研究では、ユーザが複数のタスクを並行して行なう状況においても、リアルタイムでその適切性を検証するシステムの実現を目的とする。本研究では、まずリアルタイムで単一のタスクを検証できるようにする。そして、これを複数のタスクが並行に実行される状況に対応できるよう拡張する。

まず単一のタスクの検証を行えるようにするため、タスクモデルを定義し、個々のタスクに遂行状態に対応したユーザが行うべき行動や誤った行動を定義する。そして、ユーザの行動系列とタスクモデルとを比較することで、その行動の正誤を検証する。そして、複数のタスクが並行して進められる場合に対応するために、各タスクモデルとユーザの行動系列を比較し、それぞれの検証結果を一つに統合する。この時、システムに定義された全てのタスクモデルと比較を行ってしまうと、計算量の増大や精度の低下を招くことになる。そこで、発生し得るタスクの中から実際に比較するタスクの選出を行う。これらを行うために以下の3つの段階に分けて行なう。

#### (1) 単一タスクの定義

タスクをユーザの行動に遷移する状態系列として定義する。この時、遷移の種類を「進行イベント」、「警告イベント」、「無関係イベント」に分類し、それに基づいて検証

を行う。

#### (2) 並行タスク検証

複数のタスクを並行して遂行する場合、各タスクモデルをそれぞれ評価する。それぞれのタスクの遷移が競合した場合の優先順位を定義し、最終的な検証結果を求める。

#### (3) 実行タスクの選出

検証対象とするタスクを制限するため、隠れマルコフモデルを用いて各タスクの実行尤度を算出し、より検証の必要性が高いタスクを選出する。(2)において、一度に検証するタスクの数を制限することで検証器の計算時間と検証精度の低下を防ぐ。

提案手法の概要図を図1に示す。初めに、行動認識器のイベントをシステムが受取すると、用意されたタスクの中から検証すべきタスクを選出する((3)実行タスクの選出)。これにより検証するタスク数を制限する。次に、選出されたタスクのモデルは入力されたイベントに応じてそれぞれ状態遷移する((1)単一タスクの定義)。この時、各タスクにおける遷移の種類につけられた優先順位に従って評価をおこなう。そして、各タスクの状態遷移に合わせ、ユーザにどのような支援を行なうかを決定し、必要ならばユーザに警告などを与える((2)並行タスク検証)。

#### 3.1 タスク定義

まず、ある一つのタスクをどのように検証するかを考える。本研究では、単一のタスクをユーザの行動(イベント)によって遷移する状態遷移モデル、すなわち状態系列とそれを遷移させる行動の集合として定義する。この時、発生した遷移がタスクを正しく進行させる適切なものか、将来的な失敗に繋がる不適切なものか、を確かめることによってタスク検証を行う。

一般に、ある行動系列について、それを検証するには2つのアプローチが考えられる。一つは、正当な行動系列を定義し、その系列以外の行動系列が生じた場合に警告を発するアプローチである。もう一つは、逆に、警告すべき不正な行動系列を定義しておき、定義された行動系列が生じた場合に警告を発する(定義された行動系列以外の系列は正当なものとして扱う)アプローチである。いずれについても、母集団となる行動系列が無数にある中、定義される行

動系列の背反となる系列の正しさを保障することが困難である。そこで、本研究では双方のアプローチを組み合わせると共に、正当でも不正でもない行動系列を考慮する。

すなわち、あるイベントによって遷移が発生したとき、それがタスクを正しく進行させる適切なものならば、**進行イベント**と定義する。また、禁止されるべき警告を要するイベントであれば、**警告イベント**と定義する。さらに、タスクの進行には寄与せず、警告対象でもない行動を**無関係イベント**と定義する。前述のように、進行イベントと警告イベントの数はタスクによって限定され得るのに対し、無関係イベントは無数に存在する。従って、本研究では、進行イベントと警告イベントのみを明示的に定義し、それ以外のイベントを無関係イベントとして扱う。

表 1 及び図 2 に本研究におけるイベントと状態遷移をまとめる。これらの定義によって、タスクは任意のイベントを入力系列として受理する状態遷移モデルとして構成される。

表 1 イベントの種類

|         |                                      |
|---------|--------------------------------------|
| 進行イベント  | タスクを正しく進行させるイベント<br>受理後、タスクは適切な状態に遷移 |
| 警告イベント  | 禁止・警告すべきイベント<br>警告後、そこから復帰するための状態に遷移 |
| 無関係イベント | タスクの成功・失敗に関与しないイベント<br>タスクは現状態を維持する  |

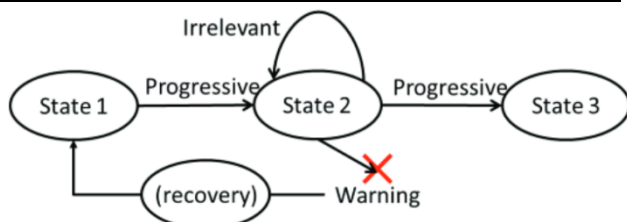


図 2 イベントに対応する状態遷移

### 3.2 並行タスク検証

複数のタスクを並行して遂行する場合に対応するため、定義された（生じうる）複数のタスクについて同時に行動系列を検証する。直接的な手法では、全てのタスクについて状態及び遷移の可能な組み合わせを網羅的に作成し、行動系列と比較する必要がある。しかし、タスクの数やタスクの状態数に応じて組み合わせの数は膨大化するため、それら可能な組み合わせ全てを考慮するのは現実的ではない。

そこで、本研究では各タスクをそれぞれ個別に、かつ、並行して検証する。すなわち、各タスクモデルはそれぞれの現状態についてイベントを評価し、イベントに応じて遷移する。この時、あるタスクにおいて、警告イベントとして受理されるならばユーザに対して警告を与えるようにする。このとき、タスクモデル間で異なるイベントの種類または重複するイベントの種類に対処するために、イベント

の受理に対して優先順位を定義する。基本的には警告イベント、進行イベント、無関係イベントの順に扱う。

一つ以上の警告イベントとその他の遷移との組み合わせは、他のタスクでは正常に受理されたとしてもあるタスクについては失敗に繋がる可能性を示す。そこで、警告イベントの優先度を高くし、それらの組み合わせではユーザに警告を与える。一つ以上の進行イベントと警告以外のイベントの組み合わせでは、タスクを適切に進行させるイベントとして受理される。すなわち、進行イベントとして扱われるタスクモデルでは次の状態へと遷移する。全てのタスクモデルについて無関係イベントに分類される場合、状態は一切遷移せず、全体としても無関係イベントとして扱われる。

また、並行タスクにおける特殊な事例として、あるタスクにおいて他のタスクの作業に移らず、不可分に実行することが要求される場合がある。例えば、重大事故に繋りかねない医薬品の充填作業などは、取り違い防止のために一つずつタスクを進め、完了してから他のタスクに移るべきである。こういった状況下では、並行タスクを禁止するため、不可分な実行を要求されているタスク以外の進行イベントを全て警告対象と見なすようにする。

### 3.3 実行タスクの選出

3.2 節で述べた手法を用いる場合、全てのタスクモデルを網羅的に評価することになるため、業務内容が多岐にわたるような環境では計算量の増加と検証精度の低下を招くことが予想される。そこで実行尤度の高いタスク、すなわち検証の必要があるタスクのみを選出し、検証対象となるタスクモデルの数を限定する。

#### 3.3.1 隠れマルコフモデル

本研究では実行尤度の算出に隠れマルコフモデルを採用する。ユーザが複数のタスクを並行して遂行する状況は、タスクを切り替えながら具体的な行動を生成している状態と考えることができる。すなわち、システムが直接観測するのはイベント(行動認識器の出力)であり、その内容は実行中のタスクに依存したものとなる。例えば、看護師が点滴業務とベッドメイクとを並行して行う場合の模式図を図 3 に示す。以上のことからそこで、タスクを隠れ状態とし、行動ラベルを観測出力とした隠れマルコフモデルを考えることができる。この隠れマルコフモデルについて隠れ状態の尤度を求めることは、行動認識の結果(観測系列)からタスク実行の流れ(状態遷移)を推定することになる。すなわち、あるタスクに対応する状態が遷移中に現れるならば、そのタスクを遂行中と見なし高い実行尤度を得ると予想される。

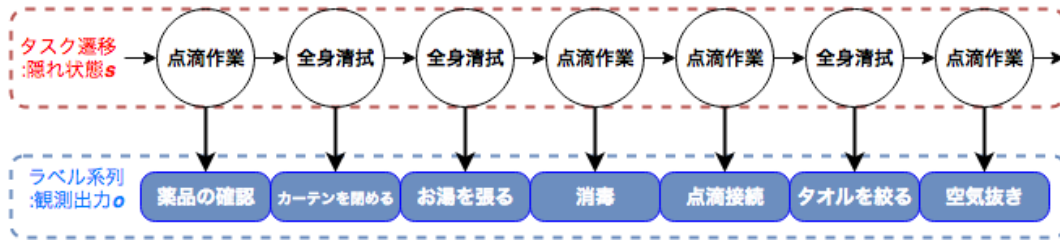


図 3 隠れマルコフモデル(点滴とベッドメイク)

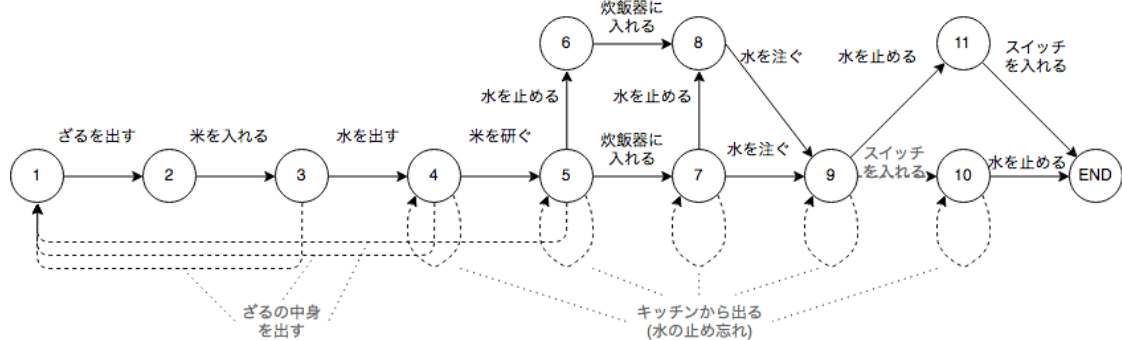


図 4 タスクモデルの例(ご飯を炊く)

### 3.3.2 Viterbi アルゴリズム

本研究ではタスクの選出のために隠れ状態の尤度を求めるにあたり、Viterbi アルゴリズム [21]を採用する。

隠れマルコフモデルにおいて隠れ状態の尤度を得るには可能な状態系列全てに対して実際の観測系列を当てはめた場合の尤度を計算する必要がある。しかしこの方法の計算量は $O(N^T)$ ( $N$ =状態数,  $T$ =系列長)となるため系列長の長さで爆発してしまう。

Viterbi アルゴリズムは観測された事象の系列から隠れ状態の最も尤もらしい系列(ビタビ経路)を探索する前向き動的計画法アルゴリズムであり、その計算量は $O(N^2T)$ である。Viterbi アルゴリズムを数式で表すと以下になる。

$$s^* = \underset{s}{\operatorname{argmax}} P(o, s | A, B) \quad (1)$$

$s^*$ は最尤の状態系列,  $s$ が状態系列,  $o$ が観測系列,  $A$ は状態 $x_i$ から状態 $x_j$ への遷移確率行列,  $B$ は状態 $x_i$ での事象 $o_k$ の出力確率行列を表す。式(1)を解くために下記の漸化式を利用する。

$$s_i(0) = 0 \quad (2)$$

$$P(s_j(t)) = \max_i P(s_i(t-1)) \cdot a_{ij} \cdot b_i(o(t)) \quad (3)$$

$s_i(t)$ は時刻  $t$  で状態 $x_i$ に至る最尤確率の状態系列,  $o(t)$ は時刻  $t$  での観測出力を表す。上記式を基に下記式によって時刻  $t$  での最尤状態系列(以下ビタビ経路)を算出する。

$$s_j(t) = \left[ \underset{s_i(t-1)}{\operatorname{argmax}} P(s_j(t)), x_j \right] \quad (4)$$

$$s^*(t) = \underset{s_i(t)}{\operatorname{argmax}} P(s_i(t)) \quad (5)$$

これらの式を利用するためには隠れマルコフモデルを構成するパラメータ、遷移確率  $A$  と出力確率  $B$  が既知である必要がある。これらのパラメータは一般に学習によって算出するが、パラメータのばらつきが大きいほど精度が悪化することが先行研究により明らかにされた。そこで学習で算出したパラメータ $\{\bar{p}_{ij}\}$ に対し、 $\{\alpha(\bar{p}_{ij}-0.5) + 0.5\}$  ( $0 < \alpha \leq 1$ )で補正したパラメータを利用する。

### 3.3.3 実行尤度

タスクの選出は、1)隠れマルコフモデルの隠れ状態の西遊系列を求める、2)求められた状態遷移に基づき、各タスクに実行尤度を与える、3)実行尤度の高いタスクを選出する、の3ステップから成る。これらは以下の式によって表される。

$$\{s_i, P(s_i)\} = \text{Viterbi}(o) \quad (6)$$

$$L(x_k) = \sum_i P(s_i) \{i | x_k \in s_i\} \quad (7)$$

$$E = \{x_k | L(x_k) > \epsilon\} \quad (8)$$

式(6)では、ビタビアルゴリズムを用いて観測系列  $o$  から尤もらしい状態遷移の候補 $s_i$ とその確率 $P(s_i)$ を求める。式(7)では、各タスクの実行尤度 $L(x_k)$ を、その状態(タスク)  $x_k$ を含む遷移 $s_i$ が起こる確率 $P(s_i)$ の総和として求める。最後に式(8)では、閾値  $\epsilon$  以上の実行尤度をもつタスクのみを選出し、検証タスク集合  $E$  を生成する。

## 4. 評価

### 4.1 実験データと評価方針

提案手法の有効性を示すために、一般家庭における実際の家事を対象としてタスクモデルの作成、および、実データの取得を行なってシステムに入力し、検証を行なった。検証対象とするタスクは、家庭内で日常的に行われるタスク 58 種類(表 2)を設定した。タスクモデルの例を図 4 に示す。点線で表された遷移が警告イベントを示す。利用される行動ラベルは米をとぐ、水を使う、コンロを付ける、などがある。

行動データは、被験者 3 人から計 13 日分の家事における行動をビデオで記録し、ラベル付けを行なった。一日あたりの記録時間は約 15~20 分であり、全体で 244 分である。行動データは本来、人の動きを種々のセンサデータと

して収集し、行動認識器の識別ラベルが検証器へと入力することを想定している。しかし行動認識器には誤認識の可能性があり、認識器の精度は検証精度に大きく影響する。そこで本研究では記録したデータに手作業でラベル付けしたデータ(正解データ)をそのまま検証器へと入力して誤認識による影響を排除した。なお、加速度センサなどのウェアラブルセンサを用いて行動を取得することを想定し、ラベル付けはウィンドウ幅 512[msec]、ずらし幅 256[msec]でウィンドウイングを行った各ウィンドウに対して行なった。結果、延べ 57017 個のラベル付き行動データが得られた。

提案手法の実現可能性と最適なパラメータを知るために 3 種類の実験を行った。(A) 実行中のタスクを予め手作業で選出しておく検証器、(B) 用意されたタスクモデル全てを常に評価する検証器(提案手法から(3) 実行タスクの選出を除いたもの)、(C) 実行尤度を用いたタスクの選出を行う検証器(提案手法)、である。(A)はタスクの選出が理想的に行われた場合の性能を示すものである。

検証精度を基に、タスクモデルの定義((1))及び並行タスク検証((2))の妥当性を明らかにする。(B)では検証精度及び計算時間を(A)と結果と比較し、検証するタスクモデルの数の増加が検証システムの計算時間及び検証精度を悪化させることを確認する。(C)では(A)および(B)と比較を行い、提案手法の性能およびその有効性を明らかにする。

また、(C)では、先に補正値  $\alpha$  及び閾値  $\varepsilon$  を変化させて、それぞれの値における検証精度及び計算時間を比較することで、最適なパラメータの値を決定しておく。この時、並行タスクの遂行に被験者の特性が存在する可能性を考慮し、隠れマルコフモデルの遷移確率行列  $A$  及び出力確率行列  $B$  の学習を全てのデータセットを用いて学習させた場合(以下、「全データ学習」と呼ぶ)と、各被験者のデータのみで学習させた場合(以下、「被験者データ学習」と呼ぶ)の両方で行なう。そして、達成される提案手法の性能を(A)、(B)の結果と比較し、提案手法の有用性を明らかにする。本実験における評価指標の定義は以下の通りである。

**検証精度:** タスク検証器に行動認識器のラベルが入力されると検証結果としてタスクの進行、ユーザへの警告、無関係イベントの発生、のいずれかを出力する。各時点でのグランドトゥルースを目視にて記録し、出力された検証結果の一致を正解とした適合率 (Precision), 再現率 (Recall)。

**計算時間:** タスク検証器に行動認識器のラベルが入力されてから、検証結果が出力されるまでにかかる応答時間。

## 4.2 評価結果

### 4.2.1 実行中のタスクを手作業で選出した場合

(A) 実行中のタスクを予め手作業で選出しておく検証器でタスク検証を行った場合の結果を表 3 の上段に示す。適合率 0.944, 再現率 0.742 であり、大部分の行動を正確に検証できている。しかし、適切なタスクを選出しているにも関わらず、100%の精度にはならなかった。

表 2 採用されたタスクモデル(58 種類)

|                        |                          |                      |
|------------------------|--------------------------|----------------------|
| BathPreparation        | CookCutMelon             | CookWheatTea         |
| BodyTemperature        | CookCutPersimmon         | DehumidifierWater    |
| Measurement            | CookFriedRice1           | ThrowAway            |
| ChangeClothesInMorning | CookFriedRice2           | GarbageThrowAway     |
| ComeHomeFromParttime1  | CookGyozaSoup            | GetUpInMorning       |
| ComeHomeFromParttime2  | CookHotCake              | GoOutHoliday         |
| ComeHomeFromUniversity | CookKatsuDon             | GoOutParttime        |
| CookBroiledFish        | CookMapoTofu             | GoOutUniversity      |
| CookBroiledGyoza       | CookMisoSoup             | HairDryerWorking     |
| CookBroiledMeat1       | CookMochiSoyflour        | LaundryHangOut       |
| CookBroiledMeat2       | CookMochiSoyssauce       | NotebookComputerWork |
| CookBroiledWiener      | CookNabeCooking          | PersonalComputerWork |
| CookButaDon            | CookOyakoDon             | TablewarePuttingAway |
| CookCabbage            | CookRiceSteam            | TablewareSoaking     |
| CookCocoaBread         | CookSanshokuDon          | TablewareWashing     |
| CookCreamStew1         | CookScrambleEgg          | TableWipeOff         |
| CookCreamStew2         | CookSpaghetti            | TakeNap              |
| CookCurryRice1         | CookSunnySideUp          | VacuumCleaner        |
| CookCurryRice2         | CookToastedBread         | WearPajamas          |
| CookCutApple           | CookVermicellifineNoodle | WheatTeaSetting      |

表 3 3 手法による実験結果

|              | 適合率   | 再現率   | 計算時間[msec] |
|--------------|-------|-------|------------|
| (A)          | 0.944 | 0.742 | 0.151      |
| (B)          | 0.456 | 0.767 | 17.038     |
| (C) (全データ)   | 0.517 | 0.767 | 2.812      |
| (C) (被験者データ) | 0.543 | 0.773 | 2.469      |

この原因として、主にタスクモデルの定義の不備であったことが上げられる。本研究では状態遷移の入力イベントとして、行動識別のラベルのみを用いており、行動の実行時間などを考慮していない。そのため、例えば、食洗中に一時的に手を休めるなどしてイベントが二重に入力された場合に対して、本来であれば前者だけでなく後者も進行イベントとして判断されるべきだが、前者の時点で状態遷移が完了しているため後者を進行イベントと判断できない場合があった。また、ある実験データにおいては単に著者が想定していなかった手順でタスクが遂行されており、途中から適切な状態遷移が行えていない場合があった。

一方で、並行タスク検証方法についてはその妥当性が示されている。タスク間の優先順位付けは警告イベント、または、進行イベントを受理されやすくする。受理されるイベントの大部分が無関係イベントであり、優先順位付けが誤りであった場合、適合率が大きく低下すると考えられるが、適合率 0.944 と高い結果を得ることができた。このことから、優先順位付けが妥当であるとともに、並行検証も問題なく動作している、と考えることができる。なお、100%でない主な原因は前述されたタスクモデルの定義の不備に

よるものであると考えられる

#### 4.2.2 用意されたタスクモデル全てを常に評価する検証器

(B) 用意されたタスクモデル全てを常に評価する検証器でタスク検証を行った場合の結果を表 3 の二段目に示す。適合率 0.456, 再現率 0.767 であり, (A)の結果と比較して大きく適合率が下がっている。また, 計算時間が 17.038msec であり, (A)の計算時間 0.151 の 100 倍以上になっている。一方で, (B)の適合率の方が(A)の適合率よりもわずかに高い。その原因として, (A)では進行イベントが二重に入力された場合に二回目を進行イベントと判断できなかったが, (B) では本来実行していないタスクの進行イベント受理することで, 結果的に進行イベントとタスクの状態遷移を生じてしまう場合があったことによる。この原因は, 4.2.1 節で述べたよう, タスクモデルの定義の不備であると考えられる。一方で, (A)と比較して適合率及び計算時間が大きく低下したことは, 検証するタスクモデルの数の増加が検証システムの計算時間及び検証精度を悪化させる, と予想に適合するものとなった。

#### 4.2.3 提案手法

まず, 補正值  $\alpha$  及び閾値  $\epsilon$  の最適値を決定するため, それぞれのパラメータを変化させた場合の適合率, 再現率の結果を図 5, 図 6 に示す。それぞれ上側の図が全データ学習の, 下側の図が被験者データ学習の結果である。全データ学習でも被験者依存学習でも, 補正值  $\alpha$  及び閾値  $\epsilon$  が小さい程精度が上がる傾向がある。パラメータが小さくなるにつれて, 適合率は全データ学習で 0.517, 被験者データ学習で 0.542 に収束している。同様に, 再現率は全データ学習で 0.767, 被験者データ学習で 0.772 に収束している。

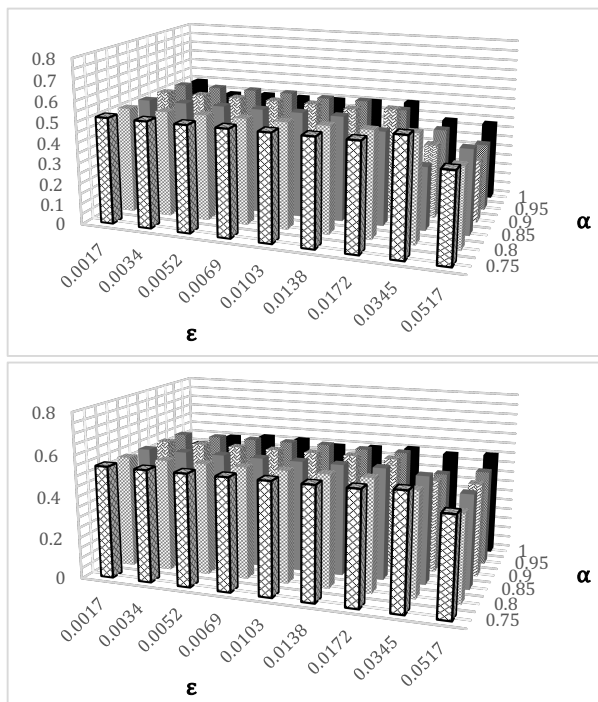


図 5 補正值及び閾値ごとの適合率  
(上) 全データ学習, (下) 被験者データ学習

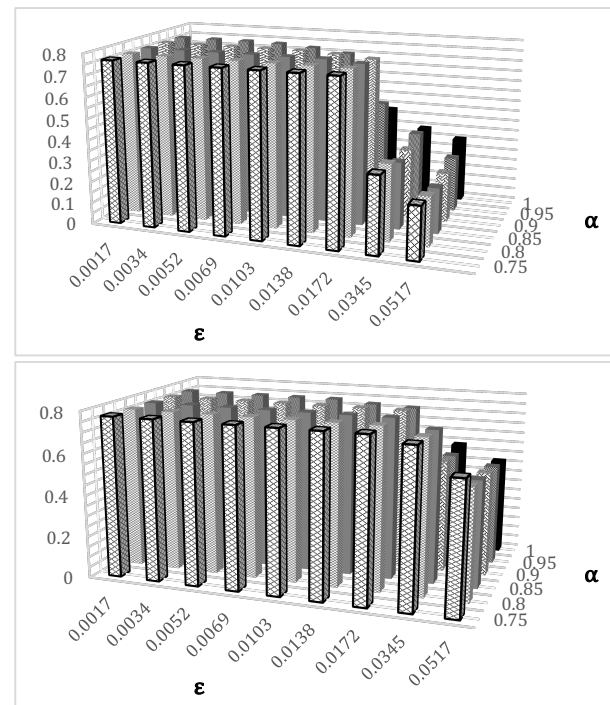


図 6 補正值及び閾値ごとの再現率  
(上) 全データ学習, (下) 被験者データ学習

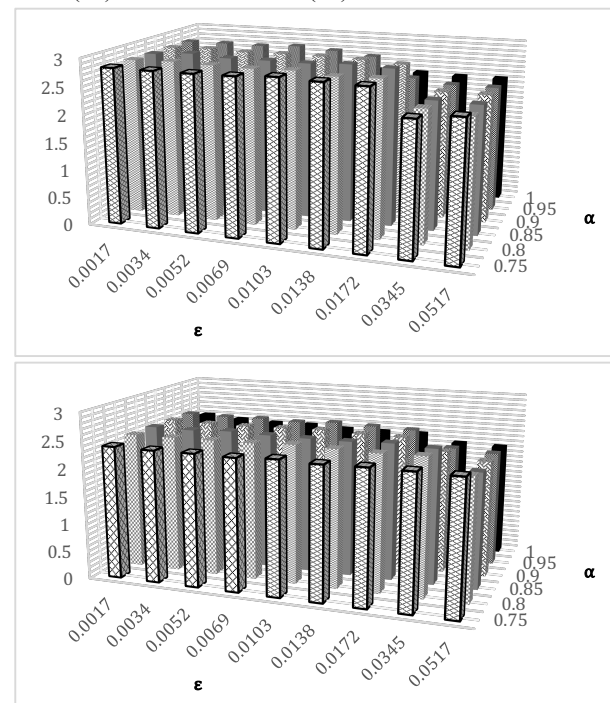


図 7 補正值及び閾値ごとの計算時間[msec]  
(上) 全データ学習, (下) 被験者データ学習

図 7 に提案手法における計算時間の結果を示す。単位は [msec] である。上が全データ学習, 下が被験者データ学習の結果である。全データ学習でも被験者依存学習でも, 補正值  $\alpha$  及び閾値  $\epsilon$  が小さい程計算時間が長く傾向がある。しかし, 計算時間では, パラメータの変化によって一定の値に収束していない。被験者データ学習では  $\alpha=1.00$  で最小の計算時間 2.057msec をとったが, 全データ学習では  $\alpha=0.80$  が最小の計算時間 2.289msec となった。

提案手法の計算時間は、全体を通して2~3msecのである。これは、(B)の計算時間 17.0038msec から大きく改善されている。一方で、提案手法の適合率、再現率は補正值  $\alpha$  及び閾値  $\varepsilon$  の変化によって大きく変化する。従って、パラメータは計算時間より検証精度に合わせて設定すべきである。適合率、再現率の両方を考慮し、全データ学習及び被験者データ学習において  $F\text{-measure}(\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}})$  が最大となったパラメータを最適値として設定する。その時の結果を表3の下段に示す。全データ学習では、適合率 0.517, 再現率 0.767, 計算時間 2.812 となった。被験者データ学習では、適合率 0.543, 再現率 0.773, 計算時間 2.469 となった。いずれの値も(B)の結果よりも良く、実行タスクの選出の有効性を確認できた。

## 5. まとめ

本研究ではユーザが多様なタスクを並行して進める環境下においても、ユーザのタスク遂行をリアルタイムで検証できる手法の提案を行なった。まず、タスクをユーザの行動によって遷移する状態系列として定義し、その遷移を進行イベント、警告イベント、無関係イベントに分類することで単一のタスクの遂行を検証できるようにした。次に、複数のタスクが並行して遂行される場合に対応するため、システム上に存在するタスク定義を全て同時に評価するようにすると共に、上記の遷移の種類に対して優先順位を設定し、タスク間の遷移の衝突においてシステムで対処すべき内容の検討を行なった。最後に、発生し得るタスクの中から、各タスクの実行尤度から、検証するタスクの選出を行なったその数を制限し、検証器の計算時間及び検証精度の低下を抑えた。

評価実験では、一般家庭での家事における 58 種類のタスクを定義し、3 人の被験者が家事を行ったデータを主として提案手法によるタスク検証を行った。全てのデータセットを用いて学習させた提案手法は適合率 0.517, 再現率 0.767 で被験者の行動を検証することができた。この時、入力から出力までの計算時間は平均で 2.812msec であった。

## 参考文献

- [1] 寺田 努. ウェアラブルセンサを用いた行動認識技術の現状と課題. コンピュータ ソフトウェア. 2011, vol. 28, no. 2, p. 43-54.
- [2] Aggarwal, J. K. and Ryoo, M. S.. Human Activity Analysis: A Review. ACM COMPUTING SURVEYS. 2011, vol. 43, issue. 3, no. 16.
- [3] Vishwakarma, S. and Agrawal, A.. A survey on activity recognition and behavior understanding in video surveillance. Springer Berlin Heidelberg. 2009, vol. 29, p. 983-1009.
- [4] 大星 直樹, 田中 達也, 桑原 教彰, 小作 浩美, 納谷 太, 小暮 潔. 看護師の行動データに基づく点滴業務の3階層モデルの構築. 医療情報学. 2008, vol. 28, no. 2, p. 83-91.
- [5] Chen, S. et al.. Automatic and Continuous User Task Analysis via Eye Activity. IUI '13 Proceedings of the 2013 international conference on Intelligent user interfaces. 2013, p. 57-66.
- [6] 宮脇 健三郎, 佐野 睦夫, 西口 敏司, 池田 克夫. 動作同期型調理ナビゲーションのためのユーザ適応型調理タスクモデル. 情報処理学会. 2009, vol. 50, no. 4, p. 1299-1310.
- [7] Kornela, J. et al.. Fine-grained Recognition of Abnormal Behaviors for Early Detection of Mild Cognitive Impairment. UbiComp'15 Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2015, p. 109-120.
- [8] Forkan, M. A. R. et al.. A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. Pattern Recognition. 2015, vol. 48, issue. 3, p. 628-641.
- [9] Riboni, D. et al.. Evaluating tooth brushing performance with smartphone sound data. 2015 IEEE International Conference on Pervasive Computing and Communications. 2015, p. 149-154.
- [10] Rubbi, M. et al.. Automated Personalized Health Feedback from User Behavior and Preference using Smartphones. UbiComp'15 Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2015, p. 707-718.
- [11] Thompson, R. et al.. Dancing with horses: automated quality feedback for dressage riders. UbiComp'15 Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2015, p. 325-336.
- [12] Khan, A. et al.. Beyond activity recognition: skill assessment from accelerometer data. UbiComp'15 Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2015, p. 1115-1166.
- [13] Murao, K. et al.. Recognizing activities and identifying users based on tabletop activities with load cells. iiWAS '15 Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services. 2015, no. 39.
- [14] Enamul, H. et al.. Holmes: A Comprehensive Anomaly Detection System for Daily In-home Activities. DCOSS '15 Proceedings of the 2015 International Conference on Distributed Computing in Sensor Systems. 2015, p. 40-51.
- [15] Steil, J. and Bulling, A.. Discovery of everyday human activities from long-term visual behaviour using topic models. UbiComp'15 Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2015, p. 75-85.
- [16] Hong, Y. J. et al.. Mobile health monitoring system based on activity recognition using accelerometer. Simulation Modelling Practice and Theory. 2010, vol. 18, issue. 4, p. 446-455.
- [17] Rabbi, M. et al.. MyBehavior: automatic personalized health feedback from user behaviors and preferences using smartphones. UbiComp'15 Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2015, p. 707-718.
- [18] Lester, J. et al.. A hybrid discriminative/generative approach for modeling human activities. IJCAI'05 Proceedings of the 19th international joint conference on Artificial intelligence. 2005, p. 766-772.
- [19] Caroux, L. et al.. Verification of Daily Activities of Elders; A Simple, Non-Intrusive, Low-Cost Approach. Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility. 2014, p. 43-50.
- [20] Rim, H. et al.. A probabilistic ontological framework for the recognition of multilevel human activities. UbiComp'13 Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2013, p. 345-354.
- [21] Forney, J. D.. The Viterbi algorithm. Proceedings of the IEEE. 1973, vol. 61, no. 3, p. 268-278.
- [22] Ting-Fan, W. et al.. Probability Estimates for Multi-class Classification by Pairwise Coupling. The Journal of Machine Learning Research. 2004, vol. 5, p. 975-1005.