

携帯端末におけるユーザ操作支援方式の提案

清原 良三[†] 清水 直樹^{††} 松本光弘[†] 栗原 聡[‡] 沼尾正行[‡]

[†] 大阪大学大学院 情報科学研究科

^{††} 三菱電機 (株) 情報技術総合研究所

[‡] 大阪大学 産業科学研究所

近年、携帯電話やカーナビゲーションシステムは高機能化し、広域、近距離の無線通信を備えたユビキタス時代の中心端末となっている。そのため、使い方は複雑になっている。しかしながら、ユーザごとに見ると一定の機能は良く使うが、それ以外の機能はたまに使う傾向があると考えられる。そのため、ユーザごとに利用履歴からその状況によって適したUIを提供するコンテキストアウェア技術が盛んに研究されている。本報告では、ユーザの操作は、時間と場所によって決まることが多いという仮説に基づいて、アプリごとに異なる時間や場所の扱いかたに着目し、状況に依存したアプリケーション候補の抽出方式の構成案を提案し、操作履歴の扱い方を中心に簡単な評価を行う。

A Proposal of User Support System for Mobile Devices

Ryozo Kiyohara[†], Naoki Shimizu^{††}, Mitsuhiro Matsumoto[†],

Satoshi Kurihara[‡], Masayuki Numao[‡]

[†] Graduate School of Information Science and Technology of Osaka University

^{††} Mitsubishi Electric Corp., Information Technology R & D Center

[‡] The Institute of Scientific and Industrial Research, Osaka University

Recent technology advancement makes mobile terminals such as smartphones and car navigation systems small PCs with full of functions. The increasing number of functions designed to help user could produce a undesigned problem; increasing complexity of the operation. To solve the problem many research has been proposed to utilize context-aware technology, which could predict a user's preference based on past user-experiences in similar context. In this paper we propose a user support system using an application-dependent operation log which is separately designed for each applications. We also present preliminary evaluation based a simulation using the proposed system.

1 はじめに

近年、携帯電話やカーナビゲーションシステムは端末として高機能化し、広域、近距離の無線通信を備えたユビキタス時代の中心端末となっている。そのため、使い方は複雑になっており、ユーザごとに見ると一定の機能は良く使うが、それ以外の機能はたまに使うという傾向があると考えられる。

携帯電話では歩きながらの操作する時にはあまり画面を見ずに操作できれば都合の良い場合が多いし、カーナビゲーションシステムでは信号で停止している短い時間での操作を要求する場合も多い。これらの操作はユーザごとに異なることも多い。予め短操作で作業できるようなメニューを準備しておくことで対応可能であるが、多くのユーザが満足する

ような準備は困難である。

一方で、ユーザは時間や場所というまわりの状況や、それまでの操作などの状況に依存して端末を利用する傾向がある [1]。これらの傾向はユーザごとに異なるため、予め端末のユーザインタフェースをすべてのユーザにあわせた形で設計することは困難である。そこで、ユーザごとに利用履歴からその状況によって適した UI を提供するコンテキストアウェア技術の研究が多く実施されている。

ユーザのコンテキストを知るためには、端末上で知ることのできる時刻情報、どこでもほぼ入手可能な GPS 情報、あるいは限られた範囲で入手できる情報として近距離無線のインフラが充分整ったところで入手できる情報など様々な情報を利用することが想定されている。

本論文においては、インフラの整わない地域での利用も考え、近距離無線のインフラを想定せずに時刻と位置に基づいた方式を検討する。状況に依存して使い勝手を良くする方法として、メニューの順序を前回のアクセス履歴に基づいて変更する方法や、予めユーザがルールを定義しておきそのルールに従って動作させる方法 [2] など提案されているが、ここでは一般ユーザが使う携帯電話やカーナビシステムを対象とするため、自動的に学習していく方式を検討対象とした。

本論文では、このような位置や時刻といった携帯端末で取得できるコンテキスト情報を利用して携帯端末を使おうとしたときに、所謂あうんの呼吸で可能性の高いアプリケーションの候補を提示し、多くのケースでは少ない手順で所望のアプリケーション起動ができる状況依存型の UI の構築手法に関して述べる。

2 関連研究

人によって使いやすい UI を設計する手法として、予測 / 例示インタフェースの研究 [3] が報告されている。携帯端末上で一般ユーザが例示しながら覚えさせるという方法はあまり考えられないが、予測インタフェースは仮名漢字変換を代表とするように有効な手法と考える。また、一般ユーザでなくある程度携帯電話の知識を持ったユーザであるならば実世界指向プログラミング [4] も有効と考える。

携帯端末は、時計機能、加速度センサ、周辺の

音センサといった端末独自の機能のほか、GPS、広域通信機能といったほぼどこでも使えるインフラ機能と近距離無線など特定のエリアで利用できる機能を備えた機種が多くなっており、ユーザの状況を的確に把握できるようになりつつある。このようなコンテキスト情報を利用する研究としては、状況に依存して起動するアプリケーションが変わる研究 [5] がある。この方式は、近距離無線を活用して状況を確認し、一つのボタンで所望の動作をさせようというコンセプトである。このような考え方は携帯端末に対しても有効であると考えられる。携帯端末に応用した例 [6] では、環境側のコンテキストとユーザ側のコンテキストを分けて、双方において状況に依存した動作ができるようにしており、非常に有効な方法と考える。

また、どのようにしてユーザの操作を予測するかも課題であり、状態遷移を含めてモデル化を行う研究 [7],[8],[9] がある。操作をどのような観点から見るかによってモデルは変わる。

ある状態においてあるイベントがおきた場合にどうなるかを予測に関しては、一般に状況は連続値で得られるものが多いため離散化する必要がある、離散化の際の状態空間は構成法によっては、メモリ爆発を起こしたり、情報の補完が難しい問題があるが、いかに状態空間を効率よく構成するかの研究 [10] もあり、自己組織化マップによる解決を行っている。また、[2] では GPS 内蔵の携帯電話をターゲットにして位置情報をクラスタリングすることにより、操作履歴を利用した操作予測をサーバ上で位置分析と連携して行う手法を提案している。

一方、このような研究で想定する近距離無線のインフラは少なくともここ数年はどこでも使えるという状態ではない。そこで本論文では環境に依存せず、ほぼどこでも使える機能のみを利用し、端末のみでコンテキスト情報を利用してユーザ操作を支援する機能を携帯端末に搭載するためのモデル化と方式の設計を行い評価し、その有効性を示す。

3 提案携帯端末の構成

本提案システムの全体構成ブロックとして携帯電話やカーナビゲーションシステムとして考えられる基本構成を図 1 に示す。UI は部品など含めてアプリケーションと独立の機能である。つまり UI

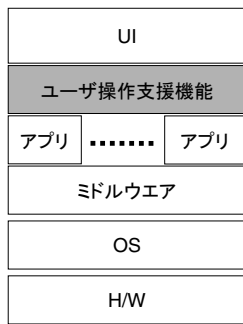


図 1: 全体ブロック構成

部は画面の大きさや解像度、あるいは入力手段といったハードウェアに依存する部分が含まれるのに対し、アプリケーションはハードウェアに非依存でサービス内容に依存する部分である。ユーザー操作支援機能を UI とアプリケーションの間に配置する。ユーザー操作支援機能がアプリケーションの操作ログを UI 部から取得し、実際に利用するアプリケーションの候補を抽出するなど実行する。

図 2 にユーザー操作支援機能の詳細機能ブロックを示す。操作ログは、アプリケーション利用時の状態を記憶するために時刻情報や位置情報といった環境の情報を取得し合わせて保存する。また必要に応じて、例えばブラウザならこの URL にアクセスしたかの情報や、メールなら誰にどんな情報に基づいたメールを出したかなどアプリケーション内部の情報も合わせて保存する。

状態空間構成機能は、取得した操作ログをシステムで利用しやすいように連続値データは離散化する。元のデータの少ない部分は補完処理を行う。その結果を重み付き状態遷移表という形で保持する。

候補抽出メニュー合成エンジンは、重み付きの状態遷移表から次に来る可能性の高いアプリケーション候補を選択したり、引き続き起こるであろう複数の処理をまとめた一連の動作を図 3 に示すようなメニューとして合成し、アプリケーションの候補に入れる機能を持つ。

提示された候補に対してユーザー操作支援機能はユーザの行動を結果のフィードバックとして利用し、結果を評価したうえで重み付き状態遷移表の重みを修正したり、場合によっては状態遷移表を再構

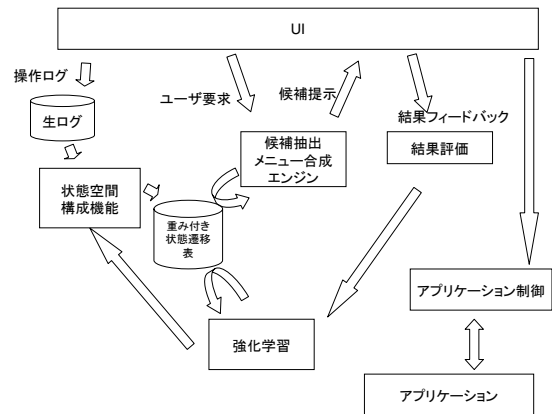


図 2: ユーザー操作支援機能ブロック

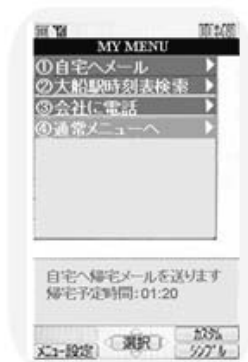


図 3: メニュー合成例

成する。

3.1 操作ログ

操作ログは各種イベントがあるたびに図 4 に示すような形で取得する。本来、状況が変わったことも適宜取得していれば、例えば会社から駅に出たときと家から会社に出たときの区別ができる。この 2 つはコンテキストは違うはずである。しかし、位置情報を時々によ取得すると電池の問題が発生する。そこで、次に示すようなイベントがあった場合のみのログを取得することとする。

- アプリケーションのユーザ操作による起動イベント
- アプリケーション操作に関する操作履歴

2006/10/20	09:55:47.301000	3011462e	3	システム起動完了
2006/10/20	09:55:48.235000	301147d5	3	アラーム設定完了
2006/10/20	13:21:29.086000	302007c5	4	電話着信
2006/10/20	13:21:29.086000	302007c5	5	電話着信終了
2006/10/20	13:25:31.086000	30200843	1	Java起動
2006/10/20	13:27:10.086000	302006e3	1	スケジューラ起動
2006/10/20	13:28:52.086000	302005f3	1	ブラウザ起動
2006/10/20	13:35:17.086000	30200844	2	Java終了

図 4: アプリケーションログ

- メール着信時のメール機能起動イベント
- 音声着信時の電話機能起動イベント

近距離無線機能や NFC(Near Field Communication) [11] 機能を搭載している携帯電話も多いが、ここでは想定しないこととする。

3.2 状態空間

3.2.1 状態空間構成問題

状態空間は予測する行動パターンが多い場合にはメモリ爆発を起こす可能性がある。しかし、携帯電話の操作のように限られたパターンしかない場合はその可能性は低いと考える。また行動を起こす場合の状態は必ずしも過去に経験した場所や時刻には起こらないと考える。そこで状態の補完処理が必要になる。そのため状態空間のクラスタリング構成が必要となる。

状態空間の構成は以下に示す 2 点から考える。

1. 各状態に応じて次に起動されるアプリケーションが予測できる構造
2. 各状態に応じて、連続して起こすべき複数の機能を予測できる構造

(1) は例えば駅で帰宅時にはブラウザを起動することが多い場合には、ブラウザを起動候補にするための仕組みとなる。

(2) の場合は、例えば駅で帰宅時にはブラウザでスポーツサイトを見るというのが起動候補になる仕組みとなる。通常は (2) のようなメニューは存在しない。ブラウザメニュー内のブックマークに存在する可能性がある程度であるため、メニューの合成が必要となるケースである。

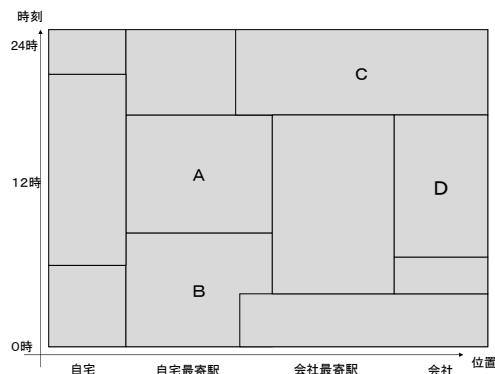


図 5: 状態空間

(1) のケースを想定し、状態空間は図 5 に示すようなクラスタリング構成をとる。この例は 2 次元の例で、待ち受け状態を意味している。細かく記載すると各 ICON の状態なども含む 10 次元以上の空間になるが、2 値を持つケースが多く実際に連続値を持つ部分だけで空間を考えると 4 次元から 5 次元程度である。

図 5 中 A の部分は自宅最寄駅で昼間の時間帯である。これは昼間に自宅最寄駅にいるのは出勤ではなく自由行動のときを示している。この場合は会社に行くのではないので、時刻表などを参照するなどが考えられる。B は朝であり、朝の出勤時の自宅駅の近くの場合は時刻表も候補だがむしろニュースサイトを見るケースの方が多いかもしれない。C のように夜遅くになると、会社近くの駅だろうが会社にしようが、仕事を終えた後の帰宅なので例えばプロ野球の結果を見たいかもしれない。このように考えると連続値で示される情報の離散値への変換では、状態空間は別の要素とも関連して細かくわけたり、広くわけらなどする必要が出てくると想定できる。

3.2.2 時間優先による状態空間の構成法

まず初期状態クラスタリング構成を決める。例えば、時刻であれば 24 時間を 1 時間ごとにする。次に位置情報も適切なメッシュに切ることにより決める。高さの情報も取得できるのであれば立体扱いとなる。このように連続値をとるものは初期状態のクラスタリング構成を決める。次に操作ログを解析し、それぞれの対応する空間位置を決める。図 6 は

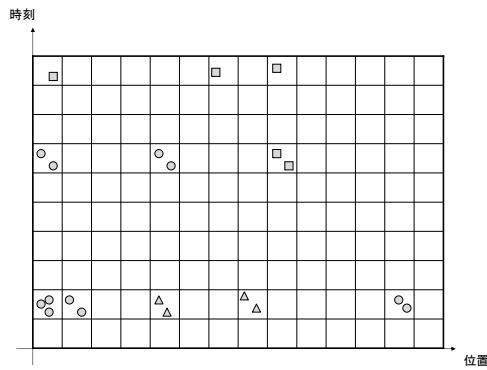


図 6: 状態空間の初期状態

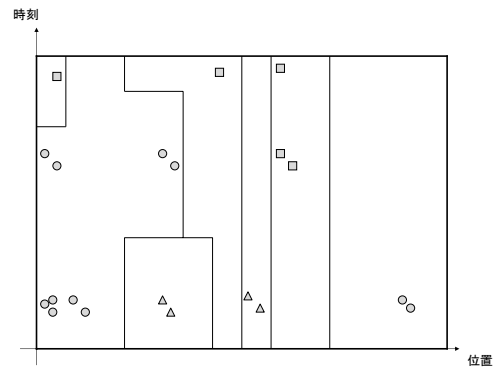


図 7: 状態空間の初期構成例

2次元の例である。

このような状態空間のクラスタリング構成においては、相当の期間の準備期間を設けログデータを取得したとしてもユーザ操作の時間や位置には偏りがあり領域上で満遍なく操作が行われるわけではない。例えば図5のDの領域であまりアクセスが昼間になかったため、会社として定時内を一律に扱っていたとする。たまたま昼休みにアクセスする場合においては本当に定時内として扱って正しい結果がでるかどうかはわからない。このような部分の補完が必要になる。ここでは、補完は、近い周辺の状態に合わせる。合わせるべき時間や位置といった情報に優先度を設ける。例えば位置優先とすると位置が同じで近い時間のもと同じとみなすという補完を行う。

そこで、各状態を構成する要素間で優先度を決める。例えば時刻優先とした場合には時刻は違っても他の条件が同じものはなるべく同じ空間になるように空間を構成する。このように構成する図6の例では図7に示すような状態空間が構成される。

3.3 メニュー合成

次にこれらの状態に対して特定のイベントが起きた場合に状態遷移を図8に示すように重み付き状態遷移表として保持する。この例ではわかりやすいように重みを確率で示した。確率の合計が100%でないものもあるが小さなものは省略して記載している。

例えばAという状態で特定のボタンが押されるとか、携帯を開くあるいはスライドさせたなど

のイベントが起きたときには、ブラウザを利用するケースが40%であり、メール利用するケースが30%であることを示している。さらに、ブラウザで上り時刻表を見る確率が70%、下り時刻表を見る確率が20%である。友人にメールを送る確率は90%となる。そうすると、Aの状態では、友人にメールを送る確率が27%である、上り時刻表を見る確率が28%、下り時刻表を見る確率が8%となる。

ここで、例えば確率20%以上のケースのみをアプリケーションの実行候補とするのであれば、友人A宛メール起動と上り時刻表をブラウザで見るという2つの候補をユーザに示す。ここで、ブラウザの起動は40%の確率なので、ブラウザの起動というメニューを入れるかどうかであるが、時刻表を見ないケースは40%*30%であるため12%に過ぎないため入れないが、もし10%以上の確率のケースをメニュー候補に入れるとすれば候補に入ることとなる。

3.4 強化学習

携帯端末上でのアプリケーションの起動においては、ユーザが望むアプリケーションを起動できたかどうかは、その後のユーザの操作によって正しかったかどうか分かる。アプリケーションの候補を出した場合は、ユーザの操作は以下に示す動作のうちいずれかである。

- 一定時間以内に候補の中からアプリケーションを選んで選択してしばらく実行（正解）

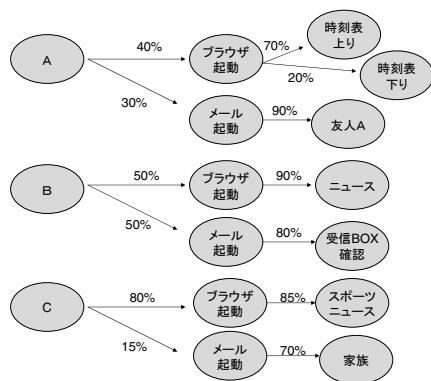


図 8: 重み付き状態遷移表

- 一定時間以内に候補の中からアプリケーションを選択するが一定時間以内に終了。(間違っって選択したと考え不正解)
- アプリケーションの候補リストを強制終了し、別のアプリケーションを起動した。(不正解)
- アプリケーションの候補リストを強制終了したが、候補リストに含まれるアプリケーションを一定時間以内に起動した。(正解)
- 上記以外の場合は正解か不正解か不明

多くの操作においては正解か不正解かがわかるため、この結果をフィードバックすることにより強化学習することが可能である。

一般的な TD 学習とすることにより、機会が多ければ適切なパターンに落ち着くと考え。TD 学習での行動予測では、目先の可能性の高いもので判断するのではなく、最終目的の可能性（報酬の最大化）を狙う。そういう意味でメニューの動的合成を行い、最終目的を意識したアプリケーションの選択候補をユーザに見せる方式となる。そのため、式 1 に基づいて、メニューにするべき候補を評価する。 $V(n)$ を時刻 n における各候補の評価値とする。 $\gamma(x)$ は時間 x 前の報酬の割引率、 $r(t)$ は時刻 t における報酬である。

$$V(n) = \sum_{t=0}^{n-1} \gamma(n-t)r(t) \quad (1)$$

一般に強化学習においては、正しい選択がされた場合には報酬が与えられ、次の候補選択の際には、報酬が最大になるように候補を選択することになる。学習を重ねれば 1 回違う操作をしたからといって、次回に別の候補が出てくるわけではない。しかし携帯電話の使い方では、時間と場所によるパターンが突如変わるケースも当然ある。このような場合には一般に割引報酬の考えを使う。

与えられた報酬は、時間とともに価値を減らす。 $\gamma(n-t)$ はその係数である。これによって、最近の行動パターンでの報酬が高くなり、何回か同じパターンを実施すると学習効果をすぐに表すことができる。ここで学習による報酬の割合と割引率をどのようにするかで実際の学習の効果が決まってくる。

実際のインプリメンテーション上は時刻を実時刻で扱う方法と候補抽出要求があった回数で扱う方法が考えられる。また、状態ごとの候補抽出要求回数で扱うことも可能である。実時刻で扱う場合は、しばらくその状態にならなくて、複数回たまたまその状態になった場合などに最近の操作パターンの優先度が高くなるという特性がでてくる。一方要求の回数を時刻とみなすとまったく要求がなかった時間が連続してあったかのように扱われる。どちらが良いかはユーザの特性によるが、ここでは実時間で扱うこととする。

また、割引率に関しても図 9 に示すように一次関数的に減っていくのか、二次関数的に減るのかまたは、直角に極端に減るのかなどパターンがある。これはユーザに依存すると考えるが、二次関数的に減り、ある時刻以前のものは計算しないモデルと考える。しかし、時によって直角に曲がるパターンなどの場合は次に述べる状態空間の再構成が必要なケースで対応する。

3.5 状態空間再構成

状態空間をクラスタリング構成し運用を始め、強化学習が進むと状態空間を構成する上で未踏の部分であって、補完をしたことにより状態空間を決めた場合もあれば、ユーザの使い方が何かをきっかけに変わる場合もある。このような場合には、最近の操作ログから新たに状態空間を再構成する方がユー

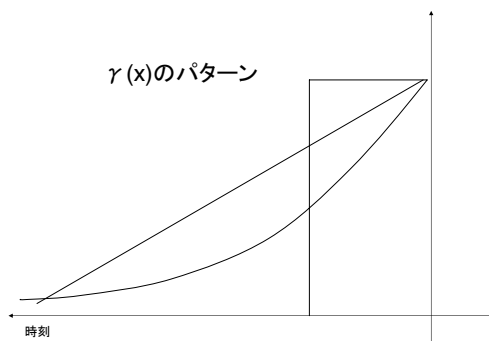


図 9: 時間による割引率

ザの要望が正しく伝わることになることを考える。再構成のきっかけは次のいずれかが考えられる。

- 毎日あるいは毎週決まった時間や充電中などに再構成を試みる。
- 学習したら必ず再構成を試みる。
- 学習不正解だった場合は必ず再構成する。

充電中で待ち受け状態の場合が電池の状況に影響を与えず計算量が多くても気にせずに再構成ができることを考える。

4 評価

4.1 ログ

実際の携帯電話の使用状況 [1] のように数ユーザーの自己申告に基づき実際の操作をシミュレートすることによりログデータおよび操作数に関して評価を行った。実際にログに入るべき情報とテキストファイルとして必要な情報サイズを表 1 に示す。

アプリケーションのログは特定のパターンを知るという意味では休日の状況も含めるため最低 1 週間分を取得しておく必要がある。1 週間分では、未踏の領域も多いと考えられるためできれば 1 ヶ月分程度はログを取得しパターンを洗い出す必要がある。これらのログは 1 つのログで図 1 にあるように 64 バイト程度であり、1 日あたりの平均したアプリケーション操作数 100 として、6.4K バイトあれば充分であり、1 月分ためたとしても

表 1: ログ情報の種類と必要情報サイズ例

ログ情報	情報サイズ (バイト)
日付	10
時刻	8
GPS	20
騒音レベル	10
方向センサ	10
アプリケーション ID	7
動作種別	1
合計	64

200K バイトもあれば充分である。現在の携帯端末であればこの程度は問題のない容量と考える。

実際に、社会人である数人の状況をヒアリングしたところ、多い人で 10 日ほどでの操作数が 250 であった。高校生などであれば、この数倍にはなると想定されるが、アプリケーションの操作数が 1000 だとしても問題のない大きさと考えられる。

4.2 操作数

実際の使い方としては、ユーザが操作支援モードを設定しておいた場合にログを収集して学習し始めることになる。特定のボタンを押した場合に、自動的に候補が抽出されることになる。この場合、正しく候補アプリから操作を抽出できた場合には、操作数はメニューの中のカーソルのダウンまで 1 操作と考えると最低が 2 操作であり、候補の数を足し分が最大の操作数となる。多くの場合はこの操作よりもメニューを入れていくだけで多くなる。ましてや、URL を入れたり、アドレス帳からあて先を選択する操作を入れるようなケースでは明らかに操作数が小さくなる。

一方、誤った場合には操作数は確実に 2 増えることになる。特定のボタンを押すことと、戻るボタンを押すことによる。または図 3 に示すように間違った場合のボタンを用意することにより 1 操作を減らすことも可能と考える。

確率を考えると正解の方が多いと考えるため、全体として平均操作数は小さくなることを考える。

4.3 学習

学習機能を携帯電話で実現する場合には、CPU速度とメモリ容量、電池残量といったリソースの制限が課題になると考える。イベント発生時にのみGPSなど電池消費量の多い機能を使うことにより残量の問題は解決するが、候補抽出時にGPSで測位をしていては候補の抽出に時間がかかる問題がある。そのため、加速度センサなどを利用してイベント予測を行い、ユーザが動作をはじめそうな場合には予め状況の情報収集をしておく必要はある。

また、重み付きの状態遷移表では、割引率を再計算して表の再作成が必要となるが、当該状態の場合の部分だけを再計算し、古い重み付きのデータに関しては充分低い割引率になっていると考え再計算しない方式とするとともに、携帯電話の操作パターンは限られた種類に納まると考えられるので、計算量も抑えられ問題のない範囲と考える。

5 おわりに

本論文では、携帯端末におけるユーザ操作支援機能の手法の提案として、操作履歴を取得し、この結果を状態空間に変換し、未踏の領域を時刻優先によって補完することによりクラスタリングを行い、強化学習によってより確率の高い状態遷移表としていく手法を提案した。また使い方の突如の変化に対しては時間がたつと割引率の上がる割引報酬の方式を導入することにより、突如の使用方法の変換に耐えられる方式とした。また、状態空間の特性が変わることに対しては充電時間などを利用した状態空間の再構成を提案した。

今後は提案した方式に以下の課題があると考えられる。

- 報酬の率である学習率や割引率に関して、実際の携帯電話への適用による最適な値の評価
- ログ保持期間の評価
- 状態空間の構成における補完方法や時刻優先、位置優先による方法の比較評価

これらの項目を今後実際のPC上のシミュレーションや実際の携帯電話への適用によりモデルとして適切かどうかを評価するとともに、人ごとに違う行動

パターンのモデル自身も学習していく方式を検討していく予定である。

参考文献

- [1] Jan Blom et al., "Contextual and Cultural Challenges for User Mobility Research", CACM, Vol.48, No.7, pp.37-41, 2005
- [2] 遠山緑生他, "コンテキスト情報と操作履歴の関連付けによる操作予測システムの提案", 情報処理学会, 第6回 UBI研究会, pp.83-90, 2004
- [3] 増井俊之, "予測 / 例示インタフェースの研究動向", コンピュータソフトウェア, Vol.14, No.1, pp. 1-16, May 1997
- [4] 増井俊之, "実世界指向プログラミング", 第40回情報処理学会冬のプログラミングシンポジウム予稿集, pp.19-25. January 1999.
- [5] 中島秀之, "マイボタンによる状況依存支援", 人工知能学会誌, Vol.16, No.6 PP.792-796, 2001
- [6] 河口信夫他, "ユビキタス情報環境における履歴を用いた機器操作支援手法", 情報処理学会, 第4回 UBI研究会, pp.57-62, 2004
- [7] Ivo Widjaja et al., "SPHERES OF ROLE IN CONTEXT-AWARENESS", OZCHI 2005
- [8] Panu Korpipää et al., "Utilising Context Ontology in Mobile Device Application Personalisation", MUM'04
- [9] Karen Henriksen et al., "Developing context-aware pervasive computing applications: Models and approach", pervasive and mobile computing, Vol.2, No.1, 2006
- [10] 岩崎秀樹他, "強化学習における自己組織化マップを用いた状態空間の自立的構成法", 人工知能学会全国大会, 2005, 1D3-05
- [11] <http://www.nfc-forum.org/>