

# スマートフォン利用者のコンテキストログを用いた アプリケーション推薦システム

嶋谷 健太郎<sup>1,a)</sup> 間下 以大<sup>1,2,b)</sup> 原 隆浩<sup>1</sup> 清川 清<sup>1,2,c)</sup> 竹村 治雄<sup>1,2,d)</sup> 西尾 章治郎<sup>1</sup>

**概要:**近年、スマートフォンは一般的なものとなり、情報端末として利用者にとって生活になくはならないものとなってきている。しかし、現状ではメールやニュース、周辺情報などの情報コンテンツを得るためには、情報を得るためのアプリケーションの選択、起動、必要な情報の検索、検索結果からの選択などを利用者が自ら行う必要がある。本論文ではこのような問題に対して、利用者の状況や利用履歴をもとに、利用者の求めているアプリケーションや情報を推定、提示する手法を提案する。提案手法ではスマートフォン利用者の状況(コンテキスト)を抽出するために混合メンバシップブロックモデル(Mixed Membership Blockmodels:MMSB)[1]を用い、センサ情報とアプリ起動履歴を学習し、アプリケーションを推薦する。提案手法の有用性について検証するために従来手法である、利用回数の多い物が上位に並ぶ、Most Frequently Used (MFU) 法、最近使用した物ほど上位に並ぶ、Most Recently Used (MRU) 法との比較実験を行った。その結果、提案手法は一部のアプリケーションで有効であることが確認された。

## 1. はじめに

スマートフォンにインストールしたアプリケーションや利用登録をしたサービスが増加するにつれ、利用者がその時々で必要なアプリケーションや得たい情報へ到達する手間や手順が増加している。現状ではメールやニュース、周辺情報などの情報コンテンツを得るためには、情報を得るためのアプリケーションの選択、起動、必要な情報の検索、検索結果からの選択などを利用者が自ら行う必要がある。例えば駅に向かって利用者が列車の時刻表を知りたいと考えると、スマートフォンのホーム画面から乗り換え案内アプリを起動し、駅名と時刻を入力して検索する必要がある。

このような問題に対して、コンテキストアウェアシステムが有効であると考えられる。コンテキストアウェアシステムでは、利用者自身や周囲の状況などをコンテキストとして認識し、コンテキストに応じた適切な情報の提供などを行う。このようなコンテキストアウェアシステムが実現されると、利用者が必要とする情報に到達するまでの手間

を短縮することが出来る。このような情報への迅速なアクセスは、移動中などの情報の取得に十分な時間や手間をかけられない状況で利用されることの多いスマートフォンでは重要である。

本研究では利用者の状況や利用履歴をもとに、利用者のコンテキストを推定し、求められているアプリケーションを推定する手法を提案する。近年ではスマートフォンに各種センサが搭載されている事から、センサの情報を利用したシステムも提案されているが、コンテキスト認識にセンサを用いる場合、コンテキストの定義がヒューリスティックな仮定に基づく場合が多い。そのため、アプリケーションや利用者によっては仮定されたコンテキストが不適切な場合も考えられる。そこで本研究では、利用者のスマートフォンの利用履歴とセンサログからコンテキストを学習し、アプリケーションを推薦するシステムの構築を目的とする。本研究では、コンテキストの学習に、Airoldiらの提案した混合メンバシップブロックモデル(Mixed Membership Stochastic Blockmodels: MMSB)[1]を用いる。本研究の応用により、同じ場所、同じ時間帯であっても利用者や利用者の動作、周囲の状態等に応じて適切なアプリケーションの推薦が可能になると期待できる。

## 2. 関連研究

コンテキストアウェアシステムに関する研究は様々なものが行われている。Chenらは現実のコンテキストを以下

<sup>1</sup> 大阪大学 大学院情報科学研究科  
〒565-0871 大阪府吹田市山田丘 1-5

<sup>2</sup> 大阪大学 サイバーメディアセンター  
〒560-0043 大阪府豊中市待兼山町 1-32

a) shimatani.kentaro@lab.ime.cmc.osaka-u.ac.jp

b) mashita@ime.cmc.osaka-u.ac.jp

c) kiyo@ime.cmc.osaka-u.ac.jp

d) takemura@ime.cmc.osaka-u.ac.jp

の四つに分類した [2].

- (1) ネットワーク接続の有無及び接続速度・ディスプレイやプリンタの有無などの計算機周辺コンテキスト
- (2) 位置や近くにいる人, 利用者自身の社会的地位などの利用者コンテキスト
- (3) 明るさ, 騒音レベル, 混雑度, 気温などの物理的コンテキスト
- (4) 日付, 時刻, 季節などの時間的コンテキスト

Chen らは, これらのコンテキストを認識し, その変化に対して人の介在なく計算機が応答するシステムを“コンテキストウェア”なシステムと定義している. コンテキストウェアシステムは, コンテキストの変化を認識した直後の振る舞いによって二種類に分類される. 変化の認識に応じて動作を変更するシステムを能動的コンテキストウェアシステム, 変化の認識に応じて利用者に新たな状況の通知や元の状況に戻ることを推奨するシステムを受動的コンテキストウェアシステムと呼ばれ, 近年盛んに研究されている [3], [4], [5], [6], [7], [8]

コンテキストウェアな携帯電話を実現するシステムとして, Siewiorek らの SenSay[3] がある. SenSay は, 3 軸加速度センサ・利用者の音声解析マイクロホン・周囲の騒音解析マイクロホン・可視光センサ・気温センサといった複数のセンサからのデータと, あらかじめ登録された利用者のスケジュール情報から, 利用者コンテキストを Uninterruptible(Busy)・Active・Idle・Normal の四つの状態のうちどれにあたるかを判断し, その状態に応じて, 着信音の大きさ・バイブレーションの有無などを自動で調整する.

利用者の現在位置から有用であると判断される店舗情報を選択し, 利用者が閲覧しやすいタイミングでその情報を配信する手法として林らの CoCo[4] がある. CoCo は, 利用者が身に付けている加速度センサと GPS, 紫外線センサなどからなるセンサアレイの情報から利用者コンテキストを推定し, あらかじめ登録した嗜好・目的を基にして, 利用者の持つ Personal Digital Assistant(PDA) に店舗を推薦し, 表示する. この研究では, 利用者が情報閲覧が可能であると考えられるタイミングを利用者コンテキストに基づいて決定し, 推薦する店舗情報を PDA に表示する. 利用者が望めば詳細な店舗情報とともに現在地と店舗を含む地図も表示させることが可能である. 推薦する店舗情報の決定は, 利用者コンテキストと利用者があらかじめ登録した嗜好による順位付けによって行っている.

スマートフォンや携帯電話でアプリケーションを選択, 起動する手間を減らすための技術や研究は多く存在する [9], [10], [11], [12]. 例えば利用者がアプリケーションの並び順を変更可能な方法や, 利用回数の多い物が上位に並び, Most Frequently Used (MFU) 法, 最近使用した物ほど上位に並び, Most Recently Used (MRU) 法などはす

に用いられている. 状況に応じてアプリケーションを推薦する研究として, 松本らは操作履歴を用いて状況に応じたアプリケーションを推薦するシステムを提案している [10]. アプリケーション毎の位置情報と時刻を三次元空間上にプロットすることでアプリケーションを利用する際の行動パターンを可視化し, 時刻と位置に依存して利用されるアプリケーションがあることを示した. 上坂らの研究 [12] では, 長期的に携帯電話の操作履歴を収集し, 曜日や位置情報, マナーモード状態, バッテリー残量, 電波強度といった属性の操作予測への寄与度を示している. この報告では時刻と直近の操作に関する属性が推定に有用であるとされている一方で, 位置情報はあまり有用でなかったとし, 緯度や経度から位置情報を抽出する方法に改善の余地があったとしている.

### 3. MMSB を用いたアプリケーション推薦手法

提案手法では, コンテキストはスマートフォンのセンサから得られた情報を前処理することで特徴量を抽出し, 複数の特徴量を組み合わせて表現される. 例えば「月曜日の朝に駅のホームで電車を待っている」という状況を考えると, この状況は, 曜日, 時間帯, 場所, 動作という特徴量のうち「月曜日」, 「朝」, 「駅」, 「静止している」という特徴が組み合わさって生起していると考えられる. 提案手法では, コンテキストの定義に人間による仮定を用いず, 特徴量の組み合わせの中から, 特定のアプリケーションの起動時に発生する組み合わせを抽出することでコンテキストとする. すなわち, 特徴量と特徴量の関係をグラフネットワークと考え, ネットワークのクラスタリングによってコンテキストを抽出する. 本手法ではノードが潜在的なグループの分布を持っており, さらにグループ間のリンク生成確率も考慮したモデルである MMSB をコンテキストの抽出と認識に用いる.

アプリケーションの推薦は, 現在の特徴量を用いてコンテキストのスコアリングを行い, その結果に基づいて行う. すなわち, コンテキストはそれぞれアプリケーションに関連づけられており, 各コンテキストのスコアに基づいて利用者が利用するであろうアプリケーションを推定する.

上記のように, センサとアプリケーションの履歴からコンテキストを学習する手法であるため, 本手法は MMSB を用いてモデルを作成する学習フェーズ, 順位づけられたアプリケーションの一覧を出力する推薦フェーズに分けられる. 学習フェーズでは, センサログから特徴量を求め, アプリケーションごとに隣接行列の生成, MMSB によるクラスタリング(ブロック生成)を行う. 推薦フェーズでは, センサの入力データから特徴量を求め, その特徴量を用いてアプリケーションごとにスコアリングし, スコアの高いものから順に並べ替えた一覧を推薦結果として出力す

	Mon	Tue	Wed	AM	Thu	Fri	PM	Sat	Sun
Mon	■								
Tue		■							
Wed			■						
AM				■					
Thu					■				
Fri						■			
PM							■		
Sat								■	
Sun									■

図1 ブロックモデルの例

る。本論文では学習及びスコアリングに用いる入力データの一単位をシーケンスと呼ぶ。

### 3.1 隣接行列の作成

起動したアプリケーションごとに学習用シーケンス群から特徴量を抽出し、各特徴量ごとに出現確率を求める。これを隣接行列として表現する。隣接行列は、特徴ベクトルの組み合わせの出現割合を値に持つ行列で、特徴量が  $N$  種類あったとすると  $N \times N$  行列となる。特徴量同士は独立であると仮定し、隣接行列を  $A$  とすると、各要素は式1で定義される。

$$A_{ij} = \frac{P(F_i, F_j)}{P(F_i)} \quad (1)$$

ここで、 $F_i$  を  $i$  番目に割り当てられた特徴量、 $P(F_i, F_j)$  を学習データ中で  $F_i$  と  $F_j$  が同時に出現した回数と定義し、 $F_{\text{all}}$  を取り得る全ての特徴量の集合として、 $P(F_i)$  を式2で定義する。

$$P(F_i) = \sum_{j \in F_{\text{all}}} P(F_i, F_j) \quad (2)$$

### 3.2 MMSB によるコンテキストの抽出

隣接行列を MMSB を用いてブロック化する。ブロック化された隣接行列はノード(特徴量)間のリンクがクラスタリングされ、複数のブロックとして存在する。このブロック一つ一つが、アプリケーションを起動するコンテキストを表現しているとみなす。

例えばあるアプリケーション  $\alpha$  を起動した学習用シーケンス群から得られた特徴量が、“月曜日の午前”、“火曜日の午前”、“水曜日の午前”、“木曜日の午後”、“金曜日の午後”であったとすると、隣接行列を作成し、MMSB を用いてモデル化すると、図1のように二つのブロックが生成される。一方が“月曜日、火曜日、水曜日、午前”の特徴量を含むブロックで、もう一方のブロックは“木曜日、金曜日、午後”の特徴量を含むと考えられる。このようにクラスタリングされたブロック一つが特徴のグループを作り、そのアプリケーションを選択するコンテキストを表現していると考えられる。

### 3.3 コンテキスト認識に基づく起動アプリケーションの推定

特徴ラベル間の関係が、アプリケーションにおいてどれだけ面積を持ったブロックに含まれているかをスコアとして、アプリごとに集計し、スコアの大きいものから順にソートした結果をシステムの推薦結果とする。つまり、一番スコアの高いアプリケーションが Rank 1 で、スコアの低いものが最低 Rank (= アプリの種類数) となる。

ブロックの面積が大きいほど、そのコンテキストを表現するのに必要な特徴が多いので、面積の大きいブロックを占める特徴が多いほど高いスコアをつけるように重み付けを行う。これは、例えば“金曜日、午後”という二つの特徴で表現されるコンテキストよりも、“月曜日、午前中、自宅の最寄り駅、走っている”という四つの特徴で表現されるコンテキストの方が限定的でより詳しい。入力となる特徴ベクトルが例えば“金曜日、午前中、走っている”であったときに、先の二つのブロックを占める割合はどちらも50%であるが、より詳細なコンテキストを表現していると考えられる後者の方が高いスコアになるように設計する。

以上の設計方針より、入力シーケンス  $l$  から抽出した特徴量を  $F_{\text{in}}$  としたときのアプリケーション  $\alpha$  のスコアを式3で定義する。ここで  $B_\alpha$  はアプリケーション  $\alpha$  のブロック化後の隣接行列のブロック数を意味する。

$$\text{Score}(\alpha, F_{\text{in}}) = \sum_{b \in B_\alpha} P(B = b | F_{\text{in}}) \quad (3)$$

ブロック番号  $b$  のスコアである  $P(B = b | F_{\text{in}})$  は式4で求める。 $F_b$  は  $b$  番目のブロックに含まれる隣接行列の要素の集合である。

$$P(B = b | F_{\text{in}}) = \frac{\sum_{i,j \in F_{\text{in}} \cap F_b} A_{ij}}{\sum_{i,j \in F_{\text{in}}} A_{ij}} \quad (4)$$

この定義に従い、入力シーケンスに対し、全てのアプリケーションについてスコアを求め、スコアの大きいものが上位になるようにソートされた結果を推薦結果として出力する。

## 4. 実装

上述の設計に基づき、アプリケーション推薦手法を用いるシステムを実装する。本研究ではスマートフォンとして iOS を搭載した iPhone4S を用いる。以降、提案手法の学習に用いるセンサデータのログを収集するためのアプリケーションと提案手法を用いたアプリケーション推薦システムについて述べる。

### 4.1 ログ収集アプリケーション

利用者が普段の利用シーンの中で起動したアプリケーションとセンサのデータを収集するため、ログ収集アプリケーションとしてランチャーアプリケーション(以下ロ



図 2 ロガーの画面例

ガーと呼ぶ)を実装した。ロガーは起動可能なアプリケーションをリスト状に表示し、利用者はリストの中からアプリケーションを起動し、同時にロガー起動中のセンサデータを収集し続ける。iOS の仕様により、他のアプリケーションの起動や、ホームボタン押下によってアプリケーションがバックグラウンドへ移行した後は位置情報以外のセンサデータを収集する事は困難であるため、本アプリケーションがフォアグラウンドに移行されてから他のアプリケーションを選択するまでの数秒間のセンサデータを収集する。本手法ではこの数秒間のデータを 1 シーケンスとして扱う。本実装で対象とするセンサ及び情報は、三軸並進加速度センサ、ジャイロセンサ、近接センサ、位置情報(GPS,wifi,基地局)、バッテリー残量、電波状況、画面の向き、電子コンパスとする。

図 2 にロガーの画面例を示す。アプリケーション推薦機構が実装されることを想定した作りになっており、この例では画面内にシステムが推薦したアプリケーションを順にリスト表示している様子を示している。画面下部のボタンにより、システムのアプリケーション推薦方法を切り替える事ができ、左から順に提案手法、利用者が並び順を設定できるブックマーク、使用頻度の多い物が上位に並ぶ MFU 法、最近使用した物ほど上位に並ぶ MRU 法、設定変更用ボタンとなっている。

#### 4.2 特徴量の抽出

特徴量を求めるにあたり、以下の処理を行う。連続値、実数値を持つデータについては、主に k-means により抽出したクラスターを属性として追加し、MMSB で利用できるデータへと離散化した。

- 利用者の位置情報  
 GPS を用いて得られた緯度、経度、緯度と経度の組、水平方向の精度、高度、垂直方向の精度、速度、移動

方向(方位)をそれぞれ 1 シーケンス内のデータで平均値、分散値、標準偏差値を求める。学習用データに k-means を用いてあらかじめクラスタリングしておき、入力シーケンスに対しても同様に平均値、分散値、標準偏差値を求め、それぞれどのクラスターに属するかを特徴量として用いる。

- 利用者の動作状況  
 加速度センサ、ジャイロセンサについて x,y,z 方向のデータのうち 1 シーケンス内のデータで平均、分散、標準偏差の値を求める。学習用データに k-means を用いてあらかじめクラスタリングしておき、入力シーケンスに対しても同様に平均値、分散値、標準偏差値を求め、それぞれどのクラスターに属するかを特徴量として用いる。
- 利用者の向いている方向  
 電子コンパスによる水平方向の方位、三次元方位、精度、磁場センサによる三次元磁場のセンサデータの値をそれぞれ 1 シーケンス内のデータで平均値、分散値、標準偏差値を求める。学習用データに k-means を用いてあらかじめクラスタリングしておき、入力シーケンスに対しても同様に平均値、分散値、標準偏差値を求め、それぞれどのクラスターに属するかを特徴量として用いる。
- スマートフォンの画面の向き  
 iOS の API で得られる Orientation(画面の向き)の値にそれぞれラベルをつけ特徴量として用いる。
- 時間帯  
 曜日や年、月、日、時、分、秒、タイムゾーンの値をそのまま特徴量として用いる。また、午前か午後か、平日か土日か、今年何日目か、今年何週目か、というデータを特徴量として用いる。
- 電池残量  
 電池残量は 1 シーケンス内で最新のデータ 1 件を用いる。学習用データに k-means を用いてあらかじめクラスタリングしておき、入力シーケンスに対しても同様に最新の 1 件がどのクラスターに属するかを特徴量として用いる。
- 近接センサ  
 iOS の API で得られる近接センサの反応状況から、近接センサが反応したときは“ProxOn”、反応しなかったときは“ProxNone”を特徴量とした。
- インターネットの接続状況  
 iOS の API で得られるインターネットへの接続状況にそれぞれラベルをつけ特徴量として用いる。また、SSID 名に前置文字列として“SSID”を付与したものの、BSSID 名に前置文字列として“BSSID”を付与したものを特徴量として用いる。無線 LAN へ接続されていない場合にはそれぞれ“SSIDNone”、“BSSIDNone”と

表 1 前処理により付加された情報

利用した情報とセンサ	前処理で取り出した情報
日時	曜日, 午前/午後, 年月日時分秒, 平日/土日, 日番号, 週番号, タイムゾーン
三軸並進加速度センサ	三次元位置方向の平均, 分散, 標準偏差のクラスタ
ジャイロセンサ	三次元位置方向の平均, 分散, 標準偏差のクラスタ
近接センサ	反応あり, 反応無し
画面の向き	画面の向き (上下左右縦横), 記録無し
バッテリー残量	電池残量 (割合) のクラスタ
電子コンパス	方位, 三次元方位, 精度の平均, 分散, 標準偏差のクラスタ, (精度については最小値, 最大値のクラスタを含む)
磁場センサ	三次元磁場の平均, 分散, 標準偏差のクラスタ
位置情報	二次元位置, 精度, 速度のクラスタ
電波状況	圏外, 3G 回線, 無線 LAN
SSID	無線 LAN のアクセスポイント SSID 及び BSSID

表 2 k-means における k の値

センサ情報	クラスタ数
加速度センサ (平均, 分散, 標準偏差)	3,10,50,100
ジャイロセンサ (平均, 分散, 標準偏差)	3,10,50,100
電子コンパス (三次元方位の平均, 分散, 標準偏差)	3,10,50,100
電子コンパス (方位の平均, 分散, 標準偏差)	2,4,6,8
電子コンパス (精度の平均, 分散, 標準偏差)	2,4,6,8
電子コンパス (精度の最小値, 最大値)	2,4,6,8
バッテリー残量	2,3,4,5,6
磁場センサ (三次元磁場の平均, 分散, 標準偏差)	3,10,50,100
位置情報 (速度)	2,4,6
位置情報 (精度の平均, 分散, 標準偏差)	2,3,4,5,6
位置情報 (二次元位置)	50,100,200

なる

以上の処理により抽出された特徴量を表 1 に示す。ここで、日番号、週番号とは、1 年のうち何番目の日や週であるかを示す。k-means でクラスタリングする際には、クラスタ数を複数設定し、それぞれのクラスタリング結果をすべて特徴量として用いた。クラスタ数は表 2 のように設定した。最終的に、特徴量は 892 種類となった。すなわち隣接行列は  $892 \times 892$  行列となる。

#### 4.3 隣接行列の作成

隣接行列の各要素の値は以下の式 5 によって得る。

$$A_{ij} = \frac{P(F_i, F_j)}{P(F_i)} = \frac{\sum_l R_l(i, j)}{\sum_j \sum_{l \in L_\alpha} R_l(i, j)} \quad (5)$$

ここで、 $L_\alpha$  はあるアプリケーション  $\alpha$  を起動したログに含まれるシーケンスの集合である。ただし  $F_i$  を  $i$  番目に割り当てられた特徴ラベルとして、 $R_l(i, j)$  は

$$R_l(i, j) = \begin{cases} 1 & \text{シーケンス } l \text{ で } F_i \text{ と } F_j \\ & \text{の両方が抽出されていた時} \\ 0 & \text{それ以外} \end{cases} \quad (6)$$

として定義する。ここでシーケンスの数とは、アプリケーション  $\alpha$  を起動したログの中でのシーケンス数である。

## 5. MMSB によるアプリケーション推薦の評価実験

今回は提案手法の有用性調査のために、収集したログを用いてオフラインでパーソナルコンピュータを用いて実装、実験を行った。提案手法がアプリケーション推薦として有用かどうか調べるための評価実験を行った。MMSB には、統計処理言語 R の lda パッケージ [20] 内の関数 `mmsb.collapsed.gibbs.sampler` を用いた。比較対象となる従来手法として、MFU 法、MRU 法による推薦結果をあわせて示す。

### 5.1 実験内容

スマートフォンのアプリケーション利用履歴を収集するために、データ収集を行った。被験者は二十代男性の大学院生一名で、収集期間は約六ヶ月である。被験者の利用するスマートフォン (iPhone4S) に、ロガーをインストールし、ロガーから起動可能なアプリケーションについてはできるだけロガーから起動させた。収集期間終了後にデータを回収し、解析を行った。

ブロックモデルの学習には、各アプリケーションを起動したログのシーケンス群のうち十分の一をテストデータとしてサンプリングし、残りを学習データとした。MFU 法、MRU 法での推薦には、初めて起動したアプリケーションについては Rank を求めず学習のみとした。

本実験では利用者のある状況に対して、起動される確率の高い順にアプリケーションを順位付け推薦した時に、利用者が実際に選択し起動した正解のアプリケーションがシステムの推薦した順位のうち何番目に存在したかを指標とし、これを “Rank” と呼ぶ。

### 5.2 実験結果

収集したデータについて解析する。アプリケーションごとの起動回数が 10 回よりも少ないアプリケーションにつ

いては学習量が少なくなりすぎるため、学習及び推薦に利用しないこととした。これにより、推薦に利用するアプリケーションを 30 種類に絞り込んだ。この 30 種類のアプリケーションのアプリケーション ID と起動回数の対応を表 3 に示す。

図 3 に MFU 法, MRU 法, 提案手法 (MMSB) のアプリケーションごとの平均 Rank と起動回数を示す。図 3 では縦軸は平均 Rank を、横軸はアプリケーション ID をアプリケーションの起動回数の多い順を示している。右端の Avg の行は、平均 Rank の平均値を示している。平均 Rank の平均値は、全アプリケーションの平均 Rank の平均値を求めたものである。MFU 法, MRU 法, MMSB それぞれの平均 Rank の平均値はそれぞれ 13.0 位, 7.4 位, 14.8 位となった。

### 5.3 考察

提案手法は全体の平均 Rank で比較すると MFU 法, MRU 法より良いとは言えないが、一部のアプリケーションについては推薦精度が良いことがわかる。MFU 法や MRU 法

は起動回数が多ければ多いほど上位に推薦する傾向にあるが、提案手法では起動回数が少なくても上位に推薦できるアプリケーションがあることがわかる。

表 4 に提案手法の平均 Rank が MFU 法, MRU 法より高いアプリケーションを示す。また、表 6 に挙げたアプリケーションでは MFU 法の平均 Rank が高く、表 5 に挙げたアプリケーションでは MRU 法の平均 Rank が高かった、提案手法において平均 Rank が高かったアプリケーションについて詳しく見ると、表 4 に上げたアプリケーションは全て、起動回数が 50 回以下と少なく、どれも MFU 法, MRU 法では上位に推薦することが難しいアプリケーションと言える。厳密には、各アプリケーションごとに、セン

表 4 MMSB が上位に推薦したアプリケーション

App. ID	Name	Count	提案手法	MRU	MFU
51	TheFuture.fm	17	1	8.3	18.9
41	Photosynth	13	1	10.5	18.5
32	8tracks	13	3	9.5	17.4
12	Twitter	37	9.3	11.5	15.8
61	僕の来た道	26	10.5	13.1	21.2

表 5 MRU が上位に推薦したアプリケーション

App. ID	Name	Count	提案手法	MRU	MFU
10	Instagram	516	14	3.5	4.3
63	Draw Free	11	9	4.5	21.1
13	レーダー・ ナウキャスト	20	13.5	4.8	6.7
57	スヌービー	134	17.5	5.9	12.3
83	Camera!	57	23	6.2	20.8
72	Tumblr	47	15	6.6	22.9
68	Draw Free	52	7.6	6.7	16.2
28	Evernote	114	24.4	7.2	8.8
80	Offline Pages	37	20	7.7	22.8
16	Skype	52	20	7.9	9.5
49	Staccal	90	22	7.9	11
9	Notifier	26	23.5	8	7.7
42	t.free	54	11.6	8.5	12.9
30	Music	72	17.7	8.5	14
69	Tweet	44	26	8.8	18.8
58	MapFan	24	13	9.2	20.3
2	Facebook	64	15.5	9.5	12.7
29	Maps	23	19.5	10	15.5
25	Chrome	14	23	10.9	17.4

表 6 MFU が上位に推薦したアプリケーション

App. ID	Name	Count	提案手法	MRU	MFU
1	foursquare	1351	13.3	1.9	1
11	Runkeeper	465	4.4	3.8	3.2
7	Producteev	311	18.8	4.8	4.4
14	乗り換え案内	137	12.3	7.6	6.2
8	ぐっすり〜ニャ	194	7.4	7.2	6.6
0	TwitbirdFree	1305	28.4	2	2

表 3 アプリケーション ID とアプリケーション名, 起動回数の対応

ID	アプリケーション名	起動回数
1	foursquare	1351
0	TwitbirdFree	1305
10	Instagram	516
11	Runkeeper	465
7	Producteev	311
8	ぐっすり〜ニャ	194
14	乗換案内	137
57	スヌービー	134
28	Evernote	114
49	Staccal	90
30	Music	72
2	Facebook	64
83	Camera!	57
42	t.free	54
68	Draw Free	52
16	Skype	52
72	Tumblr	47
69	Tweet	44
80	Offline Pages	37
12	Twitter	37
9	Notifier	26
61	僕の来た道	26
58	MapFan	24
29	Maps	23
13	レーダー・ ナウキャスト	20
51	Thefuture.fm	17
25	Chrome	14
41	Photosynth	13
32	8tracks	13
63	Draw Free	11

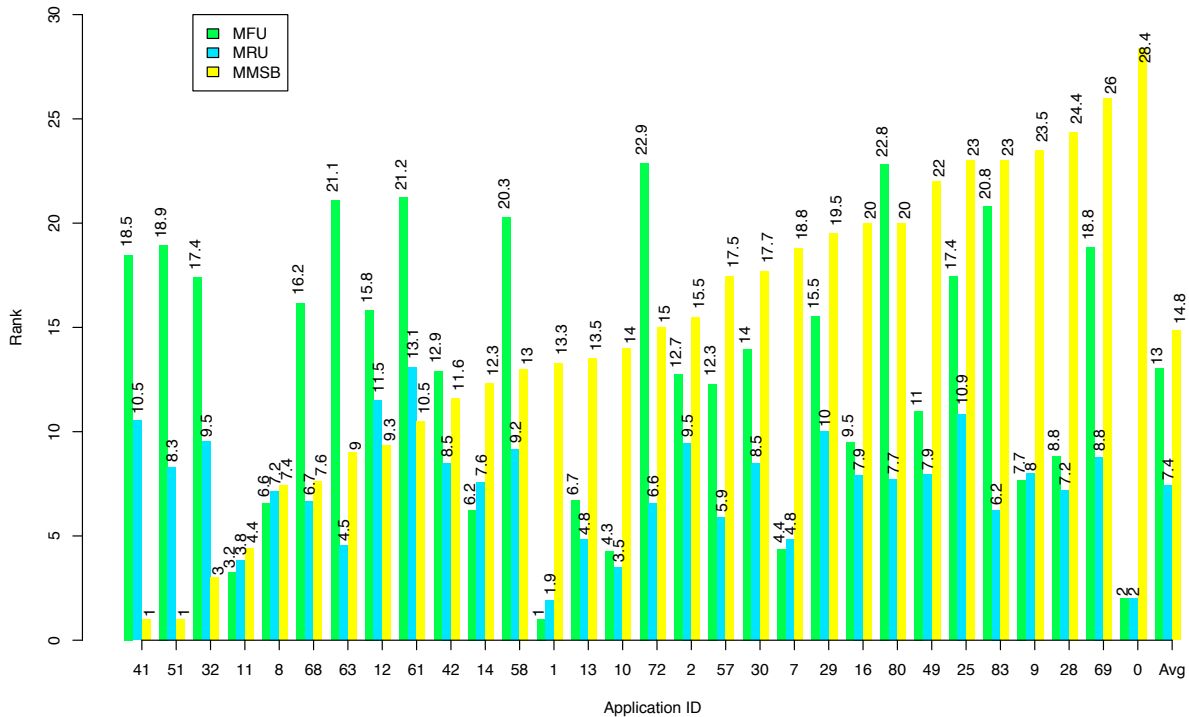


図3 MMSB を用いた場合の平均 Rank と MFU 法, MRU 法を用いた場合の平均 Rank の比較 (MMSB の平均 Rank によるソート)

サデータの平均値のクラスタや、方位のクラスタ、ジャイロセンサの分散値のクラスタなどにそのアプリケーション固有の特徴や条件が存在すると考えられる。アプリケーション ID:41 の Photosynth は、パノラマ写真を取ることのできるアプリケーションで、旅行先で利用することが多かった。その結果長期休暇中である8月、9月や、休日にデータが偏っていたこと、自宅や職場といったよく訪れる場所ではない位置での利用が多いこと、充電器につながっていないことなどがコンテキストとして特徴的であったと考えられる。アプリケーション ID:51 の TheFuture.fm とアプリケーション ID:32 の 8tracks はインターネット上の音楽を視聴するアプリケーションで、移動中に聞くことが多いが、地下鉄に乗っているときにはインターネットが利用できないため起動することは無い。また TheFuture.fm については7月、11月の起動が多い。といった傾向が見られ、それぞれコンテキストとして表現しやすい状況にあったと考えられる。アプリケーション ID:12 の Twitter は、Twitter クライアントであるが、普段は起動回数から明らかなようにアプリケーション ID:0 の TwitbirdFree を利用している

ため、このアプリケーションを起動する状況や時期が限定されていたと考えられる。具体的には、6月と11月の利用回数が多かった。アプリケーション ID:61 の僕の来た道は、日々の移動を記録するアプリケーションで、普段はバックグラウンドで起動しており GPS 情報を記録し続けている。このアプリケーションは自宅にいる間に起動することが多く、また月曜日と火曜日の起動回数が多かった。このように、アプリケーション毎に利用時期が限定されていたり、特定の条件が組合わさった状況でよく利用されるアプリケーションについては推薦の精度が良くなることがわかった。逆に、アプリケーション ID:0 の TwitbirdFree のように起動回数が極端に多いアプリケーションについては状況によらず起動されると言えるため平均 Rank が低かった。これは MMSB のブロック数 B の設定が少なすぎ、表現しようとするコンテキストの数が制限されてしまったためであると予想される。

この被験者の利用傾向では、30個中20個のアプリケーションについて MRU 法による推薦精度が最も良かった。これより、MMSB に用いる特徴量として“直前にどのアプリケーションを起動したか”を追加することや、スコアリ

ング手法を、前回そのアプリケーションが起動されたのはいつだったかを考慮するように改善することでより MMSB による推薦精度が高まると期待できる。またそれぞれの特徴量についても推薦への寄与度合いが異なり、ノイズとなっている特徴量も存在すると考えられるため、どの特徴量を利用すれば(しなければ)より精度が高くなるかを調査する必要があると考えられる。

## 6. おわりに

本論文では、スマートフォン利用者にコンテキストに応じたアプリケーションを推薦する手法として、各種センサを用いたコンテキストログを収集し学習、推薦する手法を提案した。提案手法では、コンテキストの定義に人間による仮定を用いず、特徴量の組み合わせの中から、特定のアプリケーションの起動時に発生する組み合わせを抽出することでコンテキストとする。そして、現在の特徴量を用いて各アプリケーションに対応付けられているコンテキストのスコアを求め、スコア順に並んだアプリケーションリストをシステムの推薦結果とする。

提案手法の有効性を評価するため、スマートフォンで利用できるセンサのログを収集するアプリケーション及びアプリケーション推薦システムを構築し、センサログの収集を行った。収集したログを用いて学習と実験を行い、提案手法と MFU 法、MRU 法とを正解アプリケーションの Rank で比較する実験を行った。MFU 法、MRU 法、提案手法それぞれの平均 Rank の平均値はそれぞれ 13.0 位、7.4 位、14.8 位となった。提案手法は全体の平均 Rank で比較すると MFU 法、MRU 法よりは良いとは言えないが、特定のアプリケーションについては推薦精度が良いことがわかった。今後の課題として、各パラメータを自動的に決定する手法の開発や、位置情報のクラスタリング手法の検討、特徴量の改良などが上げられる。

## 謝辞

本研究の一部は、「文部科学省グローバル COE プログラム (研究拠点形成費)」による。

## 参考文献

- [1] Airoldi, E. M., Blei, D. M., Fienberg, S. E. and Xing, E. P.: Mixed Membership Stochastic Blockmodels, *J. Mach. Learn. Res.*, Vol. 9, pp. 1981–2014, (2008).
- [2] Chen, G. and Kotz, D.: A Survey of Context-Aware Mobile Computing Research, *Technical Report TR2000-381, Dept. of Computer Science, Dartmouth*, (2000).
- [3] Siewiorek, D., Smailagic, A., Furukawa, J., Krause, A., Moraveji, N., Reiger, K., Shaffer, J. and Wong, F. L.: SenSay: A Context-Aware Mobile Phone the 7th IEEE International Symposium on Wearable Computers, ISWC '03, pp. 248–249, (2003).
- [4] 林智天, 川原圭博, 田村大, 森川博之, 青山友紀: 小型モバイルセンサを用いたコンテキスト適応型コンテンツ配

- 信サービスの設計と実装, 電子情報通信学会技術研究報告. 情報ネットワーク, Vol. 104, No. 691, pp. 149–154, (2005).
- [5] 中西泰人, 辻貴孝, 大山実, 箱崎勝也: Context Aware Messaging Service: 位置情報とスケジュール情報を用いたコミュニケーションシステムの構築および運用実験, (<特集>ITS とモバイルコンピューティング). 情報処理学会論文誌, Vol. 42, No. 7, pp. 1847–1857, (2001).
- [6] 今西孝也, 久住憲嗣, 中西恒夫, 北須賀輝明, 福田晃: コンテキストウェア電子メール配送システム, 情報処理学会第 65 回全国大会, Vol. 1H-1, pp. 223–224, (2003).
- [7] 鈴木啓章, 加藤俊一: ウェアラブル環境での利用者への控えめな情報提示とそれに対する自然な反応の観察による興味・関心の計測・モデル化, 映像情報メディア学会, Vol. 34, No. 18, pp. 47–50, (2010).
- [8] Ishida, S. and Kawaguchi, K.: Context-Aware Information Recommendation Method using Personalized Server System. Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO2009), pp. 1162–1170, (2009).
- [9] 巖 康平, 福井健一, 森山甲一, 沼尾正行, 栗原 聡: 携帯電話の利用傾向に基づくアプリケーション推薦法の提案, 人工知能学会全国大会, Vol. 25, 人工知能学会, pp. 1–4, (2011).
- [10] 松本光弘, 清原良三, 沼尾正行, 栗原 聡: 携帯電話におけるユーザの操作パターンを用いたアプリケーション推薦方式の提案, マルチメディア, 分散, 協調とモバイルシンポジウム DICOMO2009, pp. 1149–1155, n (2009).
- [11] Yano, M., Kaji, K. and Kawaguchi, N.: App.Locky: Users' Context Collecting Platform for the Context-aware Service Recommendation, *IPJS Journal*, Vol. 52, No. 12, pp. 3274–3288, (2011).
- [12] 上坂大輔, 岩本健嗣, 村松茂樹, 横山浩之: 携帯電話における長期的操作履歴の解析と状況適応型操作予測技術の検討, 情報科学技術フォーラム講演論文集, Vol. 8, No. 4, pp. 23–29, (2009).
- [13] Nowicki, K. and Snijders, T. A. B.: Estimation and Prediction for Stochastic Blockstructures, *Journal of the American Statistical Association*, Vol. 96, No. 455, pp. 1077–1087, (2001).
- [14] Kemp, C., Tenenbaum, J., Griffiths, T., Yamada, T. and Naonori, U.: Learning Systems of Concepts with an Infinite Relational Model, *Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, pp. 381–388, (2006).
- [15] Kononenko, I.: On Biases in Estimating Multi-Valued Attributes., *International Joint Conference on Artificial Intelligence*, pp. 1034–1040 (1995).
- [16] Kirmse, A., Udeshi, T., Bellver, P. and Shuma, J.: Extracting patterns from location history, *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, New York, NY, USA, ACM, pp. 397–400, (2011).
- [17] 奥 健太, 中島伸介, 宮崎 純, 植村俊亮, 加藤博一: 情報推薦におけるユーザの価値判断基準モデルに基づくコンテキスト依存型ランキング方式, 情報処理学会論文誌. データベース, Vol. 2, No. 1, pp. 57–80, (2009).
- [18] Masinter, L., Berners-Lee, T. and Fielding, R. T.: Uniform Resource Identifier (URI): Generic Syntax, Technical Report 3986, RFC Editor, Fremont, CA, USA (2005).
- [19] Masinter, L., Hoffman, P. and Zawinski, J.: The mailto URL scheme, Technical Report 2368, RFC Editor, Fremont, CA, USA (1998).
- [20] Chang, J.: *lda: Collapsed Gibbs sampling methods for topic models*. (2012). R package version 1.3.2.