

個人を見守るサーバシステムに基づく コンテキストウェアな情報提示手法

石田将吾[†] 河口信夫[†]

近年、膨大な情報源の中からユーザに適した情報を見つけ出し、それを提示する推薦技術が注目されているが、十分な個人化はなされていない。本研究では、ユーザ情報を基に推薦情報の収集・計算を行い、個人の状況に応じて柔軟な提示を行うコンテキストウェアな推薦システムを提案する。嗜好だけでなくコンテキストにも依存した推薦情報をユーザは受け取ることができ、推薦情報の緊急性に適した提示方法を用いることで、ユーザの状況に応じた端末に対して、必要な情報を必要な時に知らせることを可能とする。

Context-Aware Information Recommendation Method using Personalized Server System

SHOGO ISHIDA[†] NOBUO KAWAGUCHI^{††}

Recently, we have several information recommendation systems. However, they are not fully context-aware and personalized. In this paper, we propose a context-aware information recommendation system which fully personalized. By using personal server, one can obtain suitable information at anytime, anyplace by proper method.

1. はじめに

近年、膨大な情報源の中からユーザに適した情報を見つけ出し、それを提示する推薦技術が注目されている。しかし、従来の推薦システムはユーザの嗜好のみに依存したものが殆どであり、また、推薦の提示方法もメール配信のみであったり、推薦サービスが携帯電話でのみ利用可能であるなど択一的なものが多い。例えば、推薦情報として「今日の天気」と「居酒屋情報」を提示する場合、朝に「居酒屋情報」を提示することは不適切であるし、深夜に「今日の天気」を提示することも適切ではない。そのため、提示する情報はユーザを取り巻く状況として時間や位置など所謂コンテキストを考慮する必要がある。また別の例として、「地震速報」などのような緊急情報は

きるだけ急いで提示して欲しいが、「グルメ情報」など不急な情報は毎日メールで提示されると煩わしく感じることもある。そのため、推薦情報の緊急性に応じた提示方法の変更や、携帯端末やパソコンなど既存のインフラに柔軟に対応するシステムが必要である。

本研究では、ユーザ情報を基に推薦情報の収集・計算を行い、必要に応じて柔軟な提示を行うコンテキストウェアな推薦システムを提案する。嗜好だけでなくコンテキストにも依存した推薦情報をユーザは受け取ることができ、推薦情報の緊急性に適した提示方法を用いることで、ユーザの状況に応じた端末に対して、必要な情報を必要な時に知らせることを可能とする。さらに、推薦に必要な計算をサーバで行うことで、端末の計算システムを消費することなく応用的な計算が実現でき、ユーザ情報、推薦情報をデータベースで保持することを実現する。

実装では、GPS機能を持った携帯端末であるAndroid G1, iPhoneをユーザ情報取得する端末とした。また、サーバではCGI, データベースを構築し、提案した推薦システムを実装した。これらにより、本研究で提案した個人に特化した推薦システムが実生活で利用可能であることを確認した。

以下、2節では従来の推薦技術及びその研究を取り上げ、その問題点、課題を述べる。3節では課題を考慮した推薦システムの提案を述べる。4節では提案の実装手法を述べる。5節では提案手法、実装に関する考察、類似システムとの比較を行う。6節では本研究のまとめ及び今後の課題を述べる。

2. 既存の情報推薦・提示手法

情報推薦システムには大きく分けて3つの段階がある。1つ目は、ユーザの嗜好やコンテキストを認識、把握するためのユーザプロファイリングであり、2つ目はユーザプロファイルからそのユーザにとって適切な情報を選択する情報フィルタリング、3つ目は選択した情報をユーザに提示する情報提示である。本節では、それぞれの段階における既存の手法や研究を挙げ、また従来手法の問題点を述べる。

2.1 ユーザプロファイリング

情報推薦は、ユーザの嗜好や興味に基づき情報の提示を行う。そのユーザ情報取得する技術をユーザプロファイリング技術と呼ぶが、それは大きくは、明示的（直接的）手法と暗黙的（間接的）手法の2種類に分けられる[8]。ここでは2種類の手法について解説する。

2.1.1 明示的手法

明示的手法は、ユーザから直接に嗜好やコンテキストに関する情報を入力してもらうものである。この手法には大きく2種類に分けられ、(i) ユーザの嗜好やコンテキストに関して、アンケートに答えさせ、キーワードの形で情報を取得する方法、(ii) 見

[†]名古屋大学大学院工学研究科

たもの、聞いたもの、行ったことに関してどれだけ興味があったか数段階評価でアンケートを行い嗜好を取得する方法がある。(i) の例としては SIFT[9] がある。これは、ユーザが興味あるトピックをユーザプロフィールとして記述し、それをサービス提供者にメールで送ることで、情報推薦サービスを受けるものである。ユーザプロフィールは一般的なベクトル空間でモデル化され表される。(ii) の例としては GroupLens[10] がある。これは、ユーザがニュース記事に対し評価値を与え、それをもとにユーザプロフィールを行うものである。明示的手法はユーザプロフィールの作成が容易なため、初期の推薦システムでは多く取り上げられたものの、ユーザへの負担を課すため、最近の研究では取り上げられることは少ない。

2.1.2 暗黙的手法

暗黙的手法は、ユーザの行動、履歴などから嗜好に関する情報を取得するものである。この手法は(i) ユーザのユーザが行った行動の対象物全てに対し興味を持ったと仮定し、その履歴を用いる方法と、(ii) ユーザの履歴において何らかの方法でユーザが興味を持ったかを判定する方法の大きく 2 種類に分けられる。(i) では、代表的なシステムとして WebWatcher[11] や WebMate[12] などがあり、これらは閲覧した Web ページを平等に興味を持ったページとして扱い、評価付けを行うものである。(ii) の代表的なシステムとして森田らの研究[13] があるが、これはネットニュースの記事との閲覧時間を観測しているもので、閲覧時間とユーザにとっての記事の有用さの度合いと不要さの度合いとの相関関係を示し、プロフィールを作成している。

暗黙的手法は、行動や履歴からプロフィールを作成するためユーザへの負担が少ない。しかし、] 価のように確実なユーザ情報を得られるわけではないので明示的手法に比べプロフィールは困難である。

2.2 情報フィルタリング

ユーザプロフィールによって得られた情報を用いて、ユーザに適した情報を選択する必要がある。その選択技術を情報フィルタリングと呼ぶが、これには、コンテンツに基づくフィルタリングと協調フィルタリングの 2 種類がある[8]。ここでは 2 種類の手法について解説する。

2.2.1 コンテンツに基づくフィルタリング

コンテンツに基づくフィルタリングは、対象とするコンテンツに基づき、ユーザに推薦する情報を選択する手法である。この手法では、マルチメディアデータを対象とした場合、フィルタリングに用いる特徴量を抽出することが困難なため、テキストデータを対象とした研究が多く、マルチメディアデータを対象とした研究は少ない。代表的なシステムには WebWatcher[11] や WebMate[12] などがあり、これらはユーザの閲覧した Web ページ履歴からユーザプロフィールを作成し、そのプロフィールに基づいて情報推薦を行うものである。また、音楽情報フィルタリングシステムとして岩濱らの研究[14] があるが、これはユーザが音楽データに対して評価を行い、決定木を作

成し、それを用いて推薦を行うシステムである。この手法はユーザに特化した情報を選択するため、後述する協調フィルタリングに比べ精度は期待できる。しかし、フィルタリングに必要なユーザプロフィールの情報量は多くなるため、Web 上で展開されているサービスでこの手法を取り入れているものは少ない。

2.2.2 協調フィルタリング

協調フィルタリングは、ネットワーク上に存在する類似した好みを持ったユーザを発見し、その類似ユーザが共通して好む情報を選択する手法である。代表的なシステムに Amazon.com[15] の推薦機能があるが、これはユーザがある商品を開覧すると「この商品を買った人はこんな商品も買っています」というように、他のユーザの購入履歴から商品を推薦するものである。また、ネットニュースを対象とした協調フィルタリングシステムである GroupLens[10] では、嗜好が類似するユーザの評価値から、対象ユーザのニュース記事に対する評価値を推測し、ニュース記事を推薦する。このシステムを映画の推薦に応用した MovieLens[16] なども存在する。この手法は、フィルタリングに必要なユーザプロフィールの情報量はコンテンツに基づくフィルタリングに比べると少なくとも行うことができるが、類似するユーザ情報が少ないと満足なフィルタリングができない。また、他のユーザの嗜好に基づいたフィルタリングであるため、厳密にはユーザに特化した情報フィルタリングではない。

2.3 情報の提示手法

情報の提示手法には大きく分けると(a) Push 型、(b) Pull 型の 2 種類の手法がある。

Push 型は、ユーザの意思に関係なく、ユーザの下まで情報が提示が届けられるものである。具体的なものとしては広告などのダイレクトメールがこれにあたる。重要な情報などの提示では必要となるが、ユーザにとって緊急を要さない場合はユーザにとって負担となることが多い。Pull 型はユーザが明示的に情報を取得しに行くものである。具体的なものとしては Web ページ上の推薦情報がある。この手法は、ユーザの能動性を尊重するためユーザの負担になることはないが、急を要する情報の場合、必要な時に提示できない可能性をはらんでいる。

2.4 コンテキストに依存した推薦

ユーザに推薦すべき情報はコンテキストによって大きく変化する。例えば、現在地とは関係ない地区の天気を提示することは適切とは言えず、また、朝から周辺の居酒屋店舗情報を提示することは適切ではない。推薦システムは、ユーザのコンテキストを取得し、それに合わせた情報の提示をする必要があると考えられる。しかし、従来推薦手法の主流は、Web ページや音楽などといったユーザのコンテキストに関係なく嗜好のみに依存して提示できるものである。ここでは、推薦に必要なコンテキスト、また従来のコンテキストに依存した推薦システムの研究を紹介する。

2.4.1 推薦に必要なコンテキスト

コンテキストとは、一般にユーザを取り巻く状況、背景、前後関係などを意味する。

ここでは、奥らの研究[21]で定義された「人の情報選択の判断に対する意思決定に影響を及ぼす状況」としてのコンテキストの例をあげる。

- 時間情報：年月日、季節、曜日、時刻
- 気象情報：天気、気温、湿度、花粉情報、緊急気象情報（台風、津波など）
- ユーザ情報：休日であるか、予算、余裕時間、目的、気分、体調
- 同伴者情報：関係（家族や恋人、友人、上司・部下など）、人数
- 周辺環境：所在地のエリアタイプ（歓楽街や郊外、観光地など）、周辺店舗の分布状況、混雑状況
- 時系列的要因：過去の行動履歴、将来の行動予定

推薦情報には上述した要素は考慮する必要があると考えられる。

2.4.2 コンテキスト依存型推薦技術

アプリケーションとしてユーザのコンテキストに依存した様々な研究[17]が行われている。例えば、店内の買い物客の位置情報に基づいて、商品情報を提示する Shopping Assistant[18]や、現在時刻や旅行客の位置に基づいて観光ガイドなどを提示する Cyberguide[19]などがある。しかし、これらの研究は、特定の場所でのみ情報の推薦を受けることができ、あらゆる場所で使用できるものではない。また、ユーザのコンテキストに依存した情報の提示を行ってはいないが、ユーザの嗜好に依存した推薦は行っていない。それに対し、矢野らの研究[20]では、ユーザが興味を持った情報に対して情報タグをストックし、同時に位置やショップ情報などライブなメモを作成することにより的確な情報提供やレコメンドなど、購買行動のサポートを行っている。また、奥らの研究[21]では、コンテンツデータを p 次元、コンテキストを q 次元の特徴空間上に配置し、それを SVM を用いてクラス分類することによりコンテキストに応じて変化するユーザの嗜好を適切にモデル化する手法を提案している。しかし、これらの研究では、ユーザにとって適切な情報の推薦は受けることができるものの、特定のインフラを用いる必要があったり、入力すべき情報が多すぎるなどといった、ユーザの負担を増やすことが考えられる。

2.5 既存の情報推薦システム

表 1：既存の情報推薦システムの比較

既存手法	コンテキスト依存	PC	携帯端末	プロファイル方式	フィルタリング方式	情報提示方式
i コンシェル [22]	△	×	○	明示+暗黙	コンテンツに基づくフィルタ	Pull+Push
clon[23]	×	×	○	明示	協調フィルタ	Pull
Amazon[15]	×	○	△	暗黙	協調フィルタ	Pull+Push

既存の推薦システム(表 1)に関し、上記の項目について比較する。

i コンシェル[22]は NTTdocomo が開発した携帯電話向けサービスである。配信する情報は、ユーザが入力した端末内の情報や i モードからの情報であり、例えば、スケジュールを見ると一週間の天気や、登録しておいたコンテンツ情報（テレビ番組情報やライブ情報など）などが表示される。また、それらの情報は自動的に更新され、サイトや店舗で取得したクーポンなども自動で更新される。鉄道における運転見合せなどの情報や交通における渋滞などの情報、また、気象情報や地震情報など緊急情報も配信される。これらの情報には、ユーザが明示的に欲した場合に配信されるコンテンツ情報と、ユーザの能動性に関係なく配信される緊急情報がある。グルメ情報、ショッピング情報などのコンテンツ情報は、ユーザが登録をしたときのみの配信であるので、興味をもったコンテンツのみを受け取ることができる。また、交通情報、気象情報などの緊急情報は、ユーザの登録など関係なく配信される。また、コンテンツ情報はスケジュールに書き込まれるなど、基本的には Pull 配信なのに対して、緊急情報は着信音が鳴り、待ち受けに表示されるなど Push 配信である。

しかし、情報を受け取る端末が常に携帯電話を使うため、情報量に限界が生じると考えられる。例えば、ユーザがパソコンを使用している状況では、携帯電話を用いた情報の提示よりも、パソコンを用いた情報の提示の方が、より多くの情報が配信でき、ユーザにとって適切な情報を与えやすいと考えられる。また、緊急情報はコンテキストに依存して情報を提示されているものの、その他のコンテンツはユーザが明示的手法でキーワードを与え、それに関する情報が表示されるというシステムのため、推薦システムとは言い難い。

clon[23]も、携帯電話向けサービスだが、i コンシェルとは異なり、待ち受け、スケジュール画面に情報が配信されるわけではなく、専用の Web ページでのみ情報を受け取ることができる。それゆえ、完全な Pull 型配信システムといえる。そして、ユーザの嗜好を考慮した推薦情報の選択を行うために、嗜好の取得も明示的手法で行っている。しかし、clon では、ユーザの分身となる自分だけの「クロン」を作成し、そのクロンと会話する形式でユーザの嗜好を取得する。例えば、クロンと会話するという表示に進むと、クロンから適当な質問、「好きな漫画は？」や「どこに行きたい？」などと問われる。それに対し、ユーザがそれに答えることで回答の中のキーワードから協調フィルタリングを行い、情報を推薦するというものである。完全な Pull 型配信であるため、ユーザは好きな時間にクロンと話したり、情報を受け取ることができる。また、嗜好の取得方法も、明示的手法というユーザに負担を与えるものではあるが、クロンと会話するというスタイルをとることで、入力というユーザにとっての手間を楽しむ時間に変えている。ユーザプロファイリングにおいて、最も難があると思われる嗜好の取得を、ユーザを楽しませることによって解決した試みは面白いものの、コンテキストの認識、提示手法の柔軟性はなく、推薦システムとしては弱いところが多い。

Amazon[15]はネット通販での「おすすめ商品」や「この商品を買った人はこんな商品も買っています」といった機能である。実用レベルにある推薦システムにおいては最先端にあると思われる。このシステムは、ユーザアカウントによって得られる購入履歴、閲覧履歴などを用いることにより、それぞれの商品において、同時に購入、閲覧された商品に関連商品とみなし推薦するものである。また、ユーザ自身がアカウントを作成することにより、その購入履歴などからユーザプロフィールを作成し、メール、あるいはホームページ上でおすすめ商品を提示することもできる。

このシステムは、「普通の商品」に対しては満足される推薦が行われていると言える。しかし、閲覧、購入頻度の少ない商品では十分な協調フィルタリングが行われないため、的外れな結果が提示されることもある。また、配信方法では、メールによる情報の配信、あるいは amazon のホームページにおける提示など、Push 型配信、Pull 型配信をユーザが選ぶことができるため、不満の少ないシステムになっていると言える。しかし、基本的なユーザプロフィールは協調フィルタリングで行われているため、ユーザの嗜好に対して十分に解析された推薦は行われない。また、他人の購入履歴などを自動的に利用するため、プライバシーの問題も叫ばれている。

2.6 既存推薦システムの問題点

既存の推薦システムは、ユーザの嗜好のみに依存したものが一般的である。しかし、ユーザにとって有用な情報とは嗜好とコンテキストに依存した情報であり、その提示が推薦には必要である。コンテキストとは位置や時間だけでなく、状況、例えば、誰といるのか、何人であるのかなどの情報や予算がない、病気でできることが限られているなどの要素も考えられる。また、時系列で見たときに必要になる情報も考えられ、例えばユーザが新幹線に乗っているとシステムが認識した場合、降車駅の情報をあらかじめ提示することも可能であると考えられる。

情報の提示方法は、一方的にユーザに情報を提示するものが多い。しかし、情報によってその緊急性は異なり、例えば交通情報や緊急速報など、ユーザに急いで知らせるべき情報などはメールや何らかの形でユーザの注意を引く Push 型の配信をすべきであるし、そうでなければ、ユーザの能動性を尊重した Pull 型の配信を行うなど、情報によって配信方法は変える必要がある。また、情報の提示は位置や時間、状況などに囚われない手段で行うべきである。特定の場所や時間でしか使えない手段では常に有用な情報をユーザに提示することはできず、また、特定のインフラを必要とする手段でも不便をしいられることとなる。既存のインフラ、例えば PC や携帯電話などの端末機器で同じように使え、配信を享受できる必要がある。

以上より、従来手法における問題点は以下のようにまとめられる。

- 推薦情報がユーザのコンテキストに依存していない
- ユーザプロフィールに必要な情報が多い、ユーザへの負担が大きい
- フィルタリング手法が択一的であり、情報に対する柔軟性がない

- 提示手法 (Push 型, Pull 型) がユーザの能動性、情報の緊急性に対応していない
- 推薦を享受できる端末が限られている (携帯, パソコンのみ・特有のインフラが必要)

次節より、上にあげた問題点を考慮した推薦手法について述べていく。

3. 有用情報提示システムの提案

ユーザの嗜好、コンテキストの取得をその性質によって柔軟に行い、また、推薦情報の選択にサーバを用いることで、推薦の提示方法を、ユーザの端末や情報の性質によって変化させる推薦システムの提案を行う。

3.1 システムの概要

推薦システムを以下の3つの構成に分ける。

- i. ユーザ情報の取得
- ii. 推薦情報の収集、選択
- iii. 推薦の提示、配信

(i)では、ユーザの嗜好、コンテキストの取得を目的とする。情報取得に際しては携帯端末、具体的には携帯電話やスマートフォンを用い、内蔵されているセンシングデバイス、またはアンケート形式の質問によりユーザ情報を取得する。また、ユーザの家や会社、学校などの情報もコンテキストから計算し、ユーザ情報として取得する。(ii)では、(i)で取得したユーザ情報を基に推薦する情報の収集、選択を行う。ユーザ情報をサーバに送信することにより、情報の収集、選択はサーバを用いて行う。これにより、端末の計算システムやストレージの消費を防ぎ、同時にユーザ情報、推薦情報の保存を行う。(iii)では、(ii)で選択した情報を適切な型でユーザに提示する。これは、Web ページ、またはメールで行い、ユーザの状況に制約を受けず、また情報の緊急度に柔軟に対応できるシステムを実現する。

3.2 ユーザ情報の取得

ユーザの嗜好の取得方法を、明示的手法、暗黙的手法で分け、それぞれに適した嗜好情報、取得情報を述べる。明示的手法は、ユーザに対し、直接的にキーワードの入力、評価付けを強いるため、ユーザに負担を与えるものである。そこで、ユーザに対する負担を減らす、または楽しませる工夫をすることで、問題を解決し、暗黙的手法では測定困難な嗜好情報の取得が可能と考えられる。まず、単純な文章のみによる質問ではなく、ユーザにとって親しみのあるキャラクター、写真などを用いることでユーザへの精神的な負担を減らす。質問の文章体もキャラクターや写真によって変化させることで、より親しみを持たせ、ユーザが楽しめるような形式を考える。また、ユーザからの応答を、簡単な単語や選択式で済むような質問形式にすることにより、より負担の少ない手法を提案する。

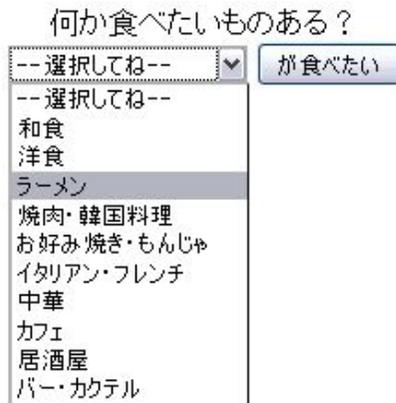


図 1 :明示的手法による嗜好取得例

暗黙的手法は、ユーザに見えないところで嗜好を取得するため、ユーザの負担にはならない。しかし、行動履歴などから嗜好を測ることは容易ではなく、また、ユーザのコンテキストや行動履歴からは測ることのできない嗜好情報などは取得が非常に困難となる。そこで、ユーザのコンテキスト、またはその履歴から判断できる嗜好を中心に取得する。具体的には、ユーザが頻繁に訪れる店舗は興味のある店舗と判断できるし、平日の日中にユーザが長く滞在する場所は職場や学校と判断できる。こうして得られる情報を蓄積していき、そのジャンルを分類することでユーザの嗜好を取得することができる。

3.3 コンテキスト

情報推薦を行う場合、ユーザのコンテキストを認識し、それに合わせた推薦を行う必要がある。ここでは、推薦に必要なコンテキストとその取得手法について挙げる。

3.3.1 位置

天気や周辺店舗情報を提示する場合など、ユーザの位置情報との結び付けは必須である。しかし、位置情報には緯度・経度、住所、建物内ならフロアなど様々な表現方法があるので、場所により異なる位置情報の取得が必要となる。例えば、住所は家や住宅の特定には便利であるが、移動中などには適していないし、逆に緯度・経度は場所がどこであっても確実に位置を特定することが可能であるがビルなどの立体構造には適さないし、フロア情報であれば立体構造にも適応しているが、建物以外の場所では使うことができない。そこで、緯度・経度、住所、フロア情報をその場所に適した方法を用いて取得する。緯度・経度は端末の GPS を用いて取得し、住所は API を用いて緯度・経度から住所を算出する。また、フロア情報は無線 LAN を用いて取得し、

同時にそれぞれの情報を結び付け、どの場所においても 3 つの情報を算出し、コンテキストとして認識する。

3.3.2 時間

ユーザにとって有用と思われる情報でも、提示する時間によってその価値は大きく変わってくる。例えば、14 時から 17 時のような中間体に周辺飲食店情報を提示することは適切とは言えず、また夕方以降にその日の天気を提示することも適切ではない。また、端末により時間を正確に調節していないものもあるため、ユーザから時間情報を送信することは適切とはいえない。そこで、時間情報はサーバで管理し、時間に依存する情報はサーバの時計を用いて配信する。また、ユーザから送信された情報は、その度にタイムスタンプを共にサーバのデータベースに保存する。

3.4 ユーザ情報の背景

同じ時間、同じ位置でもユーザによって、またはバックグラウンドによって必要な情報は変わってくる。例えば、東京在住のユーザはわざわざ周辺のホテル情報を提示される必要はなく、在住が大阪で、東京に旅行や出張にきたユーザはホテルの情報が必要になるかもしれない。ユーザの背景情報を取得する必要があるが、取得方法が明示的手法では全ての背景の取得に要する時間とユーザへの負担が非常に大きい。そこで、ユーザから取得したコンテキスト情報を計算し、背景として考慮していく。具体的には、朝、夜と滞在している場所を自宅、平日の昼間に滞在している場所を勤務地と認識し、それに合わせた推薦を行う。

3.5 提示情報の選択手法

情報推薦のためには、取得したユーザ情報に合わせて、推薦する情報を選ぶ必要がある。選択する情報によって行うフィルタリング方法は変化してくる。例えば天気や最寄り駅などの普遍的な情報に関して協調フィルタリングを適用させる必要はなく、グルメ情報など人により評価が変化する情報は協調フィルタリングの対象となりうる。

3.5.1 情報選択におけるサーバの利用

情報の選択には計算システムが必要となる。しかし、携帯端末で情報の選択を行った場合、計算システムとストレージの消費が端末に負担されてしまう。そこで、ユーザ情報をサーバに送信し、サーバで提示する情報の選択を行う。それにより、過去の履歴などをデータベースに保存しておくことが可能になり、情報の提示方法も柔軟に行うことができると考えられる。また、ユーザ情報、推薦情報は独立な形ではなくサーバ、あるいはネットワークを通じて共有されるため、蓄積された情報は協調フィルタリングソースともなりうる。

3.5.2 ユーザの行動予測を用いた選択

ユーザにとって欲しい情報は現在の位置や時間にも依存しているとは限らない。例えば、ユーザが電車に乗っていて、到着駅周辺の店舗情報が欲しい状況も考えられる。そこで、過去と現在の情報を基に未来の予定を対象とした推薦を考える。具体的

には、10分ごとに送信されたユーザの位置情報から現在移動中かを計算し、移動中であれば10分後の予定位置における情報の取得を行い推薦を行う。

3.6 推薦情報の提示手法

情報の提示は様々な手法が考えられる。しかし、その手法は提示する情報により変化し、またユーザの状況によって提示する情報体系も変化させる必要がある。情報の配信方法は、ユーザの能動性を尊重する Pull 型と、ユーザの能動性に関係なく配信する Push 型の2種類があるが、推薦における情報の配信方法はその情報によって変化させる。例えば、緊急の連絡、具体的には地震情報、気象情報などの緊急速報は Push 型で配信し、グルメ情報やイベント情報などの一般的な情報は Pull 型配信で行う。これにより、柔軟な情報の提示ができ、ユーザにとって必要なときに必要な情報を受け取ることができると考えられる。

推薦情報はいつでも、どこでも受け取ることが出来る必要がある。例えば、アプリケーションとして開発した場合、そのアプリケーションをインストールした端末でしか推薦情報を受け取ることができなくなる。また、GPS 機能を内蔵し、その中で推薦情報の計算を行うような装置を作った場合、その装置を持ちあるく前提が必要になってくる。

本研究では、Web サービスを用いた情報の提示を実現する。それにより、ブラウザを搭載した端末であれば、場所、時間を選ぶことなく推薦を受けられる。また、提示した情報の詳細を得たい場合に、リンクを提示情報に張っておくことで柔軟な情報の推薦が行える。推薦情報の提示に Web ページを用いるため、端末の電波状況により受け取られる情報に制約がかかることが考えられる。例えば、自宅でパソコンを使っている場合は、ストレスのない通信が可能だが、地下など電波が不安定な場所で携帯電話などの端末を用いて通信を行う場合、一度に大量の情報が送られてきたり、何度もページを進めなければ目的に到達できない設計では不便である。

そこで、様々な情報が一度の閲覧で認識可能な提示をする必要がある。まず、頻繁に享受したい情報を最低限の量だけ並べる。具体的には、天気、ニュース、周辺飲食店、最寄り駅などの情報である。また、時間ごとに提示する情報を変化させる。周辺飲食店の情報を提示する時間は昼と夕方以降で十分であるし、今日の天気は朝、明日の天気は夜で十分である。これらの要素をふまえ、一目で必要な情報を網羅できる提示形式を考える。

4. コンテキストウェア情報提示システムの実装

本節では、前節で挙げた提案に基づく実装について述べる。

4.1 概要

まず、クライアント側からユーザ情報が暗黙的にサーバに送信される。サーバでは受

信したユーザ情報を基に Web サービスから情報の収集を行う。この時、緊急と判断する情報があればユーザに対し Push 型の提示を行い、それ以外の情報はサーバのデータベースに保持される。ユーザからリクエストがあればサーバはデータベースから適切な情報を取り出し、提示を行う。以上の様子を図2に示す。

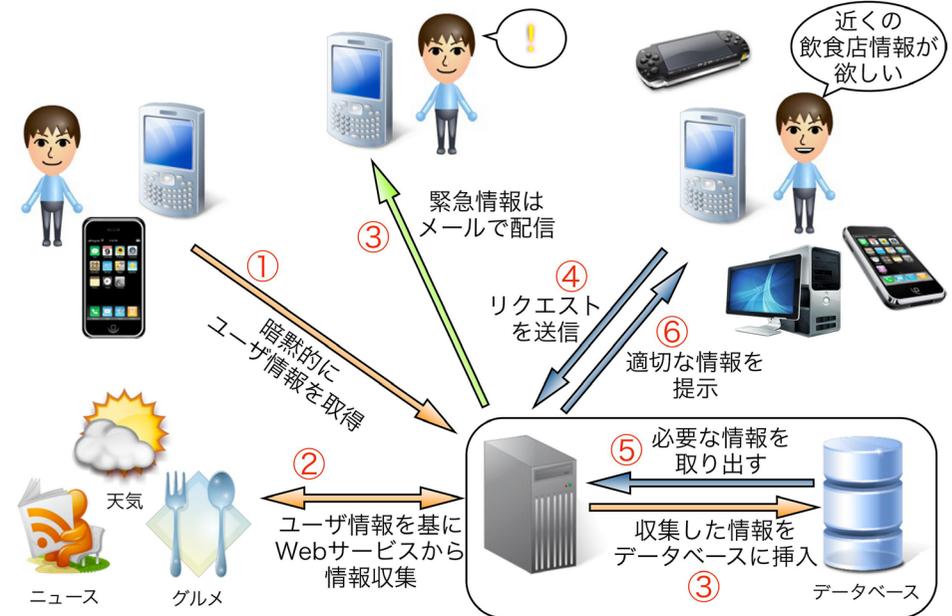


図2: コンテキストウェア情報推薦システムの概要

4.2 クライアント側の実装

ユーザの端末に対して負担を与えないため、クライアント側に必要な実装は最低限に抑える。ユーザの緯度・経度情報を定期的にサーバに送信すれば、後の計算は全てサーバが行うような仕組みを重視する。

本システムでは、クライアント側の端末として iPhone[24]、Android[25] を用いて実装した。それぞれは GPS 機能を持っており、アプリケーションの実装が可能である。エミュレータによる GPS 取得の様子を図3、4に示す。また、GPS 取得を定期的に行うことで、ユーザの位置の推移にも対応する。

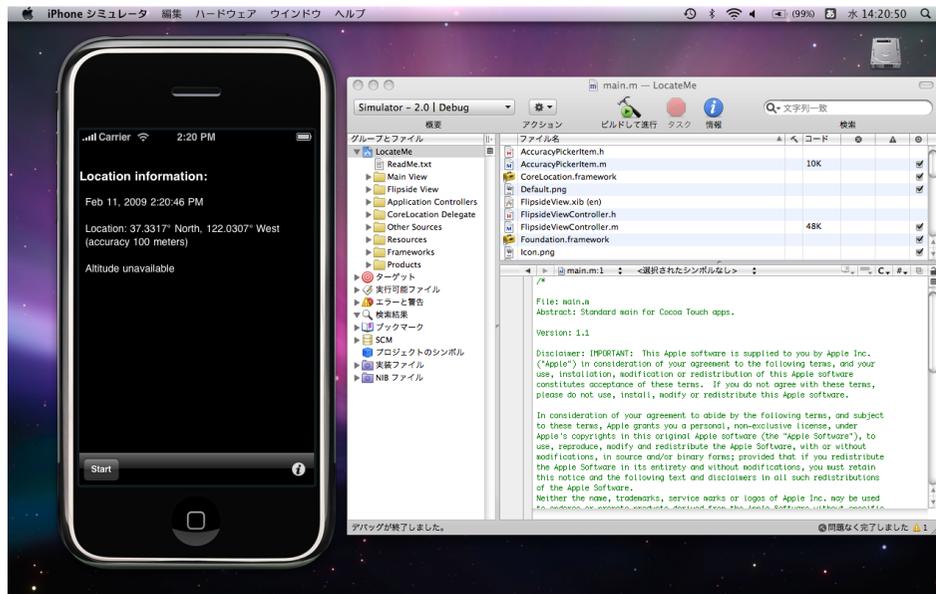


図 3 : iPhone 上の実装

4.3 サーバ側の実装

クライアント側に負担を与えないために、推薦に必要な計算は全てサーバ側で行う。具体的に行うことは、ユーザ情報の取得、推薦情報の収集、情報の提示・配信である。ここではそれぞれの実装について述べる。

4.3.1 ユーザ情報の取得

クライアント側からは GET リクエストが送信される。サーバ側は送られてきた GET リクエストのパラメータを PHP により取得する。パラメータは緯度・経度、アンケート結果である。取得した後、値を引数に Java を起動し、情報の収集を行う。

嗜好の取得は明示的手法、暗黙的手法の 2 種類を用い、明示的手法では Web ページに表示されるアンケート形式の質問に答えることによりユーザの嗜好を取得する。暗黙的手法では Web ページ上でユーザがクリックした店舗情報などから嗜好を取得する。また、ユーザにとっての背景もユーザ情報として取得する。ここでの背景とはユーザにとっての自宅や、会社、学校などの情報を指す。例えば平日の昼間に滞在している場所は学校あるいは会社などのようにユーザの時間、位置情報を基にサーバで自動的に計算する。

4.3.2 推薦情報の収集

推薦する情報の収集には WebAPI を用いることとした。収集した情報は天気、ニュー

ース、周辺飲食店、最寄り駅、住所、イベント、ホテル、ガソリンスタンドである。API はそれぞれ、WeaterHacks[26]、Yahoo!ニュース[27]、食ベログ[28]、HotPepper[29]、駅データ[30]、ReFITS[31]、eventcast[32]、楽天トラベル[33]、gogo.gs[34] を用いた。それぞれのサービスで用いられている xml 形式のファイルは Java で読み込み、サーバで実装されているデータベースの MySQL に挿入していく。また、ユーザの位置情報の推移から未来のユーザの位置を予測し、その場所に対して情報収集を行う機能も付与した。これは WebAPI から得た最寄り駅の情報の推移からユーザが移動している路線を計算し、その先の駅情報から推薦する情報の収集を行うものである。

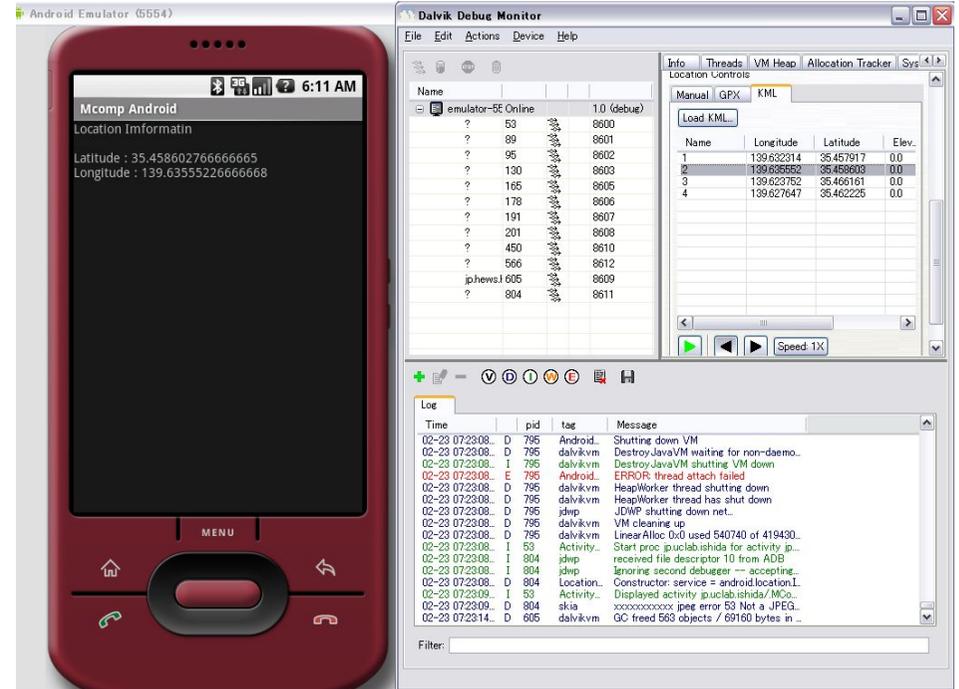


図 4 : Android 上の実装

4.3.3 情報の提示・配信

トップページに提示する情報は時間によって変化させる。まず、一日を朝：5時～11時、昼：11時～14時、中間帯：14時～17時、夕方：17時～21時、夜：21時～1時、深夜1時～5時の六つに分ける。そして、それぞれの時間帯に適した情報の提示を行う。具体的には、朝は今日の天気、ニュース、最寄り駅、住所の情報を提示し、昼にはランチ営業している周辺飲食店、今日の天気、ニュース、最寄り駅、住所の情

報を提示, 中間帯にはニュース, 今日の天気, 明日の天気, 最寄り駅, 住所の情報を提示, 夕方には周辺飲食店, 明日の天気, ニュース, 最寄り駅, 住所の情報を提示, 夜には明日の天気, ニュース, 深夜営業している周辺飲食店, 最寄り駅, 住所の情報を提示する.

また, Web ページに表示される情報にはリンクが張っており, 詳細情報を参照することを可能にする. さらに, Push 型としてメールによる情報の提示を行う. メールは JavaMail で実装しサーバから送信される. メールで提示される情報は設定可能であり, 例えば「評価の高い飲食店が近くにあったらメールで知らせる」などの設定も可能である.



図 5 : iPhone から見た本推薦システムの画面



図 6 : P C 上で見た本推薦システムの画面

5. まとめ

本研究では個人を見守るサーバを用いたコンテキストウェアな推薦システムを実現した。クライアント側では暗黙的にユーザのコンテキストを取得し、サーバに送信する。サーバはクライアント側から送られてきたコンテキストを基に情報収集を行い、データベースに保持する。コンテキストに依存した推薦として、まずユーザ情報は嗜好だけでなくコンテキストも取得し、それを基に情報収集を行う。また、情報の収集はサーバを用いて行い、クライアント側での計算機の消費をなくし、情報のデータベース保持も可能にし、ユーザにとって暗黙的なシステムを構築した。さらに、背景、行動予測などコンテキストの活用によって得られる情報を用いることで、ユーザに特化したシステムを実現した。

推薦情報に応じた柔軟な提示として、端末を選ばない提示として Web ページを用いた情報の提示を行った。また、緊急情報など早急にユーザに伝えるべき情報はメール配信を用いることで情報の緊急度に対応した提示を構築した。さらに、1つのページに多様な情報を載せることで、画面が小さく、電波強度の弱い携帯端末でもストレスが少ない提示を実現した。

これらの処理はすべてサーバ上で行われ、クライアント側では情報提示を行うのみであり、かつユーザからのリクエストより前にサーバ側でデータ収集が行われているため、14種類の情報を収集するのに必要な5秒程度の時間が不要となり、0.5秒で表示が可能になる。10分毎に最適なデータを取得した場合、1日毎に25MByte、1年で9GByteのデータが保持されることになるが、データの保持期間の調整などによって規模への対応は可能である。

参考文献

- 1) google, "Gmail", <http://mail.google.com/mail>
- 2) ヤフー株式会社, "Yahoo!Briefcase", <http://briefcase.yahoo.co.jp/>
- 3) amazon, "Amazon EC2", <http://aws.amazon.com/ec2/>
- 4) Steve Lawrence, C. Lee Giles: "Accessibility of information on the Web", Nature 400, pp. 107-109, (1999)
- 5) 情報大航海プロジェクト, "マイニング技術の検証用プロファイルデータ",
6) <http://www.igvpj.jp/contents/pdf/A-3.pdf>
- 7) 河口信夫: "Locky.jp: 無線 LAN を用いた位置推定とその応用", 電子情報通信学会, ITS 研究会, Vol.107, No.ITS-161, ITS2007-16 (2007)
- 8) 根岸佑也, 河口信夫: "ユビキタスコンピューティングにおけるコンテキストセンシングとデータ処理", 人工知能学会誌, Vol. 23, No. 6, pp. 597-603 (2008)
- 9) 土方嘉徳: "情報推薦・情報フィルタリングのためのユーザプロファイリング技術", 人工知能学会誌, Vol. 19, No. 3, pp. 365-372 (2004)
- 10) YAN, T. W.: "SIFT-A Tool for Wide-Area Information Dissemination", Proc. Of 1995 USENIX Technical Conference, 1995, pp. 177-186, (1995)
- 11) P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl: "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", Proc. of ACM 1994, Conference on Computer Supported Cooperative Work (CSCW '94), pp. 175-186(1994)
- 12) T. Joachims, D. Freitag and T. Mitchell: "WebWatcher: A Tour Guide for the World Wide Web", Proc. of IJCAI '97 (1997)
- 13) L. Chen and K. Sycara: "WebMate: A Personal Agent for Browsing and Searching", Proc. of the 2nd International Conference on Autonomous Agent (Agent '98), pp.132-139 (1998)
- 14) Morita, M.: "Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval", Proc. 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 272-281, (1994)
- 15) 岩濱敦宏, 土方嘉徳, 西田正吾: "決定木を用いた音楽情報フィルタリングシステムとその有効性の検証", 電子情報通信学会論文誌D, Vol. J88-D1, No. 3, pp. 642-656(2005)
- 16) amazon, "Amazon.com", <http://www.amazon.com/>
- 17) GroupLens Research, "movielens", <http://movielens.umn.edu/login>
- 18) Chen, G. and Kotz, D.: "A Survey of Context-Aware Mobile Computing Research", Paper TR2000-381, Department of Computer Science, Dartmouth College (2000)
- 19) Asthana, A., Cravatts, M., Krzyzanowski, P.: "An Indoor Wireless System for Personalized Shopping Assistance", Proc. IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, pp. 69-74, IEEE Computer Society Press (1994)
- 20) G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton.: "Cyberguide: A mobile context-aware tour guide", Wireless Networks, 3(5), pp 421-433, October (1997)
- 21) 矢野絵美, 飯島貴広, 篠原勲, 加藤俊一: "ユビキタスな情報タグのパーソナルライブメモ化と行動支援への応用", DBWeb2003, pp.33-38 (2003)
- 22) 奥 健太, 中島伸介, 宮崎純, 植村俊亮: "状況依存型ユーザ嗜好モデリングに基づく Context-Aware 情報推薦システム", 情報処理学会論文誌, Vol. 48, No. 11, pp.162-176(2007)
- 23) 株式会社 NTT ドコモ, "i コンシェル"
- 24) 株式会社 CLON Lab, "clon", <http://www.clon.co.jp/>
- 25) Apple Inc., "iPhone", <http://www.apple.com/iphone/>
- 26) google, "Android", <http://code.google.com/intl/android/android/>
- 27) 株式会社ライブドア, "Weather Hacks", <http://weather.livedoor.com/weather hacks/>
- 28) ヤフー株式会社, "Yahoo!ニュース Web API", <http://developer.yahoo.co.jp/webapi/news/>
- 29) 株式会社カカクコム, "食ベログ API", <http://apiblog.kakaku.com/>
- 30) 株式会社リクルート, "ホットペッパーWeb サービス", <http://api.hotpepper.jp/>
- 31) 株式会社コードプラス, "駅データ.jp", <http://www.ekidata.jp/tools/api station c.html>
- 32) ReFITS Lab, "地図関連情報検索計算サービス", <http://refits.cgk.affrc.go.jp/tsrv/jp/pinfo.html>
- 33) eventcast inc., "eventcast", <http://clip.eventcast.jp/webservices/api.html>
- 34) 楽天株式会社, "楽天トラベル施設情報 API", <http://webservice.rakuten.co.jp/api/>
- 35) 株式会社ゴーゴーラボ, "gogo.gs", <http://api.gogo.gs/>