

## 移動系列を利用した行列分解による訪問地予測

## Visit place prediction by the line resolution using move series

納竜也\*<sup>1</sup> 佐久間淳\*<sup>2\*3</sup>  
Tatsuya Osame Jun Sakuma

\*<sup>1</sup>筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻  
Dept. of Computer Science, Graduate school of SIE, Univ. of Tsukuba

\*<sup>2</sup>筑波大学 システム情報系  
Faculty of Engineering, Information and Systems, Univ. of Tsukuba

\*<sup>3</sup>科学技術振興機構 さきがけ  
Japan Science and Technology Agency

In recent years, is becoming to be able to accumulate large amounts of personal behavioral history, such as GPS. In location-based service, the prediction of the location which user will visit is important. In this paper, we propose the prediction method based on matrix factorization. We consider the attributes of user is useful for location prediction. In this paper, we propose a method incorporating the attribute to matrix factorization. I have conducted experiments using real data. As a result, we was not able to show the efficient of using the attribute.

## 1. はじめに

近年 GPS などの位置情報を取得するセンサ機器などの発達により、大量の個人の行動履歴を蓄積できるようになりつつある。また、多くの人の移動履歴を集めたデータセットも公開されている。たとえば、「人の流れプロジェクト」で提供されているパーソントリップデータがある。このデータは大都市圏の1日分の移動履歴を集めたデータであり、本研究では東京都市圏のパーソントリップデータを用いて実験を行う。

位置情報を用いたサービスにおいて、ユーザの次の訪問地を予測するタスクは、サービスの基盤として重要である。サービスの例としては、次に訪問する地点の周囲のショップ情報を提示したり、その地点へ行くための交通機関の時刻表を提示したりするものが考えられる。本研究では、機械学習の手法に基づき予測・推薦する方法について検討する。

従来の訪問地予測の手法としては、ユーザの重要な地点(自宅や勤務先など)を特定し、その地点間の遷移関係(どの地点からどの地点へ移動したか)を利用して予測する手法や移動経路(どういう経路を用いたか)の履歴を用いて予測する手法などがある。遷移関係を利用する手法では、遷移関係が存在しない移動を予測ができないという問題がある。移動経路の履歴を利用する手法では、高精度な予測を行うことができるが、多くのデータを必要とする。

これらのことを考慮し、本研究では行列分解による訪問地の予測・推薦を行う。この手法では、移動履歴ではなく、複数のユーザのどの地点を訪問したかという情報(訪問履歴)を利用して予測・推薦する。行列分解による手法では、ユーザや地点の特徴を得ることにより、単なる予測ではなくユーザに適した地点を推薦することも可能である。

一般的な行列分解に基づく手法は、映画や商品などのユーザ-アイテム型の協調フィルタリングシステム分野で成功している。しかし、一般的な行列分解の手法を訪問地予測にそのまま適用してもうまく予測することができない。その理由として、以下の3つが挙げられる。1つ目の理由は、ユーザ-アイテム型のものとは違い、「学生が訪れやすい地点や会社員が訪れやすい地点」といった事前知識が存在するが、一般的な行列

分解による推薦ではこのような情報を活用していない。この事前知識は、訪問地の予測に有益な情報であるので、行列分解に事前知識を組み込むことが望ましい。2つ目の理由は、位置情報を行列分解で扱うのが困難なことである。一般的に位置情報は緯度と経度で表現されているため、行列で表現するために何らかの離散化が必要である。3つ目の理由は、典型的なユーザ-アイテム行列で表現される購買履歴等とは異なり、訪問地予測ではデータの時系列に重要な意味がある点である。

1つ目の問題の解決策として、行列分解における学習において、訪問地予測に固有の事前知識を目的関数の正則化項として導入する。2つ目の解決策として、位置情報をグリッド分割で表現する。この方法は、位置に対して何らかの意味(たとえば、観光施設など)を付与することはできないが、計算上の取り扱いが容易である。位置に対して意味を付与する POI(point of interest)の導入は、行列分解と必ずしも相性が悪いわけではなく、今後取り組む予定であるが今回は扱わない。また、3つ目の問題も重要な論点ではあるが、行列分解において任意長の時系列情報を直接扱うことは困難である。そのため、時系列を長さ2の時系列の集合として扱う。具体的には、ある地点から違う地点への移動といった、移動のパスを扱うことで、系列情報全体ではなく部分的な情報を行列分解に導入する。

## 2. 訪問地予測

訪問地予測とは、過去の移動履歴などから次に訪問する地点を予測する手法である。従来の手法として、

- ユーザの滞在地の遷移関係を用いて予測 [1]
- ユーザの移動経路を特定して予測 [3]

がある。これらの手法は、高い精度の予測が可能であるが、多くのデータが必要という問題がある。また、ユーザが滞在地や移動した経路を利用するため、ユーザや地点の特徴を捉えるのが困難である。そのため、過去に訪問したことはないが、ユーザにとって訪問する価値がある地点を提示することができないという問題がある。

これらの問題を解決するために、本研究では、行列分解を用いて訪問地予測を行う。行列分解を用いた場合、予測精度の面

では従来の手法に劣るが、ユーザや地点の特徴を考慮した地点の提示を行うことができる。また、他のユーザのデータと合わせて予測を行うので、従来の手法よりも少ないデータ量で予測が行えるという利点もある。

### 3. 位置情報の表現

緯度と経度で表現された位置情報は連続値である。この情報を行列分解で扱うためには適切な離散値表現が必要である。位置情報を離散値で表現する方法としてグリッド分割を用いる。グリッド分割は、緯度と経度を等間隔なグリッドで分割し、1つの分割を1つの地点として表現する方法である。

移動履歴の行列表現として、訪問地点を要素として扱う方法と、A地点からB地点への移動(パス)を要素として扱う方法を考える。位置情報の行列への変換方法を以下及び図1に示す。

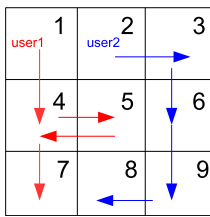


図 1: グリッド分割の例

1. 緯度と経度を等間隔なグリッドで分割し、1つの分割を1つの地点とする。
2. 各ユーザの移動履歴の位置情報を、グリッド分割表現に変換する。図1の場合、ユーザ1が(1,4,5,7)。
3. 各ユーザの移動履歴からパスを抽出する。図1の場合、ユーザ1は1→4→5→4→7と移動しているの(1→4),(4→5),(5→4),(4→7)がパスとして抽出される。
4. すべてのユーザから抽出されたパスから重複したものを除き、順にIDを割り当てる。
5. 行がユーザ、列がパスに対応する  $n$ (ユーザ数)  $\times$   $m$ (パス数)の行列  $\mathbf{R}$  を作成。行列の要素  $r_{ij}$  は、ユーザ  $i$  の移動履歴にそのパスが含まれていれば1、含まれていない場合は0とする。

パスを利用した表現では、始点と終点が存在する。そのため、移動履歴に含まれる系列情報を部分的に利用しているといえる。

### 4. 行列分解を利用した訪問地予測

#### 4.1 一般的な行列分解

行列分解 [4] では、 $n \times m$  の行列  $\mathbf{R}$  を  $d \times n$  の行列  $\mathbf{U}$  と  $d \times m$  の行列  $\mathbf{V}$  の2つの行列の積で近似する ( $\hat{\mathbf{R}} \simeq \mathbf{U}^T \mathbf{V}$ )。ここで、 $d$  は  $n$  や  $m$  より十分小さいとする。 $\mathbf{R}$  を行にユーザ、列にパスを取った行列とすると、 $\mathbf{U}$  と  $\mathbf{V}$  の各列はそれぞれユーザやパスに対応しており、それぞれの行列の列ベクトルがユーザやパスの特徴ベクトルに対応する。近似した行列の各要素は、 $\mathbf{U}$  と  $\mathbf{V}$  それぞれの列ベクトル  $\mathbf{u}_i$  と  $\mathbf{v}_j$  の積で求ま

表 1: 属性情報のインデックス例

インデックス	属性		
	性別	年齢	職業
1	男性	10代	学生
2	男性	10代	会社員
⋮	⋮	⋮	⋮
$\ell$	女性	60以上	会社員

る ( $\hat{r}_{ij} = \mathbf{u}_i^T \mathbf{v}_j$ )。分解した2つの行列の各要素は、式(1)のような目的関数を定義し、それを最小化するように、最急降下法を用いて求める。

$$L(\mathbf{U}, \mathbf{V}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m y_{ij} (r_{ij} - \hat{r}_{ij})^2 + \frac{\alpha}{2} \left( \sum_{i=1}^n \mathbf{u}_i^T \mathbf{u}_i + \sum_{j=1}^m \mathbf{v}_j^T \mathbf{v}_j \right) \quad (1)$$

ここで  $\alpha$  は正則化パラメータである。近似した行列  $\hat{\mathbf{R}}$  の各要素を以降では推定値として扱う。

#### シグモイド関数の導入

元の行列  $\mathbf{R}$  の要素は0または1の二値であるが、行列分解による推定値の範囲は  $(-\infty, \infty)$  である。この推定値の範囲を  $(0, 1)$  に制約するためにシグモイド関数を導入する。シグモイド関数を用いた推定値は  $\hat{r}_{ij} = \frac{1}{1 + \exp(-\mathbf{u}_i^T \mathbf{v}_j)}$  となる。これを確率的行列分解と呼ぶ [5]。

#### 4.2 ユーザの属性情報を利用した手法

ユーザの年齢や職業などの属性情報は、訪問する地点に影響を与えると考えられる。例えば、地域の特性によって、学生が訪問しやすい地点や会社員が訪問しやすい地点がある。この属性情報をパスの特徴として取り入れる。今回は、ユーザの属性情報として性別、年齢、職業を利用する。

まず、ユーザの各属性情報の組み合わせに対してインデックスを割り当てる。表1に例を示す。ここで、 $\ell$  は各属性情報の組み合わせ数である。

次に、 $\ell \times m$  のユーザ属性によるパスの利用頻度のクロス集計行列  $\mathbf{C}$  を作成する。 $c_{kj}$  は  $k$  番目の組み合わせと同じ属性を持つユーザ集合 ( $U_k$ ) の内、パス  $j$  を利用したユーザの総数を表す ( $c_{kj} = \sum_{i \in U_k} r_{ij}$ )。

このクロス集計行列  $\mathbf{C}$  を利用して、 $\ell \times m$  のパスの属性行列  $\mathbf{A}$  を作成する。 $\mathbf{A}$  の各列ベクトルは各々のパスの属性ベクトルであり、各要素  $a_{kj}$  は  $a_{kj} = c_{kj} / \max_f c_{fj}$  で定める。

作成したパスの属性行列より、パス間の類似度行列  $\mathbf{S}$  を作成する。類似度は2つのパスの属性ベクトルによる相関係数を利用する。

$$s_{ij} = \frac{\sum_{k=1}^{\ell} (a_{ki} - \bar{a}_i)(a_{kj} - \bar{a}_j)}{\sqrt{\sum_{k=1}^{\ell} (a_{ki} - \bar{a}_i)^2} \sqrt{\sum_{k=1}^{\ell} (a_{kj} - \bar{a}_j)^2}}$$

$$\bar{a}_i = \frac{1}{\ell} \sum_{k=1}^{\ell} a_{ki}$$

この類似度行列  $\mathbf{S}$  を用いて、次の二つの方法で行列分解を導入する。

**Algorithm 1** 行列分解のパラメータの学習**Input:**  $R, \eta, \alpha, \beta$   
**Output:**  $U^{(T)}, V^{(T)}$ 

```

1:  $t \leftarrow 0$ 
2: Initialize  $U^{(0)}$  and  $V^{(0)}$  by random values
3: while satisfy the convergence condition do
4:    $t \leftarrow t + 1$ 
5:   for  $i = 1$  to  $n$  do
6:      $u_i^{(t)} \leftarrow u_i^{(t-1)} - \eta \frac{\partial L'}{\partial u_i}$ 
7:   end for
8:   for  $j = 1$  to  $m$  do
9:      $v_j^{(t)} \leftarrow v_j^{(t-1)} - \eta \frac{\partial L'}{\partial v_j}$ 
10:  end for
11: end while

```

**4.2.1** 重み行列  $W$  の利用

$S$  を用いて、各ユーザの各パスに対する重み行列  $W$  を式 (2) で作成する。

$$w_{ij} = \frac{\sum_k r_{ik} s_{jk}}{\sum_k r_{ik}} \quad (2)$$

重み行列  $W$  の要素  $w_{ij}$  は、ユーザ  $i$  が利用したパスとパス  $j$  の類似度の平均値を表す。作成した重み行列  $W$  を式 (3) により行列分解に導入する。

$$L'(U, V) = L(U, V) + \frac{\beta}{2} \sum_{i=1}^n \sum_{j=1}^m (w_{ij} - \hat{r}_{ij})^2 \quad (3)$$

ここで、 $\beta$  は正則化パラメータである。式 (3) の右辺第二項は、重みが大きいほど推定値を大きくする働きがある。つまり、ユーザが利用したパスとの平均類似度が高いパス程推定値を大きくする働きがある。

**4.2.2** Relation Regularized Matrix Factorization

Relation Regularized Matrix Factorization (RRMF)[2] は、ウェブページや研究論文の分類タスクにおいて、ウェブページなどのコンテンツ情報に加えてウェブページ間のリンク情報を事前知識として利用する手法である。本研究では、リンク情報の代わりにパス間の類似度行列  $S$  を利用する。パス間の類似度を考慮した正則化項を式 (4) に示す。

$$L''(U, V) = L(U, V) + \frac{\beta}{2} \sum_{j=1}^m \sum_{k=1}^m s_{jk} \|v_j - v_k\|^2 \quad (4)$$

ここで、 $\beta$  は正則化パラメータである。式 (4) の右辺第二項は、パス間の類似度が大きいほど特徴ベクトルが類似するようペナルティを与える一種の正則化項である。この正則化項は、結果として属性情報が似ているパスは学習結果が似ているようにする働きがある。

**4.3** パラメータの学習

パラメータの学習には、最急降下法を用いて  $U$  と  $V$  を交互に学習する。それぞれの学習は 1 ステップにつき特定の 1 列のみ更新し、その他の列は固定して更新を行う。

Algorithm 1 に、目的関数に式 (3) を用いた場合の最急降下法による行列分解のパラメータ学習アルゴリズムを示す。アルゴリズム中の  $\eta$  はパラメータ更新のステップサイズパラメータである。式 (1) や式 (6) も同様のアルゴリズムで学習を行う。

$U$  と  $V$  を 1 回更新するのに必要な計算量は、式 (3) が  $O(nmd)$ 、式 (4) が  $O((n+m)md)$  である。 $n, m$  は大きな値であるため、このまま学習を行った場合、時間がかかってしまう。そこで  $s_{jk}$  に対して閾値 ( $\delta$ ) を利用することで、計算時間を減らす。 $K_j$  を  $K_j = \{k | \delta \leq |s_{jk}|\}$  とすると、式 (2) 及び式 (4) を以下のように変更する。

$$w_{ij} = \frac{\sum_{k \in K_j} r_{ik} s_{jk}}{\sum_{k \in K_j} r_{ik}} \quad (5)$$

$$L''(U, V) = L(U, V) + \frac{\beta}{2} \sum_{j=1}^m \sum_{k \in K_j} s_{jk} \|v_j - v_k\|^2 \quad (6)$$

しきい値  $\delta$  を調整することで、 $u_i$  の更新に必要な計算時間と近似精度のバランスを調整することができる。

**5. 実験****5.1** 実験方法**5.1.1** 実験データ

本研究では、人の流れプロジェクト<sup>\*1</sup>で公開されているパーソントリップデータを利用する。このデータは、大都市圏におけるある一日の個人の位置情報を 1 分間隔で記録したデータである。データの収集には調査票を用いてユーザがいつ、どこに行ったかの情報を集め、そのデータを元に緯度と経度の位置情報が作成されている。位置情報の他には、ユーザの属性情報や移動に用いた交通手段などが含まれている。実験では平成 10 年の東京都市圏で収集されたデータを利用し、銀座駅、新宿駅、東京駅の 3 つの場所の周囲 4km 四方の領域を対象とした。

また、位置情報は緯度と経度をそれぞれ 10 分割したグリッドで表現して扱った。実験では、すべてのユーザを実験の対象とするのではなく、5ヶ所以上違う地点を訪問したユーザのみを対象とし、対象のユーザを 20 分割し、各々をデータセットとした。

テストデータと訓練データは、各ユーザの移動履歴の内、最後に訪問した場所をテストデータ、残りを訓練データとした。また、パス表現を利用して実験を行う場合には、テストデータとして選択された地点の 1 つ前の地点を現在地として、その次の訪問地を予測するものとした。

**5.1.2** 実験方針

本実験では、

- **実験 I** 地点表現とパス表現の比較
- **実験 II** 属性情報の効果

について実験を行った。実験 I では、行列表現の際にユーザが訪問した地点を利用する場合とユーザが利用したパスを利用する場合について比較を行った。また、

**ランダム** ランダムに予測

**訪問頻度** 各地点の訪問頻度の高い順に予測

**遷移確率** 現在地からの遷移確率の高い順に予測

**類似度** 現在地を始点とするパスの中で、重み行列  $W$  の値の高い順に予測

の 4 つの手法とも比較した。

実験 II では、式 (2) を利用した式 (3) と式 (6) で定義した、ユーザ属性情報を学習に取り入れたときの精度を求め、その効果を評価した。

\*1 <http://pflow.csis.u-tokyo.ac.jp/index-j.html>

表 2: 地点表現とパス表現の捕捉率の比較

	銀座駅	新宿駅	東京駅
地点	0.072	0.107	0.130
パス	<b>0.756</b>	0.860	<b>0.744</b>
ランダム	0.055	0.055	0.056
訪問頻度	0.045	0.067	0.041
遷移確率	0.740	0.852	0.733
類似度	0.749	<b>0.882</b>	0.722

表 3: ユーザ情報の効果

	銀座駅	新宿駅	東京駅
属性を利用しない(式(1))	0.756	0.860	0.744
属性を利用 重み行列(式(3))	<b>0.774</b>	0.862	<b>0.745</b>
RRMF(式(6))	0.756	0.871	0.744
類似度	0.749	<b>0.882</b>	0.722

### 5.1.3 評価方法

実験の評価は以下の方法で行った。

1. 訓練データを用いて、各ユーザに対してすべての未訪問地に対する行列分解によるスコアを求める。
2. スコアの上位 5 個の未訪問地を推薦結果とする。
3. テストデータと推薦データを比較して、次の式で定義する捕捉率を求める。

$$\hat{c} = \frac{1}{n} \sum_{i=1}^n c_i \quad \text{where } c_i = \begin{cases} 1 & \ell_i^{\text{test}} \in \mathcal{H}_i^{\text{rec}} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

ここで、 $\ell_i^{\text{test}}$  と  $\mathcal{H}_i^{\text{rec}}$  はそれぞれテストデータに含まれるユーザ  $i$  の訪問地点とユーザ  $i$  への推薦地点集合を表す。

行列表現にパスを利用する場合は、地点と同様にまずスコアを降順に並び替える。そして、始点がユーザの現在地と一致するパスの中から上位 5 個を選択し、そのパスの終点を推薦結果として提示した。

## 5.2 実験結果

### 5.2.1 実験 I: 地点表現とパス表現の比較

行列表現の方法として、地点を利用する場合とパスを利用する場合の捕捉率を比較した。表 2 に地点を利用する場合とパスを利用する場合の比較結果を示す。学習には式(1)を用いた。また、比較手法として挙げた 4 つの手法の結果も示す。

表 2 の結果を見ると、パスを利用することで捕捉率を大幅に改善することができている。パスを利用した場合、推薦の出力が長さ 2 の系列情報であり、地点間の遷移の予測に有利であると考えられる。また、予測の際に現在地情報を利用できることも、性能向上の一因であると考えられる。

しかし、同じ系列情報を利用する遷移確率や類似度の結果と比較すると、捕捉率の差があまりない。特に新宿駅の場合は、類似度の手法が最も良い結果を得ることができた。これは、今回実験で用いたデータセットには 1 分間隔の移動履歴が含まれており、十分なデータがあったためだと考えられる。またパスと遷移確率、類似度の手法は、現在地からテストデータの地点へのパスが訓練データに含まれていなければ推薦結果として提示することができないという問題がある。この点は今後検討する必要がある。

### 5.2.2 実験 II: ユーザ属性情報の効果

ユーザの属性情報を学習に導入した場合の効果調べる。表 3 に目的関数として、式(1)、式(5)を利用した式(3)、式(6)、類似度の捕捉率の比較結果を示す。

表 3 より、属性情報を利用した式(3)や類似度の手法が各エリアで最も高い捕捉率を得ることができているので、属性情報の導入に効果はあると考えられる。しかし、導入しない式(1)の実験結果と大きな差があるとは言えないため、効率的なユー

ザの属性情報の導入方法を検討する必要がある。性能の大きな向上に貢献しているとは言えない。また新宿駅の結果では、類似度のみの結果より、行列分解に導入した時の方が捕捉率が低くなっている。この理由についても検討の余地がある。

## 6. まとめと今後の課題

行列分解を利用した訪問地予測についての研究を行った。移動履歴の行列表現の方法として、ユーザ-地点の行列を利用する方法とユーザ-パスの行列を利用する方法を検討した。その結果、パスを利用した場合のほうが予測の精度が高いという結果が得られた。

また、ユーザの属性情報を行列分解に導入する方法も検討した。実験の結果からは、数%程度の精度改善の効果があった。年齢や職業などの属性情報は訪問地予測・推薦において有用な情報だと考えられるので、属性情報の有効な導入の方法について検討する予定である。

その他の課題として、系列情報の行列分解における取り扱いがある。行列分解をベイズ的に解釈した場合、目的関数の正則化項が事前分布に対応する。この事前分布に、系列データを扱う確率モデルを利用することで行列分解に系列情報を取り入れることを検討している。

## 謝辞

本研究は最先端研究開発プログラム (FIRST) 「超巨大データベース時代に向けた最高速データベースエンジンの開発と当該エンジンを核とする戦略的社会サービスの実証・評価」の助成を受けたものである。研究実施に当たり、実験データをご提供頂いた東京大学空間情報科学研究センター「人の流れプロジェクト」に感謝の意を表します。

## 参考文献

- [1] D. Ashbrook. Learning significant locations and predicting user movement with gps. In *Proceedings of the 6th IEEE International Symposium on Wearable Computers*, pp. 101–108, Washington, DC, USA, 2002. IEEE Computer Society.
- [2] Wu-Jun Li and Dit-Yan Yeung. Relation regularized matrix factorization. In *Proceedings of the 21st international joint conference on Artificial intelligence(IJCAI09)*, pp. 1126–1131, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.
- [3] Lin Liao, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. In *Proceedings of the 19th national conference on Artificial intelligence(AAAI04)*, pp. 348–353. AAAI Press, 2004.
- [4] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. *Proceedings of KDD Cup and Workshop*, 2007.
- [5] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pp. 1257–1264. MIT Press, Cambridge, MA, 2008.