ジオタグベースモートン順序を用いた 連想型クロスドメイン推薦システム

井田　彰† 奥　健太†

† 龍谷大学理工学部　〒 520-2194 滋賀県大津市瀬田大江町横谷 1-5
E-mail: †t140432@mail.ryukoku.ac.jp, ††okukenta@rins.ryukoku.ac.jp

あらまし　クロスドメイン推薦システムは，映画や音楽，本など異種のドメインにまたがってアイテムを推薦するシステムである．クロスドメイン推薦を実現するためには，異なるドメインにおけるアイテム間で共通する特徴を抽出することが課題となる．本研究では，小説の舞台になった地域や映画のロケ地など，アイテムには地域との何らかの関連性があるという仮定をおいている．この仮定を基に，共通する地域に関連するアイテムを推薦することでクロスドメイン推薦を行う．具体的には，クラウドソーシングにより収集したアイテム-位置情報の関係データから，空間データの管理に用いられるモートン順序をアイテムに付与する．付与されたモートン順序を用いて空間的に近傍のアイテムを関連付けて推薦する．

キーワード　クロスドメイン推薦システム，ジオタグ，位置情報，モートン順序

1. はじめに

2016 年の訪日外国人旅行者数が 2,403 万人に達する (注1) など観光を楽しむ人々は増大してきている．実際に観光地を訪れた人々は，現地での景色を眺めたり，名物料理を食したりするなど，観光の楽しみ方は人それぞれである．近年，映画や小説の舞台となった場所へ訪れる観光客が多くなり，日本人だけでなく，海外からもその場所を一目見ようとやって来る人も珍しくない．作中の人物が居た地に訪れて何かを感じたり，その観光地に関連する他のアイテムを探すことも観光を楽しむ方法の一つである．

このような背景の下，我々の先行研究 [1] では，観光スポットベースのクロスドメイン推薦システムを提案してきた．このシステムでは，ユーザが興味のあるアイテムを入力すると，そのアイテムに関連した書籍や映画，楽曲など，ドメインをまたがったアイテムをユーザに提示する．このようなシステムを実現するためには，あらかじめ位置情報とアイテムとを関連付けておく必要がある．先行研究 [1] では，クラウドソーシングによりアイテムへの位置情報付与を行った．アイテムに位置情報を付与することをジオタギング，位置情報が付与されたアイテムをジオタグアイテムと定義している．これらジオタグアイテムを参照することで，位置情報とアイテムを関連付けている．本研究では，空間データを効率よく管理できる手法としてモートン順序に着目する．モートン順序は空間を階層ごとに 4 分割していき，最下層におけるセルに対し番号—これをモートン順序とよぶ—を付与しておく．一度，このモートン順序が得られれば，そのセルの親空間での位置は簡単な計算で割り出すことができる．

本研究では，このモートン順序を用いてジオタグアイテムを管理する．モートン順序を用いることにより，階層ごとに関連の強いアイテムを効率よく探索することができる．このような特性を利用し，我々は下記の二つの推薦機能をもつクロスドメイン推薦システムを提案する：

(a)　位置情報-アイテム推薦手法 : 入力された位置情報に関連するジオタグアイテムを探索する．

(b)　アイテム-アイテム推薦手法 : 入力されたジオタグアイテムに関連する他のジオタグアイテムを探索する．

さらに，これら (a) および (b) を組み合わせた連想型クロスドメイン推薦について考察する．

(注1) : https://www.jnto.go.jp/jpn/statistics/visitor_trends/index.html

2. 関連研究

2.1　クロスドメイン推薦システム

従来の推薦システム [2] は単一のドメインにおいて嗜好データを獲得し，それに基づき当該ドメインにおける推薦を行うという方法が主流であった．例えば，映画推薦システムでは，映画の視聴履歴や評価履歴に基づき対象ユーザの嗜好に関する学習モデルを獲得する．その学習モデルを用いてやはり映画ドメインにおける嗜好を予測する．しかし，単一ドメインにおいて獲得された学習モデルは他のドメインにおいては利用できない．仮に映画ドメインにおいて十分な嗜好データが蓄積されていたとしても，それをそのまま書籍ドメインにおける嗜好の予測に用いることはできない．

そこで，近年クロスドメイン推薦という考え方が注目されてきた [3]．クロスドメイン推薦システムは，複数のドメインにまたがった推薦を実行するシステムである．例えば，映画ドメインで蓄積された嗜好データを基に，書籍ドメインにおける嗜好を予測するというものである．これは異なるドメインにおいてもドメイン間の嗜好には何らかの関連性があるという考えに基づく．

クロスドメイン推薦は，一つのドメイン（ソースドメインと

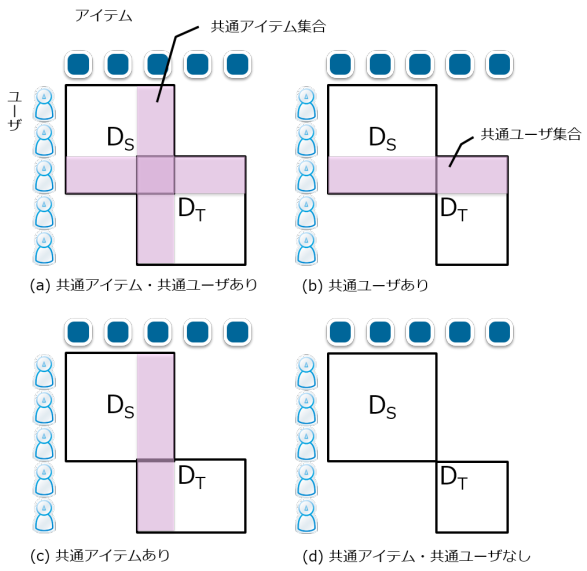

図 1　クロスドメイン推薦問題の類型．

図 2　システムインタフェース．インタフェースは，入力ビューと二つのマップビューから構成される．左側のマップビューが詳細マップビュー，右側のマップビューが広域マップビューである．

よぶ）において蓄積された評価値データを用いて，他のドメイン（ターゲットドメインとよぶ）における評価値を予測することが目的となる [4]．Cremonesi ら [5] は，クロスドメイン推薦問題を図 1 のように四つの類型に分類している．図 1 は，ユーザ-アイテムの評価値行列を表し，行がユーザを列がアイテムを表す．D_S がソースドメインの評価値行列，D_T がターゲットドメインの評価値行列を表す．クロスドメイン推薦では，ソースドメイン D_S の評価値行列を基に，ターゲットドメイン D_T の評価値を予測することが課題となる．

ソースドメイン D_S からターゲットドメイン D_T の評価値を予測するための一つの手がかりとして，共通ユーザ集合および共通アイテム集合が用いられる．図 1(b) は，D_S, D_T において共通ユーザ集合が存在することを表す．つまり，同一のユーザが異なるドメインにおいても評価値を与えていることを意味する．図 1(c) は，D_S, D_T において共通アイテム集合が存在することを表す．つまり，同一の（あるいは同等とみなせる）アイテムが異なるドメインにおいて存在することを意味する．図 1(a) は，共通ユーザ集合も共通アイテム集合も存在することを表す．このように共通ユーザ集合や共通アイテム集合が存在する場合，D_S, D_T において評価値行列が重複するため，その重複部分から D_S と D_T の関係性を推定することが可能である．例えば，Sahebi ら [4] は，この重複部分を基に正準相関分析を行い，ソースドメインの評価値をターゲットドメインの評価値への換算式を獲得している．

このように評価値行列の重複部分が存在するような場合には，ソースドメインからターゲットドメインへの嗜好の予測は可能となるが，図 1(d) のように重複部分が存在しない場合にはこのようなアプローチは適用できない．したがって，二つのドメインにおける共通属性を別途抽出するなどのアプローチが必要である．

本研究では，複数のドメインにおける共通属性として位置情報に着目している．クラウドソーシングによりアイテムに位置情報を付与しておく．位置情報を基に観光スポットと各ドメインに属するアイテムとを対応付けておくことで，観光スポットを介したクロスドメイン推薦が可能であると考える．

3. システム概要

本章では，提案システムである連想型クロスドメイン推薦システムについて説明する．まず，システムのインタフェースについて説明し，システム構成について説明する．

3.1 インタフェース

図 2 はシステムのインタフェースである．インタフェースは，入力ビューと二つのマップビューから構成される．左側のマップビューが詳細マップビュー，右側のマップビューが広域マップビューである．

ユーザは入力ビューから検索キーワードおよびアイテムカテゴリ（本，映画，音楽）を入力する．検索キーワードをタイトルに含むアイテム（対象アイテム）が両マップビューに提示される．詳細マップビューには，対象アイテムに関連する任意の一つの地点が選択され，その地点周辺の詳細マップが表示される．このとき，同時にその地点周辺に関連するアイテムが提示される．広域マップビューには，対象アイテムに関連するすべての地点を含む範囲のマップが表示される．このとき，各地点に関連するアイテムも同時に提示される．ユーザは両マップビューにおいて，地図操作（移動，ズームイン，ズームアウト）を行うことができ，周辺のアイテムを探索することができる．また，アイテムのアイコンをクリックすることで，そのアイテムの詳細ページを閲覧することができる．

3.2 システム構成

図 3 にシステム構成を示す．以下，各処理について説明する．項目の番号は図中の番号に対応する：

(1)　Web からアイテムに関するデータを収集する．各種アイテムに関するデータはそれぞれのアイテムテーブルに格納する．データの収集には各種の WebAPI を用いる．本研究では，

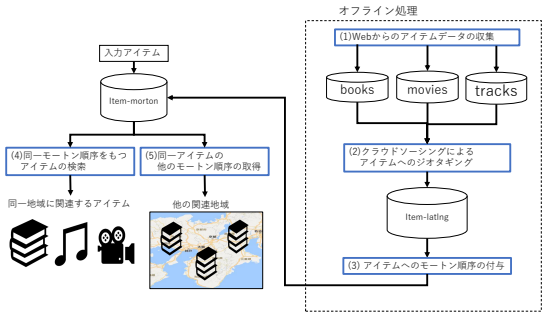

図 3　システム構成.

書籍データの収集には Google Books APIs [注2]，映画および楽曲データの収集には iTunes Search API [注3] をそれぞれ用いた.

(2)　収集したアイテムデータについて，関連する地域に基づき位置情報（ジオタグ）を付与する．アイテムとジオタグの関係をアイテム-緯度経度テーブル（4.1 節）に格納する．アイテムへの位置情報付与（ジオタギング）は，クラウドソーシングにより行う．クラウドソーシングによるアイテムへのジオタギングについては，4.2 節で説明する.

(3)　ジオタグに基づき，アイテムへのモートン順序を付与する．モートン順序を付与したアイテムを，アイテム-モートンテーブルに格納する．アイテムへのモートン順序付与については 4.4 節で説明する.

(4)　位置情報-アイテム推薦（5.1 節）を実行する.

(5)　アイテム-アイテム推薦（5.2 節）を実行する.

4.　ジオタグアイテムの管理

4.1　アイテム-緯度経度テーブル

小説，映画，楽曲のドメインごとに次のアイテム-緯度経度テーブルを用意する：

```
books－latlng(worker_id, book_id, latlng)
movies－latlng(worker_id, movie_id, latlng)
tracks－latlng(worker_id, track_id, latlng)
```

ここで，worker_id は後述するクラウドソーシングにおいてタスクを遂行したワーカの ID を表す．book_id，movie_id，track_id は，それぞれ小説，映画，楽曲のアイテム ID を表す．なお，既に図 3 の (1) の処理において，各ドメインのアイテムデータが各アイテムテーブル books テーブル，movies テーブル，tracks テーブルに登録されているものとする．book_id, movie_id, track_id は，それぞれ対応するアイテムテーブルを参照する外部キーである．latlng は緯度・経度を表し，実装上はジオメトリ型で扱われる.

4.2　クラウドソーシングによるアイテムへのジオタギング

小説や映画，楽曲の各ドメインに属するアイテムに対し，その内容に関連する位置情報を付与する．ここで，内容に関連する位置情報の例として，ドメインごとに次のように想定している：

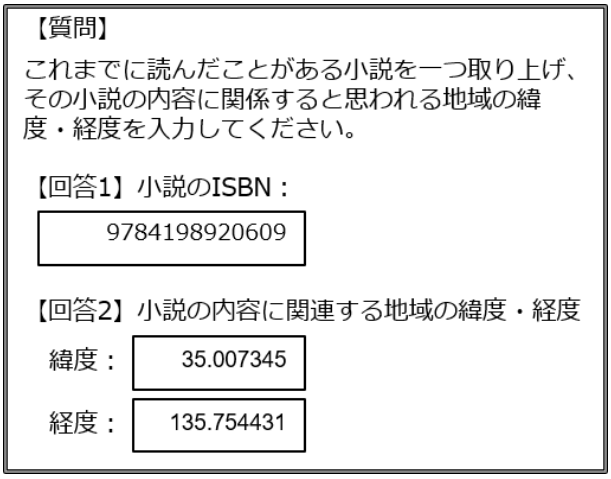

図 4　クラウドソーシングによるアイテムへのジオタギングタスクのイメージ.

小説　小説の舞台となっている地域や小説の中で紹介されている地域，主人公ゆかりの地域など.

映画　映画の舞台となっている地域や映画のモデルとなった地域，映画の中で言及されている地域，主人公ゆかりの地域，映画のロケ地など．ただし，製作国や公開国，監督の出身地など，映画の内容に関係しないものは，関連する地域とはみなさない.

楽曲　この地域が歌詞に現れる，この地域をテーマとして作曲された，この楽曲がテーマソングとして使われたドラマや映画の舞台になっているなど．ただし，楽曲が演奏されたコンサート会場やアーティストの出身地など，楽曲の内容に関係しないものは，関連する地域とはみなさない.

以上のような内容に関連する位置情報をアイテムに付与するためには，そのアイテムに関する十分な知識が必要となる．そこで，本研究では，クラウドソーシングによりアイテムへの位置情報付与を行う．なお本稿では，アイテムに位置情報を付与することをジオタギング，位置情報が付与されたアイテムをジオタグアイテムと定義する.

本研究では，先行研究 [1] で収集したジオタグアイテムを用いる．以下では，そのジオタグアイテムの収集方法について述べる.

4.2.1　タスク設定

図 4 にクラウドソーシングによるアイテムへのジオタギングタスクのイメージを示す．図には記入例も含めている．図の例では，ワーカが小説『信長新記 1 本能寺炎上』(ISBN:9784198920609) に対し，京都の本能寺辺りの緯度・経度を付与していることを表す．なお，緯度・経度の参照には，外部サイト『Google マップで緯度・経度を求める』[注4] を利用してもらった．また，ワーカには，先に挙げたアイテムの内容に関連する位置情報の考え方に加え，下記の基準を同タスク画面で示している：

(注2)：https://developers.google.com/books/

(注3)：https://affiliate.itunes.apple.com/resources/documentation/itunesstore-web-service-search-api/

(注4)：http://user.numazu-ct.ac.jp/tsato/webmap/sphere/coordinates/advanced.html

- 広い地域に関連すると思われる場合は，その地域内の任意の地点を入力すること，
- 複数の地域に関連すると思われる場合は，同一の小説について複数回に分けて回答すること．

映画および楽曲についても同様にタスクを実施した．

4.3 アイテム-緯度経度テーブルへの登録

本タスクにより得られたデータを先に用意したアイテム-緯度経度テーブルに登録する．つまり，各テーブルにはクラウドソーシングのワーカの ID，対象アイテムの ID，登録緯度・経度を一つのタプルとして登録される．ここで，一人のワーカは同一のアイテムに対し異なるジオタグを付与することが可能であるため，重複するワーカ ID およびアイテム ID の組合せが出現する可能性もある．

4.4 ジオタグに基づくアイテムへのモートン順序の付与

与えられた緯度・経度 (x, y) について，(x, y) をモートン順序 z に変換する．ここでは，最大探索レベルをレベル 12 とする．(x, y) をレベル 12 のモートン順序 z_{12} に変換する方法は下記のとおりである：

(1) (x, y) を左上を原点としたときのレベル 0 における座標 (x_0, y_0) に変換する．すなわち，(x_0, y_0) は次式で求める：

$$(x_0, y_0) = (180 + x, 90 - y) \tag{1}$$

(2) レベル 12 における座標 (x_{12}, y_{12}) を求める．レベル 12 における一つのセルの幅および高さは，それぞれ $360/2^{12}, 180/2^{12}$ で求められる．したがって，(x_{12}, y_{12}) は次式で求められる：

$$(x_{12}, y_{12}) = (\frac{x_0}{360/2^{12}}, \frac{y_0}{180/2^{12}}) \tag{2}$$

(3) (x_{12}, y_{12}) を 2 進数に変換した $(x_{12(2)}, y_{12(2)})$ を得る．

(4) $(x_{12(2)}, y_{12(2)})$ について，$y_{12(2)} \rightarrow x_{12(2)}$ の順に先頭から 1 ビットずつ取り出し，新たな 2 進数 z_{12} を作り出す．ここで得られた z_{12} が (x, y) に対応するレベル 12 におけるモートン順序である．

例えば，上記の手順にしたがうと，経度・緯度 $(134, 34)$ のモートン順序は $z_{12} = 0111\,0001\,0111\,0111\,1001\,1000_{(2)}$ となる．このとき，親レベルにおけるモートン順序は次式により容易に算出することができる：

$$z_l = z_{12} >> 2(12 - l) \tag{3}$$

ここで，$>>$ はビットシフト演算を表す．例えば，レベル 11 におけるモートン順序は $z_{11} = 01110001011101111001 10_{(2)}$ となり，レベル 8 におけるモートン順序は $z_8 = 0111\,0001\,0111\,0111_{(2)}$ となる．

以上の処理を用いて，すべてのジオタグアイテムにモートン順序を付与する．付与したモートン順序は次のアイテム-モートン順序テーブルで管理する：

```
books－morton(worker_id, book_id, morton12)
movies－morton(worker_id, movie_id, morton12)
tracks－morton(worker_id, track_id, morton12)
```

ここで，morton12 はレベル 12 におけるモートン順序である．このように，テーブルにはレベル 12 におけるモートン順序のみを登録しておくだけで良い．

5. ジオタグベースアイテム推薦

クラウドソーシングを用いて収集した各アイテムには，位置情報が付与されている．この位置情報を基にアイテムの推薦を行う．推薦方法として，(a) 位置情報－アイテム推薦，(b) アイテム－アイテム推薦の二つを提案する．

5.1 位置情報-アイテム推薦

位置情報として経度・緯度 (x, y) が入力されたとき，モートン順序が一致するアイテム集合を推薦リスト $\mathcal{L}$ として提示する．具体的なアルゴリズムを Algorithm 1 に示す．

Algorithm 1 位置情報-アイテム推薦

Require: 経度・緯度 (x, y)
Ensure: 推薦リスト $\mathcal{L}$

```
1: function LOCATIONTOITEMRECOMMENDATION(x, y)
2:     z ← toMorton(x, y)
3:     L ← []
4:     while z.length > 0 do
5:         L.append(getNearestItemList(z))
6:         if L.size ≧ k then
7:             break
8:         end if
9:         z ← z >> 2
10:    end while
11:    return L
12: end function
```

まず，(x, y) をモートン順序 z に変換する（2 行目）．つづいて，推薦リスト $\mathcal{L}$ を空にしておき（3 行目），4 行目から 10 行目のループにおいて推薦リストにアイテムをを追加していく．ここで，`getNearestItemList()` はアイテム-モートン順序テーブルからモートン順序 z に一致するアイテム集合を返す．例えば，小説を探索する場合は，下記の SQL を実行することでアイテム ID のリストが得られる：

```
SELECT book_id FROM books-morton WHERE
    morton12 LIKE '[z]%';
```

映画および楽曲についても同様である．得られたアイテム ID のリストを $\mathcal{L}$ に追加する（5 行目）．推薦アイテム数が一定数 k に達した場合は探索終了となる．一方で，推薦アイテム数が k に満たない場合は，モートン順序のレベルを一つ下げて（つまり，親空間において）探索を繰り返す（8 行目）．以上の処理を小説，映画，楽曲の各ドメインについて実行する．

5.2 アイテム-アイテム推薦

4.3 節で述べたように，アイテムによっては複数のジオタグが付与されているものもある．アイテム-アイテム推薦では，一つのアイテムが選択されたとき，そのアイテムに付与されている各々のジオタグに関連するアイテムを推薦する．具体的なアルゴリズムを Algorithm2 に示す．

Algorithm 2 アイテム-アイテム推薦

Require: アイテム ID i
Ensure: 推薦リスト $\mathcal{L}$
1: **function** ITEMTOITEMRECOMMENDATION(i)
2: $Z = \text{getMortonList}(i)$
3: **for each** $z \in Z$
4: $\mathcal{L}$.append(getNearestItemList(z))
5: **end for**
6: **return** $\mathcal{L}$
7: **end function**

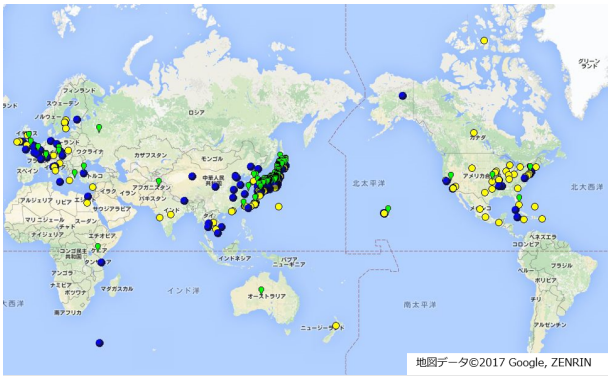

図 5　ジオタグアイテムの分布.

まず，アイテム i に付与されているモートン順序のリストを取得する（2 行目）．ここで，`getMortonList()` はアイテム-モートン順序テーブルからアイテム ID が i であるモートン順序のリスト Z を返す．例えば，小説テーブルを参照する場合，アイテム ID が i であるモートン順序は次の SQL で取得できる：

```
SELECT morton12 FROM books-morton WHERE
    book_id = [i];
```

得られた各モートン順序 $z \in Z$ について，z と同一のモートン順序をもつアイテムリストを取得し，推薦リスト $\mathcal{L}$ に追加する（4 行目）.

6. 評価実験

提案システムの有用性を検証するために評価実験を行った．6.1 節では，本実験で用いるデータセットについて説明する．6.2 節では，このデータセットを用いて，位置情報-アイテム推薦およびアイテム-アイテム推薦を実行したときの結果について定性的に評価する．さらに，これらの推薦手法を組み合わせることによる連想型クロスドメイン推薦の結果について考察する．

6.1 データセット

4.2 節で述べたとおり，クラウドソーシングによりアイテムへのジオタギングを行った．本実験では，小説，映画，楽曲の三つのドメインにおいて，それぞれ 10,000 件ずつタスクを依頼した．このうちアイテム ID に不整合がある，緯度・経度が有効な値でないなど，不備のあるデータを排除した．得られた有効データ数は，小説 9,945 件，映画 9,417 件，楽曲 9,668 件となった．ユニークデータ数は，小説 6,901 件，映画 5,279 件，楽曲 6,000 件であった．

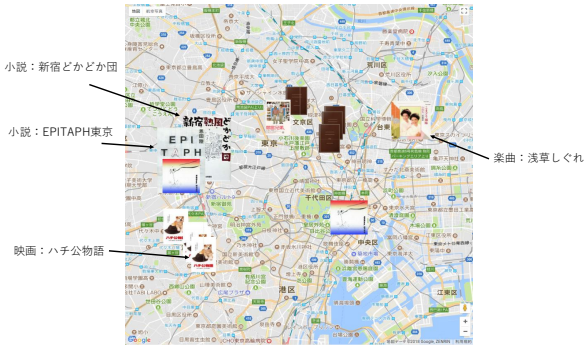

図 6　位置情報-アイテム推薦の実行例．東京のある地点 (139.716278, 35.685734) を入力したときの推薦リストを地図上に示している．

ジオタグアイテムの分布は図 5 のとおりとなった．今回はタスク依頼対象が日本人であったため，日本にデータが集中したことは当然である．一方で，ヨーロッパやアメリカ，その他の地域にもジオタグが付与されていることがわかる．

6.2 定性評価

6.2.1 位置情報-アイテム推薦

位置情報-アイテム推薦は経度・緯度 (x, y) が入力されたとき，モートン順序が一致するアイテム集合を推薦リスト $\mathcal{L}$ として提示する手法である（5.1 節）．本項では，位置情報-アイテム推薦手法について，東京のある地点および沖縄のある地点を入力としたとき，提示される推薦リスト $\mathcal{L}$ を定性的に評価する．

図 6 は，東京のある地点 (139.716278, 35.685734) を入力としたときの推薦リストを示したものである．「EPITAPH 東京」や「新宿熱風どかどか団」といった東京が舞台となった小説が推薦された．また，「ハチ公物語」や「浅草しぐれ」などの映画や楽曲が推薦された．これらも東京が舞台になっている．このように，東京においては小説，映画，楽曲の多くのアイテムに対しジオタグが付与されており，多様なアイテムが推薦された．

一方で，図 7 は沖縄のある地点 (127.902542, 26.604172) を入力としたときの推薦リストである．図 7 のように，この地点においては「天国からのエール」という映画が推薦された．しかしながら，提示されたアイテムはこのアイテムのみであった．

以上のように，入力する地域によって，推薦されるアイテムの多様性が異なる．今回はクラウドソーシングによりジオタギングを行ったが，対象とするアイテムおよび地域はワーカにより任意に選択させた．そのため，有名なアイテムや多くのワーカに馴染みのある地域などが選ばれやすかった．今後は，多様なアイテム，多様な地域においてジオタグアイテムを効率よく収集するための仕組みやタスク設定を工夫する必要があると考えられる．

6.2.2 アイテム-アイテム推薦

アイテム-アイテム推薦は，一つのアイテムが選択されたとき，そのアイテムに付与されている各々のジオタグに関連するアイテムを推薦する手法である（5.2 節）．本項では，アイテム-アイテム推薦について，映画「モスラ」を入力としたとき，提示される推薦リスト $\mathcal{L}$ を定性的に評価する．

図 8 は，入力アイテムを映画「モスラ」としたときの推薦リストである．映画「モスラ」は東京が舞台であるため，東京を中心にアイテムが推薦されている．例えば，「秒速 5 センチメートル」は東京を舞台にした映画である．また，名古屋周辺にもアイテムが推薦されていることがわかる．これは，同シリーズの映画である「モスラ対ゴジラ」が名古屋を舞台にしていることから，あるワーカが「モスラ」に対しても名古屋をジオタグとして付与したためであるといえる．これに伴い，楽曲「あゝ名古屋城」など名古屋に関連するアイテムが推薦されている．このように，一つのアイテムから異なる地域のジオタグアイテムが推薦されることで，ユーザにとって気付きを与えることができるといえる．

6.2.3 連想型クロスドメイン推薦

以上の位置情報-アイテム推薦およびアイテム-アイテム推薦を連続的に実行することで連想的にアイテムを探索することが可能となる．例えば，例示した結果を用いると，下記のような連想的探索を実現できる：

(1) 興味のある地域として東京を入力し，小説「EPITAPH 東京」や映画「秒速 5 センチメートル」を発見する（位置情報-アイテム推薦）．

(2) 小説「秒速 5 センチメートル」を選択し，東京に関連する映画として「モスラ」を発見する（アイテム-アイテム推薦）．

(3) 映画「モスラ」を入力し，名古屋に関連する楽曲として「あゝ名古屋城」を発見する（アイテム-アイテム推薦）．

このように複数のドメイン，複数の地域にまたがった連想的な探索が可能となる．このような連想的探索を繰り返すことで，最初は予期していなかったアイテムに巡り合う可能性が高まり，セレンディピティ効果を誘発することも期待できる [6]．今後は，連想型クロスドメイン推薦のセレンディピティ効果について検証する．

7. おわりに

本稿では，ジオタグベースのモートン順序を用いたクロスドメイン推薦システムを提案した．このシステムは，(a) 位置情報-アイテム推薦，および (b) アイテム-アイテム推薦の二つの機能をもつ．

定性的な評価実験により，(a) 位置情報-アイテム推薦においては，東京のような地域においては多様なアイテムが推薦されやすいことが確認できた．その一方で，推薦アイテムの多様性が低い地域もみられた．今後は，多様なアイテム，多様な地域においてジオタグアイテムを効率よく収集するための仕組みやタスク設定を工夫する必要がある．

(b) アイテム-アイテム推薦においては，一つのアイテムから異なる地域のジオタグアイテムも推薦されることが確認できた．このように，一つのアイテムから異なる地域のジオタグアイテムが推薦されることで，ユーザにとって気付きを与えることができるといえる．

さらに，(a) と (b) を組み合わせることによる連想型クロスドメイン推薦について考察した．特に，連想的探索を繰り返すことにより，最初は予期していなかったアイテムに巡り合う可能性が高まることから，セレンディピティ効果を誘発することへの期待について考察した．

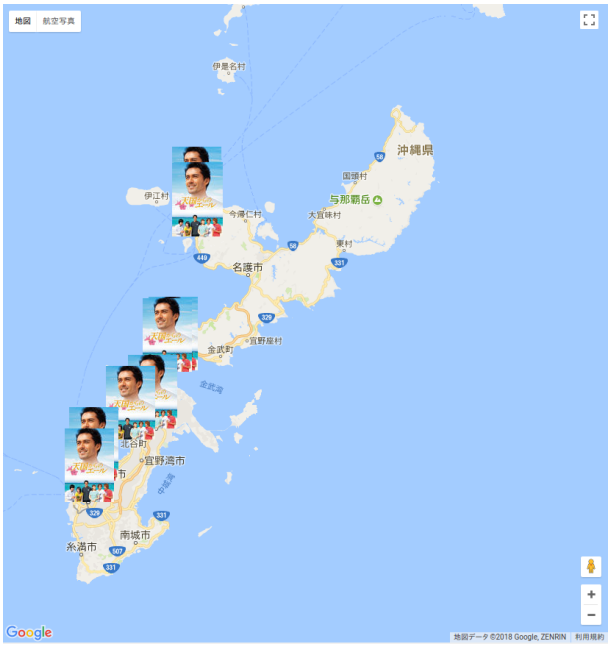

図 7 位置情報-アイテム推薦の実行例．沖縄のある地点 (127.902542, 26.604172) を入力したときの推薦リストを地図上に示している．

図 8 アイテム-アイテム推薦の実行例．入力アイテムを「モスラ」としたときの推薦リストを地図上に示している．

今後は，本システムを用いることによる有用性について，新規性，多様性，セレンディピティの観点から定量的に検証を行う．さらに，小説，映画，楽曲に限定せず，より多様なドメインを対象としてジオタグアイテムを収集する．

謝　辞

本研究は JSPS 科研費 JP16HO5932 の助成を受けたものです．ここに記して謝意を表します．

文　献

[1] 北村統太, 奥健太. アイテムのジオタギングに基づく観光スポットベースクロスドメイン推薦システム. In *DEIM Forum 2017*, pp. B2–2, 2017.

[2] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recom-*

mender Systems Handbook. Springer, 2011.

[3] Ignacio Fernández-tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. Cross-domain recommender systems : A survey of the State of the Art. In *Proceedings of the 2nd Spanish Conference on Information Retrieval2*, pp. 187–198, 2012.

[4] Shaghayegh Sahebi and Peter Brusilovsky. It Takes Two to Tango : an Exploration of Domain Pairs for Cross-Domain Collaborative Filtering. In *[RecSys2015]Proceedings of the 9th ACM conference on Recommender systems*, pp. 131–138, 2015.

[5] P. Cremonesi, A. Tripodi, and Turrin. R. Cross-domain recommender systems. In *ICDMW2011: IEEE 11th International Conference on Data Mining Workshops*, pp. 496–503, 2011.

[6] 奥健太. セレンディピティ指向情報推薦の研究動向. 知能と情報（日本知能情報ファジィ学会誌）- 特集：Web インテリジェンスとインタラクション II -, Vol. 25, No. 1, pp. 2–10, 2013.