

修士学位論文

大規模画像を用いた旅行ルート推薦とその個人化

平成 28 年度

東京大学大学院 情報理工学系研究科

電子情報学専攻

48-146458

小澤 真里奈

指導教員 山崎俊彦 准教授

内容梗概

近年, SNS サイトの人気が高まり, Flickr などの画像共有サイトには位置情報や撮影日時等のメタデータが付けられた写真が多数アップロードされている. このような大量のメタデータから分かる様々な情報を考慮することで, 個人や個別の場合に適した旅行ルートを推薦することが出来る. 本研究では, 気温と天気を考慮した旅行ルート推薦を提案する. これは, ベイズの定理を用いてマルコフモデルに気温と天気の情報を組み込むことで実現され, 人気の場所を提示する単純な推薦と比較した場合の推薦精度の向上を目指す.

目次

第一章 序論.....	4
1.1. 背景.....	4
1.2. ジオタグ.....	6
1.3. 関連研究.....	6
1.3.1. 旅行ルート推薦にまつわる種々の研究.....	6
1.3.2. 最近の旅行ルート推薦研究.....	8
1.3.3. 本研究との関連度が高い旅行ルート推薦研究.....	9
1.4. 構成.....	10
第二章 データの収集.....	12
2.1. Flickr からの画像およびメタデータの収集.....	12
2.1.1. Flickr API (1): flickr.photos.search.....	13
2.1.2. Flickr API (2): flickr.photos.getFavorites.....	16
2.1.3. Flickr API (3): flickr.photos.comments.getList.....	16
2.1.4. 3つのAPIを用いたFlickrからの画像とメタデータの取得.....	17
2.2. 気温と天気データの取得.....	18
2.2.1. 気象庁のサイトからの国内気象データの取得.....	18
2.2.2. 海外の気象情報について.....	20
第三章 提案手法.....	21
3.1. ステップ(1): データの収集.....	21
3.2. ステップ(2-1): 位置情報の解析.....	21
3.3. ステップ(2-2): その他付加情報のラベルの割り当て.....	23
3.3.1. 気温・天気ラベルの割り当て.....	24
3.3.2. 季節・時間帯ラベルの割り当て.....	24
3.4. ステップ(3): 推薦システムの実現.....	25
3.4.1. 推薦モデルの構築.....	26
3.4.2. 推薦モデルに基づく実際の処理の流れ.....	27
3.5. ステップ(4): 旅行ルートの推薦.....	30
第四章 実験.....	31
4.1. 実験の設定.....	31
4.1.1. 実験方法.....	31
4.1.2. ベースライン.....	32
4.1.3. その他の設定.....	33
4.2. 実験に用いるデータセット.....	34
4.3. 実験条件と実験結果.....	35

4.3.1 実験の条件.....	35
4.3.2 実験結果	37
4.4. 考察	47
4.4.1. ランドマークの特徴から捉える実験結果	47
4.4.2. データ数から捉える実験結果.....	49
4.4.3. 推薦手法の将来性.....	49
第五章 結論	54
5.1. まとめ	54
5.2. 今後の展望.....	54
参考文献	55
発表文献	57
謝辞.....	58

第一章 序論

1.1. 背景

近年、デジタルカメラやカメラ機能付き携帯の普及により、いつでも気軽に写真を撮影することが可能となった。そして撮られた写真は、Flickr [1] や Instagram [2] (図 1) のような画像共有を目的とした SNS サイトにアップロードすることで簡単に共有することができる。最近の SNS サイトの人気の上昇に伴い、そのような画像共有サイトには莫大な数の画像が日々アップロードされ続け、大規模な画像群を形成している (図 2, 図 3, 表 1)。

基本的に写真を撮影するというのは、何か自分が魅力的だと感じたものを記録するための行為であるため、大規模画像を利用すれば何か有益な情報を得ることが期待できる。また、上記のような画像共有サイトには、画像に撮影日時や位置情報、タグやタイトルといったメタデータを付与する機能があり、サイト上の画像はもはや「単なる一枚の画像」ではなく、様々な情報を持つ貴重な資源となっている。

以上のように非常に豊かな情報を持った大規模画像の利用法の一つとして近年盛んに研究されている分野に、旅行ルート推薦がある。ここで旅行ルート推薦とは、旅行を予定している人に対し旅先で訪問すべき名所などを提案することや、現在旅行中である場所を訪れている人に対し次に訪問すべき場所を提示することとする。無論、インターネットで検索すれば、行き先で人気の観光スポットを容易に調べることができる。しかし、インターネットでの検索では、単純に「一般的に人気のある場所」が羅列される場合が殆どであり、多くの観光スポットが提示された中で結局どこを訪れるべきか判断がつかないことも多い (これを **information overload** [3] という)。そこで、個々のユーザーに適した場所を判別して提示する旅行ルート推薦、すなわち「個人化された旅行ルート推薦」が求められる。この個人化された旅行ルート推薦は、上述の大規模画像とそのメタデータに対しデータマイニングの手法を用いて旅行の内容を左右し得る何らかの情報を抽出し、抽出された情報を推薦に活かすことで実現される。例えばこれまでには、個人の趣味・嗜好を考慮した旅行ルート推薦 [4] や、季節・時間帯を考慮した旅行ルート推薦 [5] などが行われてきた。本稿では、これらの研究を踏まえた上で、気温・天気を考慮した旅行ルート推薦を提案し、Flickr から収集した画像に対して行った実験の結果を示す。例えば、よく晴れた過ごしやすい気温の日は屋外施設の訪問に適しているが、雨の日や極端に暑い日には屋内施設が好まれるなど、気温・天気は旅行の内容を左右し得る要素であると考えられるので、この 2 つの要素を考慮することで、推薦精度の向上が期待できる。



図 1：画像共有サイトのアイコン
(左：Flickr, 右：インスタグラム)

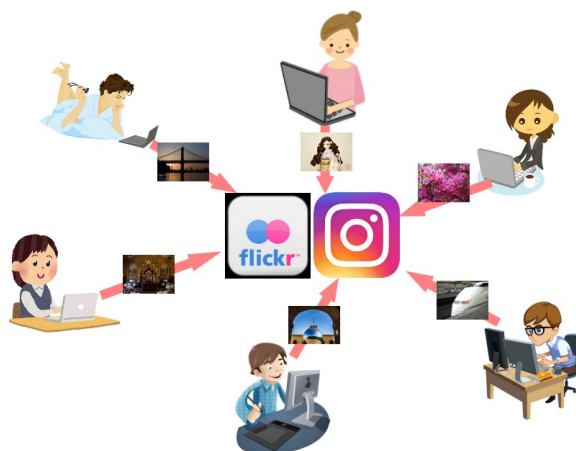


図 2：大規模画像の生成

表 1：主要な SNS サイトのユーザー数 (<http://www.uniad.co.jp/260204>, <http://expandedramblings.com/index.php/flickr-stats/>)

Flickr	Instagram	Facebook	twitter	Line
1.22 億人 (2016 年 11 月)	5 億人 (2016 年 6 月)	17.1 億人 (2016 年 6 月)	3.1 億人 (2017 年 1 月)	2.2 億人 (2015 年 12 月)

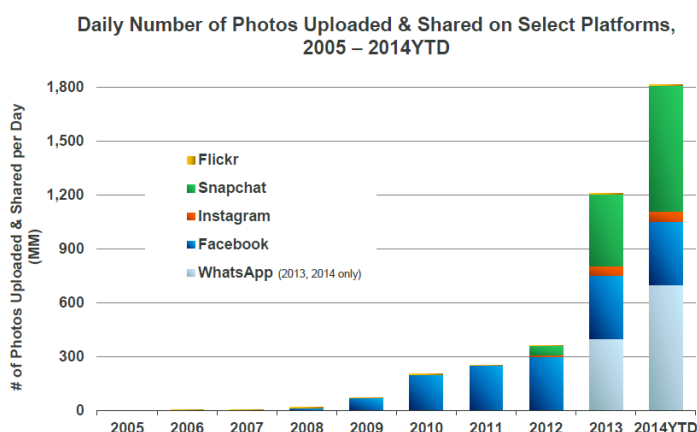


図 3：画像共有サイトにおけるアップロード数の推移
(http://kpcbweb2.s3.amazonaws.com/files/85/Internet_Trends_2014_vFINAL_-_05_28_14-_PDF.pdf?1401286773)

1.2. ジオタグ

前項で述べた通り, Flickr などの画像共有サイトには画像に撮影日時やタグ等の様々なメタデータを付与する機能がある. そのような多様なメタデータの中で, 旅行ルート推薦の研究において特に重要なのが「ジオタグ」である. ジオタグとは画像の撮影位置を緯度と経度で表したものであり, 最近のスマートフォンには画像の撮影時に自動でジオタグを付けるものも多く, Flickr 等のサイト上にはジオタグ付き画像が非常に豊富に存在する (これを Volunteered Geographic Information (VGI) [6] という). この VGI からはその記録媒体の詳細な位置や軌跡を知ることができ, 学術的にも極めて有用なもので, イベント検出や災害予測, 物流など様々な場面で利用されている. 旅行ルート推薦もその恩恵を受けている分野の 1 つである.

1.3. 関連研究

本節では, 関連研究を 3 つに分けて紹介する. 最初に一般的な旅行ルート推薦の研究について簡単に触れ, 次に比較的最近のもので新規な点や興味深い点があるものについて述べ, 最後に本研究と特に関係が深いものを説明する.

1.3.1. 旅行ルート推薦にまつわる種々の研究

旅行ルート推薦の研究は, SNS サイトの人氣が爆発的に高まり始めた 2010 年頃から盛んになってきた. 当初からこの分野において広く使われている技術に, 大手通販サイト Amazon でも使用されていることで有名な Collaborative filtering がある [7][8].

Zheng et al. [9] は, を各観光都市の地図上に描いたバウンディング・ボックスの範囲内で撮影されたジオタグ付き画像を Flickr から収集し, それらのバウンディング・ボックス内から一定の枚数以上の画像を持つものを人氣の場所として抽出した. そして各ユーザーが過去にある都市内でどの場所を訪問したかという情報を基に, 他の都市ではどのような場所を好みそうかを Collaborative filtering を用いて予測し, 推薦した. これは基礎的な技術を用いたシンプルな旅行ルート推薦システムであり, この研究は旅行推薦の研究の先駆けの 1 つといえる.

Mamei et al. [10] は, 「この activity をしたいのならばこの location がおすすめ」という「location recommendation」と, 「この location に行くのならばこの activity をするのがおすすめ」という「activity recommendation」の 2 つの推薦を提案した. これらの推薦を実現するために, location と activity の関連性を示す location-activity matrix を生成したが, 入手できるデータの制約からこの行列が疎になってしまうため, それを補うためにその場所にはどのような Point of Interest (POI) があるのかを示す location-feature matrix と activity 同士の共起性を示す activity-activity matrix の 2 つの行列も考慮する. そして

これらの3つの行列を Collaborative filtering に基づいた手法で融合し、上記の2種類の推薦に活用する。

Zahalka et al. [11] は, New Yorker Melange というシステムを構築し, ニューヨークへの旅行を計画しているユーザーに対し, 似た好みを持つニューヨーク市民が気に入っている場所を提示するという旅行ルート推薦システムを提案した. この研究でもユーザー同士の好みの類似性を利用しているが, 用いた手法は Collaborative filtering ではない. まず写真に対し deep convolutional neural network [12] を用いて視覚的特徴量を, メタデータであるタグやタイトルにトピックモデリング手法の一つである Latent Dirichlet Allocation (LDA) [13] を適用してテキスト特徴量を抽出する. 次にそれぞれの特徴量に対し k-means のクラスタリング手法によってクラスタリングを行い, 各クラスタの重心がユーザーの趣味を表していると見做して, 各ユーザーの好みの推定を行う. そしてユーザーによる選択と, 推薦する場所や似たユーザーの提示を交互に繰り返し, 対話式の推薦を実現している.

Jiang et al. [14] は, ContextRank というシステムを提案し, 画像の持つ多様なコンテキストを利用することを考えた. 利用したのは場所の人気度, ジオタグ, 画像の内容の類似性, タグの類似性, そして Collaborative filtering により求めたユーザー同士の類似性であり, これらの4つのコンテキストの情報をランキングのアルゴリズムによって融合し, 推薦する場所を決定する. この手法はデータの少ないユーザーに対して特に有効であるという実験結果が出ており, 推薦・予測の研究において常につきまとうコールドスタート問題に対応できるという利点がある.

Meehan et al. [15] も同様に, 旅行推薦に複数のコンテキストを利用することを提案した. この研究で利用したのは location, time, weather, social media sentiment, personalization の5つである. この中で, social media sentiment とは, 短文投稿サイト Twitter 上のツイートの解析によって決定した場所の「mood」のことであり, positive, negative, neutral の3つに分類される. また, personalization とは年齢や性別などのユーザー自身の性質のことである. これらの5つのコンテキストを Neural Network, Fuzzy Logic, PCA の3つを使って融合し, 推薦場所を決定する.

旅行においては観光スポットを回るルートも重要となってくるため, 訪問場所だけではなくルートを提案する研究もある. Ji et al. [16] は, ユーザーの現在地と目的地を入力として与えると, 最短でなおかつ出来る限り多くの観光名所を通ることができるルートを検索して提示する. この最短ルートの検索には, 最短経路問題で広く使われるダイクストラ法を用いている.

Arase et al. [17] は, 旅行の目的を考慮した旅行ルート推薦システムを構築した. 旅行の目的とは, 例えば有名な観光スポットを巡ることや, 美味しい食事をとることなどで, 目的が違えば行き先は変わってくるため, 旅行ルート推薦を個人化する際に有用な要素であるといえる. この研究では, まず, 写真共有サイトから集めた写真の集合を, メタデータから

分かる撮影日時や撮影位置などに基づいて旅行ごとに区切っていき、旅行の集合を作る。次にジオタグから得られる緯度と経度の情報を基に、それぞれの写真が撮影された都市の名前を割り出す。そして旅行をテーマ（旅行の主な目的）に応じて分類し、TAS mining algorithm [18] を用いてテーマごとに頻出する典型的な旅行パターンを検出し、ユーザーに提示する。提示の際には、メタデータの一つであるタグについてマイニングを行い、各旅行パターンの説明となるようなタグを抽出し、旅行パターンと共に提示している。

1.3.2. 最近の旅行ルート推薦研究

Shen et al. [19] は、複数の SNS サイトから収集した multi-modality な情報を活用することを提案した。具体的には、Flickr から取得した画像の visual feature に基づく content-based knowledge, Wikitravel の記事と Tripadvisor [20] に寄せられたコメントから得たテキスト情報に基づく semantic-based knowledge, Yahoo と Tripadvisor 上の観光名所に対する rating に基づいた social-based knowledge の 3 つのモダリティの情報を考える。これらの情報はグラフの理論を利用して融合され、Laplacian Support Vector Machine [21] によって、各観光名所はユーザーにとって”favorite”か”unfavorite”かに分類される。そして最後に、ユーザーの現在地というコンテキストの情報を加味して、最終的な観光名所のランキングを作成し、推薦する。この研究の興味深い点は、複数の SNS サイトから取得した情報を効果的に利用していることである。複数の SNS サイトから得た情報の利用は Zahalka et al. [11] も行っているが、こちらの研究の方が利用するサイトの種類や情報の利用の仕方がより多様である。

Sun et al. [22] は、ランドマークとそこへ行くためのルートを合わせて提案するシステムを構築した。従来ではルートを提案する場合、画像のジオタグを基にユーザーが過去に通った軌跡を割り出し、それらの軌跡に対してマイニングを行って最適なルートを抽出するというのが主流であった。しかしこの研究では、ルートを提案する際の単位を「過去のユーザーの軌跡」ではなく「実際の road」としている点に新規性がある。システムの流れとしては、まず画像から DBSCAN [23] というクラスタリング手法によってランドマークを抽出し、ランドマークを人気度によってランキング付けして推薦するランドマークを決定する。続いてそのランドマークへのルートを決定するが、ここでは road の人気度、road 周辺にある POI のユーザビリティ、road の長さの 3 つが考慮される。先ほど述べた通りこの研究の主な新規性はルートの単位を road としている点にあるが、それ以外にもこの研究には興味深い点が複数ある。まず、この研究では、様々なノイズが考慮されている。ノイズとは例えば居住者が撮影した旅行には関係のない画像や、イベント等により画像が多く撮影された場所（つまり場所自体に魅力があるとは限らないもの）、高い建築物等を遠くから撮影したことによるジオタグが示す位置とタグが示すランドマーク名との食い違いなどであり、細かな点にまで気が配られている。上述のクラスタリング手法 DBSCAN [23] も、クラスタを形成するのに最低限必要な画像の枚数を指定することができ、ランドマークと見做

すにはあまりにも画像の枚数が少ないクラスタが生成されることを予め防ぐことができる点において、一種のノイズの除去に一役買っていると考えられる。次に、ランドマークや road の人気度を求める際には、単純にそこで撮影された画像の枚数やそこを訪れたユーザーの数などを用いるのではなく、「各場所で最初に撮影された画像が、最もその場所の人気度に寄与しており、2 枚目以降は徐々に貢献度が下がっていく」という仮定のもと、対数を使った人気度の計算式を考案している。さらに、旅行の知識が豊富なユーザーが投稿した画像とあまり旅行に詳しくないユーザーが投稿した画像の価値やタグ付けの正確性の違い (Inter-tourist difference) にも触れている。この Inter-tourist difference については Zheng et al. [24] も言及しており、利用するデータの品質に関わる重要な要素であるが、現段階ではユーザーの旅行経験に関するデータを入手するのが困難なためシステムに組み込むのは難しく、旅行推薦の分野の今後の課題といえる。

1.3.3. 本研究との関連度が高い旅行ルート推薦研究

Kurashima et al. [4] は、個人の趣味を考慮した旅行ルート推薦を提案した。この研究では、まずジオタグから得られる緯度と経度の情報について mean-shift のクラスタリング手法 (第 3 章参照) を用いて人気のある場所を検出し、人気のある場所のシーケンスとして表される撮影者の過去の訪問履歴を生成する。そして、この訪問履歴に対しトピックモデリングの手法の一つである Probabilistic Latent Semantic Analysis (PLSA) [25] を用いることで潜在的な個人の趣味・嗜好を考慮するモデルを作り、それを現在地を考慮するマルコフモデルと結合することで推薦モデルを構築し、目標の旅行ルート推薦を達成している (図 4)。

Cheng et al. [26] は、人の属性を考慮した旅行ルート推薦を行った (図 5)。人の属性としては性別・年齢・人種を扱っており、これらの属性が異なれば好む行き先が変わることは容易に想像出来るが、筆者らはエントロピーと相互情報量 [27] を用いてそのことを確かめた上で、このような推薦手法を構築した。この研究では、Arase et al. [17] 同様、まず写真共有サイトから入手した写真の集合をメタデータから得られる撮影日時を基に旅行ごとに分割する。そして Kurashima et al. [4] と同様ジオタグから分かる緯度と経度の情報に mean-shift クラスタリング手法を用いて人気のある場所を検出し、先の過程で得られた各旅行を人気のある場所のシーケンスに変換する。人の属性については、各写真に写っている人物の顔領域を、SVM (Support Vector Machine) などの画像認識技術で解析することで検出する。旅行ルート推薦手法においては、写真のメタデータのみを解析して写真自体は直接利用しないものも多いが、この研究では写真も活用されている。最後に、ベイズの定理を用いてマルコフモデルに人の属性情報を組み込んだ推薦モデルを構築し、グラフの問題に準えて推薦する旅行ルートの生成を行い、ユーザーに提示する。

以上述べた 2 つの研究の推薦モデルは、いずれもマルコフモデルに基づいている。詳細は後述するが、本研究の推薦モデルもマルコフモデルに基づいているため、これらの研究と

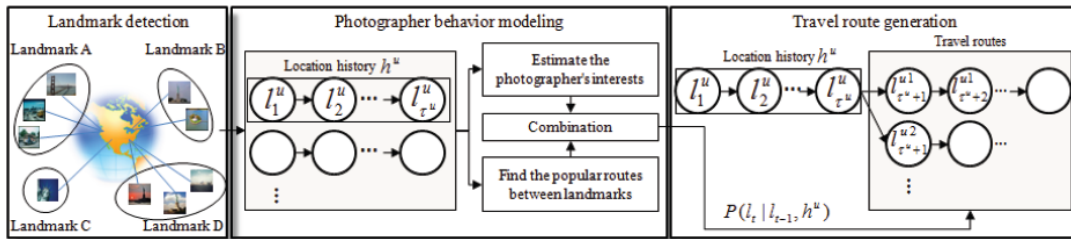


図 4 : Kurashima et al. [4] の提案システム



図 5 : 人の属性を考慮した旅行ルート推薦システム [26]

融合させることも推薦システムの将来的な発展の 1 つとして考えられる。

Yamasaki et al. [5]は、ランドマークの季節的・時間的な魅力度の変化に着目し、季節・時間帯を考慮した旅行ルート推薦システムを提案した。この研究は過去に本研究室で行われていた研究であり、本研究のベースとなっている。

1.4. 構成

本論文の構成を以下に示す。

- ・ 第一章：序論
- ・ 第二章：データの収集
- ・ 第三章：提案手法
- ・ 第四章：実験
- ・ 第五章：結論

第一章では、画像共有サイト上の大規模画像とそのメタデータの利用法の 1 つとして旅行ルート推薦システムを挙げ、関連研究を紹介した。第二章では、提案システムに利用する画像とメタデータ、および気象データの収集方法について述べる。次に第三章では本稿が提

案する「気温・天気を考慮した旅行ルート推薦システム」の構築について説明する。続いて第四章では **Flickr** から取得した大規模なデータセットに対して行った実験とその結果について言及し、最後に第五章で本稿のまとめと今後の課題を述べる。

第二章 データの収集

この章では、提案する旅行ルート推薦システムの実現に必要なデータの収集について述べる。本研究で用いるデータは、主に「画像およびそのメタデータ」と「過去の気温・天気データ」の2種類である。それぞれのデータの収集方法について以下で詳述する。

2.1. Flickr からの画像およびメタデータの収集

本研究では、画像共有サイト Flickr から取得した画像とそのメタデータを利用する。本研究だけではなく、従来の旅行ルート推薦に関する研究には Flickr から取得したデータを利用しているものが多い。Instagram 等 Flickr 以外にも多くの画像共有サイトがあるにも関わらず、Flickr を利用する研究が多いことには、大きく分けて2つの理由がある。

1つ目は、Flickr がクリエイティブ・コモンズ・ライセンスを採用していることである。クリエイティブ・コモンズ・ライセンスとは、作品の著作者が自らの作品の再利用をどの程度のレベルまで許可するかを明示的に表示できるように策定されたもので、「表示」「非営利目的での利用のみに限定」「改変の制限」「派生作品へのライセンスの継承」の4つの項目に関する条件を組み合わせて設定される。なお、本研究で主に利用するメタデータに関しては利用に関する制約はなく、全ての画像のメタデータを自由に利用することができる。前章で紹介した関連研究の中には旅行ルート推薦システムをアプリケーション化しているものもあり、アプリケーション上でユーザーが投稿した写真を、例えば推薦するランドマークのイメージ画像等として表示しようとする場合などには、クリエイティブ・コモンズ・ライセンスにより定められた条件を参照し、利用が認められた画像のみを使用する必要がある。

2つ目の理由は、Flickr 自身が画像の検索などのための API を公開しており、これを使えば、研究に必要な画像とメタデータを効率的に取得できるということである。Flickr の API を利用した画像の取得に関しては、有名な先行研究があり [28]、画像取得のソースコードも公開されているが、この研究の当時とは API の仕様が大きく変わっており、これを用いることはできないため、独自に実装した。本研究で使用する API は、以下の3つである。

- flickr.photos.search
- flickr.photos.getFavoites
- flickr.photos.comments.getList

これら3つの API を含め、Flickr が公開する API を使用するには、予め Flickr の利用登録を済ませ、API Key を取得しておく必要がある。これらの API について、その使用法・使用上の注意等を含め次項以降で詳しく説明していく。

2.1.1. Flickr API (1): flickr.photos.search

flickr.photos.search は、ユーザーが指定した条件に合致する画像を検索し、検索結果をリストとして返す API である。この API では、テキスト検索や撮影日時による検索など、様々な条件での検索を行うことが可能だが、今回は検索条件として「撮影日時」「中心地点（緯度と経度）」「半径」の3つを指定し、「指定された期間内に、指定した中心地点から半径○km の範囲内において撮影された画像」を検索することとした（図 6）。したがって引数として指定すべき条件は表 2 の通りであり、また結果として返される画像の情報は表 3 のようになる。

結果として返された情報のうち、メタデータに該当するものについてはそのまま xml ファイルとして保存する。一方画像については結果として得た情報を基に画像の URL を指定して、改めてダウンロードをする必要がある。Flickr において、画像の URL は以下のように指定される。

“URL = https://farm[*farm*].staticflickr.com/[*server*]/[*id*].[*secret*].jpg”

上の文字列の[*farm*][*server*][*id*][*secret*]の中に flickr.photos.search の結果として得られた各値を入れることで URL が形成され、画像をダウンロードすることができる。

ここで、いくつかの注意点がある。まず、この API には、一度に大量の画像を検索しようとする場合に、途中から重複する画像が出てきてしまうという広く知られたバグがある。本研究室で調査したところ、およそ 4000 枚前後で重複が発生し始めるということであった。そこで今回は、この問題に対処するべく、「撮影日時」の条件を例えば「2004 年 1 月 1 日 0 時 0 分 0 秒～2004 年 1 月 1 日 23 時 59 分 59 秒」のように 1 日ずつ細かく区切り、その都度 flickr.photos.search を呼び出すこととした。これにより一回の検索で該当する画像の枚数が最大でも 2000 枚程度となり、東京で撮影された画像が 10,232 枚集まった時点で確認したところ、重複が認められなかったため、この方法でダウンロードを続行した。また、この方法には、ダウンロードの際にどの期間のデータを取得したかのログを残しておけば、何らかの理由でダウンロードが中断された場合などに、その期間のダウンロードのみ再行えばよいという利点もある。Flickr には多くのユーザーが関わっている関係上、ある期間の検索結果のリストを返されてから、それを基に画像を 1 枚ずつダウンロードしている間に、リストに入っていたはずの画像が削除されエラーとなってしまう場合が度々あるため、この利点によるメリットは大きい。次に注意すべき点は、Flickr は公式サイト上のこの API の説明のページにて引数「per_page」の最大値を 500 としているが、実際には指定できる最大の値は 250 であり、それより大きな値を指定するとエラーとなってしまうことである。これもバグだと考えられる。最後に、表 2 の通りこの API ではジオタグを持つ画像のみを検索するよう引数で指定ができるはずなのだが、検索結果にジオタグが付いていない画像が含まれることがまれにあるという点にも注意が必要である。このような画像は、後で自ら取り除かなければならない。

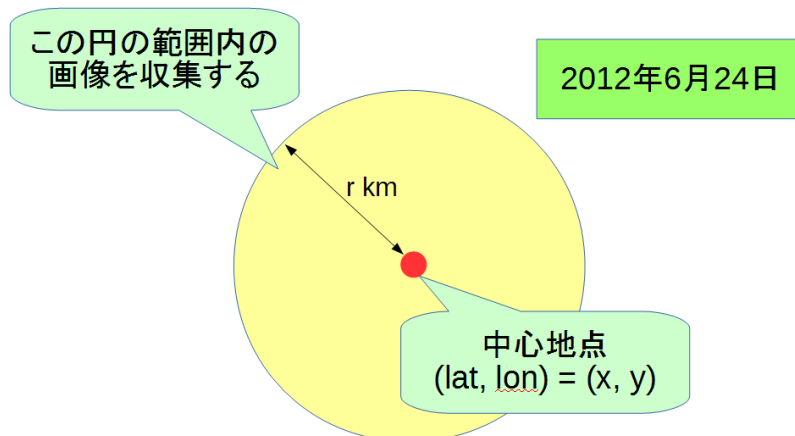


図 6 : 画像の収集範囲の指定

表 2 : flickr.photos.search に渡す引数

引数の名称	説明	今回指定した値
Method	method の名前	flickr.photos.search
api_key	API Key	事前を取得した API Key
has_geo	ジオタグ付きの画像のみを収集するか	1 (Yes)
lat	緯度	指定した中心地点の緯度
lon	経度	指定した中心地点の経度
radius	収集範囲を示す円の半径	32km
min_taken_date	画像を収集する期間の開始日	都度指定する
max_taken_date	画像を収集する期間の終了日	都度指定する
extras	追加で取得したい情報	表 3 参照
per_page	1 ページで取得する画像の枚数の上限	250 枚
page	何ページ目を取得するか	1

表 3 : flickr.photos.search によって得られるメタデータ

取得するメタデータ	説明
geotag	ジオタグ(緯度と経度)
title	画像のタイトル
tag	画像に付与されたタグ
description	画像の説明
date_taken	画像の撮影日時
date_upload	画像のアップロード日時
last_update	画像の最終更新日時
photo_id	画像の ID
owner	画像をアップロードしたユーザーの ID
owner_name	画像をアップロードしたユーザーの名前
license_id	画像のライセンスに関する ID (クリエイティブ・コモンズに対応)
views	画像の閲覧数
secret	画像の URL 生成に必要
server	画像の URL 生成に必要
farm	画像の URL 生成に必要
icon_server	ユーザーのアイコンの URL 作成に必要
path_alias	Flickr 内でグループを作る際に求められることがある
machine_tag	タグの拡張版. より詳細なタグ付けができる
original_format	元画像のフォーマット
original_dimension	元画像の大きさ.
ispublic	他のユーザーへの公開に関する ID
isfriend	友人への公開に関する ID
isfamily	家族への公開に関する ID
media	元画像の記録媒体
url_sq	75×75 サイズの画像ファイル URL
url_t	最大 100px の画像ファイル URL
url_s	最大 240px の画像ファイル URL
url_q	150×150 サイズの画像ファイル URL
url_m	最大 500px の画像ファイル URL
url_n	最大 320px の画像ファイル URL
url_z	最大 640px の画像ファイル URL
url_c	最大 800px の画像ファイル URL
url_l	最大 1024px の画像ファイル URL
url_o	オリジナルサイズの画像ファイル URL

2.1.2. Flickr API (2): flickr.photos.getFavorites

flickr.photos.getFavorites は、指定した ID に該当する画像の Favorite 数を取得する API である。ここで Flickr の Favorite とは、あるユーザーが Flickr 上の画像を閲覧しその画像を気に入った場合に、画像の上に表示される「Add to Faves」というボタンを押すことでつけることができるものであり、画像の Favorite 数とはその画像を気に入ったユーザーの数を表す、いわば画像の人気の変数というべきものである。この API について引数として指定すべき条件は表 4 の通りであり、結果として得られる情報は favorite 数である。今回は、結果として取得できる情報のうち画像が持つ Favorite 数のみを、flickr.photos.search を用いて得られる情報と共に画像のメタデータとして xml ファイルに保存する。また、この API を用いると、その画像を Favorite 登録した個々のユーザーのリストも結果として返され、こちらも今後システムを発展させていく際に有用な情報となる可能性がある。

2.1.3. Flickr API (3): flickr.photos.comments.getList

flickr.photos.comments.getList は、指定した ID に該当する画像に付与されたコメントのリストを取得するための API である。引数として指定すべき条件は表 5 の通りであり、結果として取得できる情報は画像に付けられたコメントのリストである。取得されたコメントは、flickr.photos.search, flickr.photos.getFavorites の結果として得られた情報と共にメタデータとして xml ファイルに保存する。

表 4 : flickr.photos.getFavorites に渡す引数

引数の名称	説明	今回指定した値
method	method の名前	flickr.photos.getFavorites
api_key	API Key	事前に取得した API Key
photo_id	Favorite 数を取得する画像の ID	flickr.photos.search で得た値
per_page	1 ページで取得する画像の枚数の上限	50
page	何ページ目を取得するか	1

表 5 : flickr.photos.comments.getList に渡す引数

引数の名称	説明	今回指定した値
method	method の名前	flickr.photos.comments.getList
api_key	API Key	事前に取得した API Key
photo_id	コメントを取得する画像の ID	flickr.photos.search で得た値

2.1.4. 3つのAPIを用いたFlickrからの画像とメタデータの取得

これまで説明した3つのAPIを用いて、Flickrから画像およびメタデータを取得する。全体の流れは以下の通りである。

(1) まず `flickr.photos.search` を用いて「指定された期間内に、指定した中心地点から半径〇kmの範囲内において撮影された画像」を検索する。

(2-1) `flickr.photos.search` の結果として得られた情報を基に画像のURLを形成して画像をダウンロードする

(2-2) `flickr.photos.search` の結果として得られた情報のうち、画像のIDを引数として

`flickr.photos.getFavorites` および `flickr.photos.comments.getList` を利用し、3つのAPIによって取得したメタデータを合わせて1つのxmlファイルに保存する。

今回は、この方法で国内の10都市で撮影された画像とそのメタデータを取得した。都市名と、`flickr.photos.search`の引数として渡した各都市の中心地点等をまとめたものを表6に示す。ここで1つ言及すべき点がある。本稿ではこれ以降、「各都市の中心地点から半径32kmの範囲内で撮影された画像」を例えば「東京で撮影された画像」「神奈川で撮影された画像」などと呼ぶことにするが、これらは「中心地点から32km離れた地点」が各都道府県の境界を越えていた場合、厳密には「隣の都道府県で撮影された画像」も含むことになる。しかし、我々は通常どこかに旅行し訪問するランドマークを決める際、特に県境などを意識することはなく、現在地や既に訪問すると決めたランドマーク等からの距離のみが問題となるため、収集した画像を厳密に都道府県で分ける必要はないと考えられる。したがって、便宜上「東京で撮影された画像」「神奈川で撮影された画像」という呼称を用い、そこには隣の都道府県で撮影された画像が含まれる可能性があるが、推薦の精度には特に影響はないといえる。

なお、第三章で述べる通り、本研究では現時点においては画像のメタデータのうち「ジオタグ」、「撮影日時」、「ユーザーID」の3つしか利用していない。しかし、今後システムを発展させたり他の研究と融合させたりする際に必要となる可能性を考え、入手できるメタデータは全て取得した。

さらに、海外の3つの観光都市についても、今回は海外の気象情報の入手が間に合わなかったため実験は断念したが、画像の取得は行った。その際の中心地点も表6に合わせて示す。一般的に、海外の都市の方が日本国内の都市よりも入手できる画像の数が多く、ビッグデータ活用の観点から見て興味深い結果が得られる可能性があるため、気象情報が入手できればこちらについても実験を行っていきたい。

表 6：画像収集における各都市の中心地点

都道府県名	緯度	経度	地点名
北海道	43.064307	141.346958	北海道庁
東京	35.689634	139.692101	東京都庁
千葉	35.632896	139.880394	東京ディズニーランド
神奈川	35.444159	139.638034	横浜市役所
静岡	35.096	139.07149	熱海市役所
大阪	34.693724	135.502212	大阪市役所
京都	35.009449	135.666773	嵐山エリア
奈良	34.685072	135.804848	奈良市役所
兵庫	34.690152	135.195523	神戸市役所
沖縄	26.212401	127.680932	沖縄県庁
ニューヨーク	40.758895	-73.985131	タイムズ・スクエア
パリ	48.856614	2.352222	パリ市庁舎付近
ロンドン	51.507351	-0.127758	ロンドン中心地

2.2. 気温と天気のデータの取得

本研究では、気温・天気を考慮した旅行ルート推薦システムの実現を目指すので、画像とメタデータばかりではなく気温と天気の情報も必要となる。本節では、この気温・天気データの取得について述べる。

2.2.1. 気象庁のサイトからの国内気象データの取得

国内の気温・天気データであれば、気象庁のサイトから極めて簡単な操作のみで取得することができる [29](図 7)。操作の手順は以下の通りである。

- (1) データを取得する地点を選択する（まず都道府県を選ぶと、選んだ都道府県内の全観測地点が表示されるので、その中からデータを取得したい地点を選択する）。
- (2) 取得するデータの項目と種類を選択する。このサイトでは、気温を始め風向・風速、蒸気圧、雲量など全 15 項目のデータを取得できるが、今回は「気温」「降水量」「日照時間」の 3 項目のみを選択した。また、データの種類は特別値、日別値、月別値等 6 つの中から選択が可能で、今回は特別値を選択した。
- (3) データを取得する期間を選択する。前節で取得した画像とメタデータに対応する期間（凡そ 2004 年 1 月 1 日～2016 年の途中頃まで）のデータを取得したいが、このサイトで



図 7：気象庁の過去の気象情報データベース

は 1 度にダウンロードできるデータの量に上限（地点数、項目数、期間、データの取得・計算等にかかる負荷を考慮して決まる）があるため、ここでは各年の「1 月～4 月」「5 月～8 月」「9 月～12 月」として、4 か月分ずつデータを取得する。なお、このような 4 か月分のデータで、ダウンロード上限量の約 80%程度になる。

(4) 表示オプションを選択する。例えば「データを数値のみで格納するか、それとも数値以外の記号等も含むか」など、全 4 項目のオプションについて選択を行う。

以上の 4 つのステップで条件を指定していくと、所望のデータをダウンロードすることができる。ダウンロードファイルは csv 形式となっており、データの利用も容易である。

ここで注意点がある。それは、気象庁からダウンロードしたデータには欠損している箇所があるということである。確認したところ、「日照時間」に関しては欠損箇所が非常に多かったため、本研究では利用を断念した。具体的には、当初は「降水量」と「日照時間」を合わせて天気を「晴れ」「曇り」「雨または雪」の 3 つに分ける心積もりであったが、今回は「降水量」のみを用いて「雨または雪」と「それ以外（晴れまたは曇り）」の 2 分類のみとすることにした。

一方、「気温」と「降水量」に関しては欠損が少なかったため、欠損箇所を適宜補って利用することとした。データの補完の方法は、次の通りである（図 8 参照）。

(1) データが欠損しているセル（セル A）があったら、セル A より上のセルを 1 つずつ見ていき、データがあるセル（セル B）を見つけるまで続ける。セル A より下のセルについても同様に、データが格納されているセル（セル C）を見つけるまで 1 つずつ下に下がっていく。

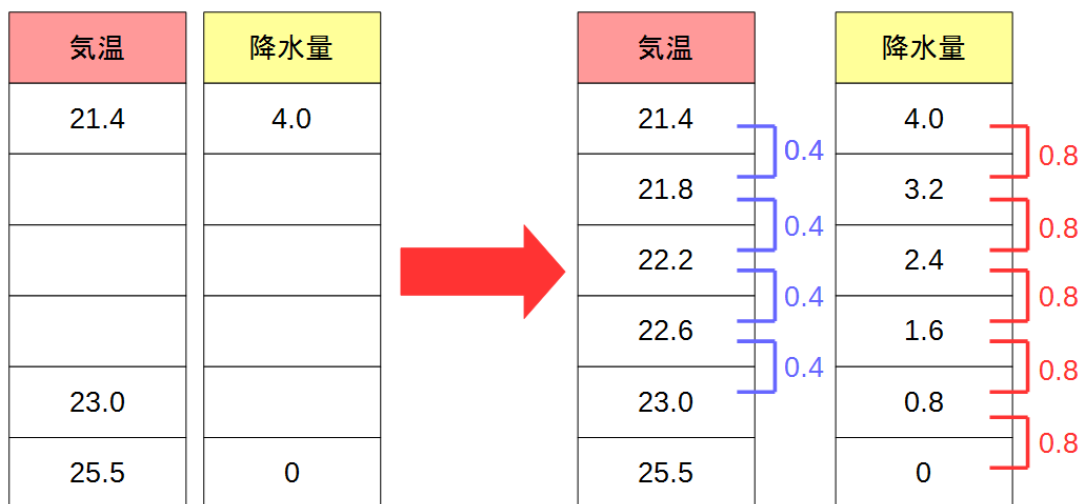


図 8 : 欠損している気象データの補完方法

(2) (1)で見つけたセル B とセル C について、

$$(\text{B と C の気温 (降水量) の差}) / (\text{B と C の時間の差})$$

として、時間を x 軸、気温または降水量を y 軸とした場合の直線の傾きを得る。

(3) (2)で得た傾きを基に、空白のセルを埋めていく。

以上の方法で気温および降水量のデータの欠損箇所を補完していく。無論この方法で補った値は正確なものではない。特に気温に関しては、例えば「10 時～18 時」のように「最初は気温が上昇していき、(12～15 時頃に) ピークに達した後に今度は下降していく」ような時間帯においてデータが欠損していた場合には、気温の上下する過程がうまく再現されなくなってしまう。しかし、上述のように気温と降水量についてはデータの欠損箇所が非常に少なかったため、この補完操作が旅行ルートのおすすめ精度に与える影響は十分に小さいと考えられるので、この方法を採用する。

2.2.2. 海外の気象情報について

本研究室で調査したが、気象庁のサイトで容易に入手できる国内の気象データと違い、海外の気象データは個人で入手するのは難しい。そこで今回は海外の会社から気象データを購入することにしたが、諸般の事情によりデータの到着が遅れてしまったため、本稿では実験の対象とするのは国内都市のみとした。

第三章 提案手法

この章では、提案手法である「気温・天気を考慮した旅行ルート推薦システム」について解説する。提案手法のフローチャートは図9のようにになっている。以下でそれぞれのステップを詳しく説明していく。

3.1. ステップ(1) : データの収集

前章の通り、Flickr から Flickr 自身が公開する API を用いて大規模画像およびそのメタデータを収集する。すると、位置情報を示すジオタグを始めとしたメタデータ付きの画像を多数取得することができる (図 10, 図 11)。また、気象庁のサイトから過去の気象データをダウンロードする。

3.2. ステップ(2-1) : 位置情報の解析

ここではまず、mean-shift というクラスタリング手法を用いて、大量のジオタグから人気のあるランドマークを抽出する。その後 Flickr からダウンロードした画像を抽出したランドマークに割り当てることにより、各画像に「場所の ID」を付ける。

まず基礎知識として、mean-shift というのは点群があった時にその「密度の局所的な極大点」を求めるクラスタリング手法である。このクラスタリング手法を、画像の撮影地点を表すジオタグの緯度と経度に対して用いると、「写真が頻繁に撮影された場所」すなわち「人がたくさん訪れた人気のある場所」を検出することができる (図 12)。本稿ではこれ以降、この検出された人気のある場所を「ランドマーク」と呼ぶ。通常どこかに旅行に行く際、その場所を何度も訪れ人気のある場所には行き尽くした人 (そしてそのような人は、旅行推薦システムによる手助けをあまり必要としないと推測される) でもない限り、あまり人気のない場所よりも人気のある観光名所を訪れようとすると考えられる。したがって、提案する旅行推薦システムではこれらのランドマークの中から訪問場所を提案することとする。

具体的な手順としては、まずステップ(1)で収集した大規模画像のメタデータについて、東京や千葉などの都市ごとに、都市内で撮影された全ての画像のジオタグ (緯度と経度) のみを抜き出して予め1つのファイルにまとめておく。この1つ1つの緯度と経度のデータが、mean-shift 法の対象とする点群の1つ1つの点に対応する。なお、前章で述べたように取得した画像の中にはジオタグが付いていないものも含まれてしまっていることがあるため、そのような画像のデータについてはこの処理の際に取り除いておく。次に、このジオタグをまとめたファイルに対して mean-shift 法を用いるが、これは python の機械学習

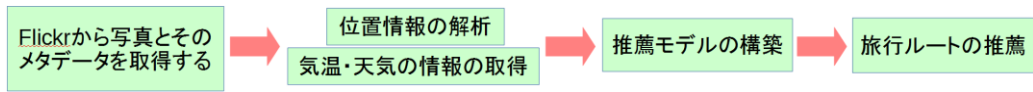


図 9 : 提案手法のフローチャート

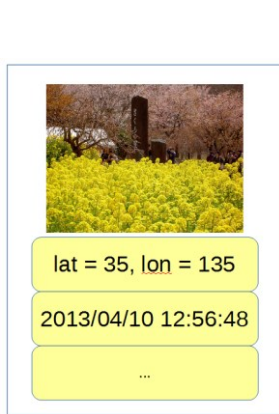


図 10 : Flickr から入手する
メタデータ付き画像

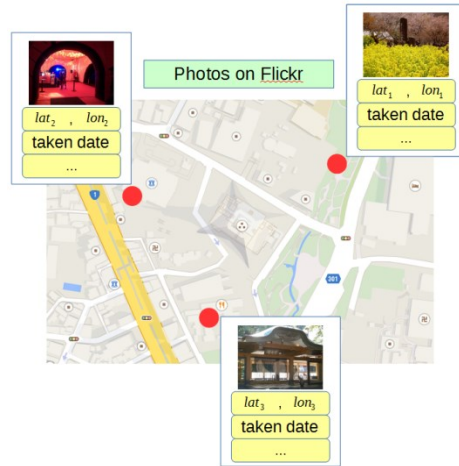


図 11 : Flickr 上の位置情報付き画像群

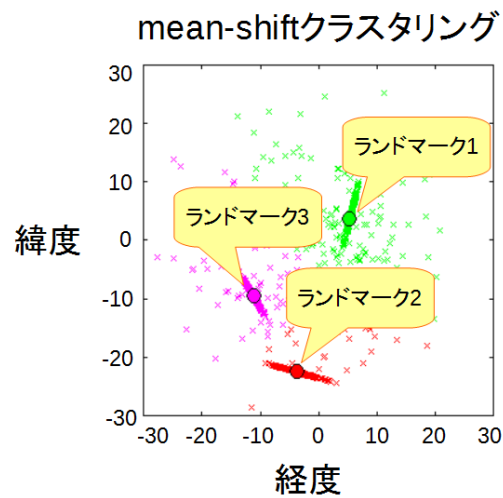


図 12 : mean-shift クラスタリング手法に基づく
ランドマークの抽出

ライブラリ `scikit-learn` に含まれる `MeanShift` モジュールをインポートすることで簡単に実現できる. `mean-shift` 法では, 事前にクラスタ数を指定する必要がなく, 設定したバンド幅によって自動的にクラスタ数が決まる. 今回は 2 パターンのバンド幅を用い, 検出さ

表 7：東京で撮影された画像から抽出されたランドマーク人気上位 10 個

順位	緯度	経度	場所の名前
1	35.66301	139.70311	渋谷駅周辺
2	35.67638	139.76458	東京駅周辺
3	35.69111	139.69985	新宿駅周辺
4	35.66290	139.73051	六本木駅周辺
5	35.70043	139.77152	秋葉原駅周辺
6	35.65840	139.74742	東京タワー周辺
7	35.69942	139.75406	東京ドーム周辺
8	35.71243	139.79636	浅草寺周辺
9	35.71428	139.77294	上野公園
10	35.63204	139.73669	品川駅周辺

れるクラスタ数, すなわちランドマーク数を変化させた. 次章では, それぞれのパターンについて実験を行う. 一例として, 東京で撮影された枚の画像に対してバンド幅を 0.01 として mean-shift 法を用いると, 604 個のランドマークが検出され, その人気上位 10 個に対して該当する場所を地図で調べると, 表 7 の通りであった. 表 7 を見ると, 渋谷駅周辺や東京タワー等が挙げられており, 確かに多くの人を訪れる場所がランドマークとして検出されているといえる.

上記の手順に従いランドマークが検出できたら, 次は 1 枚 1 枚の画像を検出されたランドマークに割り当てていく. ここでは, 各画像のジオタグを参照し, 検出された全ランドマークとの距離を計算し, 最も距離が近いランドマークに割り当てるという方法をとる. 地球上の 2 地点間の距離を緯度と経度から計算する式は複数存在するが, 今回はそのうちの 1 つを採用した. その式によると, 地点 A (緯度 x_1 , 経度 y_1) と地点 B (緯度 x_2 , 経度 y_2) の間の距離 d は, 以下のように表すことができる.

$$d = r \cos^{-1}(\sin x_1 \sin x_2 + \cos x_1 \cos x_2 \cos \Delta y)$$

$$\Delta y = y_2 - y_1$$

以上のように, このステップでの処理の結果, 東京や千葉などの各都市ごとにランドマークを抽出し, さらに各都市で撮影された画像にランドマークの ID を割り当てることができた.

3.3. ステップ(2-2)：その他付加情報のラベルの割り当て

本研究で目指すのは, 「気温・天気を考慮した旅行ルート推薦」であるため, その準備として, 考慮する要素に関するラベルを各画像に与える. 本研究の主たる要素である「気温」

と「天気」のラベルの他、関連研究 [5] を参考に「季節」と「時間帯」のラベルも与える。

3.3.1 気温・天気ラベルの割り当て

画像のメタデータと気温・天気のデータを照らし合わせて、画像に気温や天気のラベルを割り当てる (図 13)。

ここで用いる画像のメタデータは、「撮影日時」である。まず画像の撮影日時を参照し、気象情報を格納した csv ファイル中で該当する日時のデータを読み出す。前章で述べた通り、取得した気象情報のうち「日照時間」についてはデータの欠損が多いため利用を断念し、今回は「気温」と「降水量」のみを利用して、画像に「気温」と「天気」のラベルを付与する。それぞれのラベル付けのポリシーは以下の通りである。

- ・気温 (1) : 気象情報「気温」を参照し、その値に応じて以下の 4 つのパターンに分類する。
 - (1) 気温帯 1 : 気温 10°C 未満
 - (2) 気温帯 2 : 気温 10°C 以上 20°C 未満
 - (3) 気温帯 3 : 気温 20°C 以上 30°C 未満
 - (4) 気温帯 4 : 気温 30°C 以上
- ・気温 (2) : 気象情報「気温」を参照し、その値に応じて以下の 7 つのパターンに分類する。
 - (1) 気温帯 1 : 気温 5°C 未満
 - (2) 気温帯 2 : 気温 5°C 以上 10°C 未満
 - (3) 気温帯 3 : 気温 10°C 以上 15°C 未満
 - (4) 気温帯 4 : 気温 15°C 以上 20°C 未満
 - (5) 気温帯 5 : 気温 20°C 以上 25°C 未満
 - (6) 気温帯 6 : 気温 25°C 以上 30°C 未満
 - (7) 気温帯 7 : 気温 30°C 以上
- ・天気 : 気象情報「降水量」を参照し、その値に応じて以下の 2 つのパターンに分類する。
 - (1) 晴れまたは曇り : 降水量 0
 - (2) 雨 : 降水量が 0 より大きい

3.3.2 季節・時間帯ラベルの割り当て

本研究では、提案手法の肝である「気温」「天気」の他、関連研究 [5] を参考に「季節」と「時間帯」の 2 要素についても考慮することとした。そのため、この 2 要素に関するラベル付けも必要となる (図 14)。

「季節」「時間帯」に関するラベル付けは、画像の「撮影日時」のみで可能であり、その他のメタデータや気象情報等は必要としない。今回は、以下のポリシーに則ってそれぞれのラベル付けを行った。

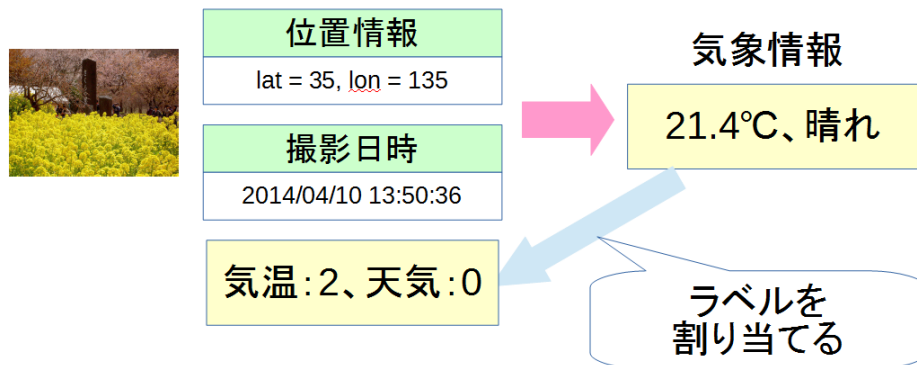


図 13 : 気温・天気ラベルの割り当て



図 14 : 季節・時間帯ラベルの割り当て

- ・季節：画像の「撮影月」に着目し，以下の4つのパターンに分類する．
 - －(1) 春：3月～5月
 - －(2) 夏：6月～8月
 - －(3) 秋：9月～11月
 - －(4) 冬：12月～2月
- ・時間帯：画像の「撮影時間」に着目し，以下の4つのパターンに分類する．
 - －(1) 時間帯1：0時～5時59分59秒
 - －(2) 時間帯2：6時～11時59分59秒
 - －(3) 時間帯3；12時～17時59分59秒
 - －(4) 時間帯4：18時～23時59分59秒

3.4. ステップ(3)：推薦システムの実現

この節では，本稿が提案する「気温・天気を考慮した旅行ルート推薦」システムについて解説する．まず，システムの基礎となる推薦モデルを導き出し，次にその推薦モデルを用いる際の実際の処理の流れを説明する．

3.4.1. 推薦モデルの構築

本研究の根幹をなす推薦モデルの構築について説明する。提案手法である「気温・天気を考慮した旅行ルート推薦」を実現するため、ここでは関連研究 [5][26] を参考にベイズの定理を用いてマルコフモデルに気温と天気の情報を組み込むということをする。

まず、気温 t と天気 w を合わせて $S_u \in (t, w)$ と表すこととする。すると、現在ランドマーク L_i にいる人物が、気温・天気が S_u の条件で次にランドマーク L_j を訪問する確率は、以下の式で表すことができる。

$$P(L_{i \rightarrow j} | S_u) = \frac{P(L_{i \rightarrow j}, S_u)}{P(S_u)} = \frac{P(L_{i \rightarrow j})P(S_u | L_{i \rightarrow j})}{P(S_u)}$$

ここで、直感的に $P(L_{i \rightarrow j})$ は $P(L_i)P(L_j | L_i)$ と等しいことが分かるので、

$$P(L_{i \rightarrow j} | S_u) = \frac{P(L_i)P(L_j | L_i)P(S_u | L_{i \rightarrow j})}{P(S_u)}$$

結局、次の目的地として最適なランドマークは、この式を最大化するようなランドマーク L_j であるので、

$$L^* = \arg \max_{L_j} P(L_j | L_i) P(S_u | L_{i \rightarrow j})$$

気温 t と天気 w の間に独立性を仮定すると、この式はさらに以下のように変形することができる。これが推薦モデルを表す式となる。

$$L^* = \arg \max_{L_j} P(L_j | L_i) P(t | L_{i \rightarrow j}) P(w | L_{i \rightarrow j})$$

この式から分かるように、本研究の推薦モデルは非常に単純な式で表すことができる。なお、式中の $P(t \text{ or } w | L_{i \rightarrow j})$ および $P(L_j | L_i)$ は、以下のように簡単に計算することができる。

$$P(t \text{ or } w | L_{i \rightarrow j}) = \frac{\text{count}(L_{i \rightarrow j} \cap S_u = t \text{ or } w)}{\text{count}(L_{i \rightarrow j})}$$

$$P(L_j | L_i) = \frac{\text{count}(L_{i \rightarrow j})}{\sum_{L_j \in L} \text{count}(L_{i \rightarrow j})}$$

ここで、 L は全ランドマークの集合であり、 $\text{count}(L_{i \rightarrow j})$ は L_i から出発し L_j に到着するデータの総数である。同様に $\text{count}(L_{i \rightarrow j} \cap S_u = t \text{ or } w)$ は、ある気温 t もしくはある天気 w の時に L_i から L_j に行くデータの総数である。

さらに、先述の通り本研究では関連研究 [5] を参考に「季節」と「時間帯」についても追加で考慮することとしたので、季節 s と時間帯 $time$ についても同様に考えると、本研究の最終的な推薦モデルは次の式で表される。

$$L^* = \arg \max_{L_j} P(L_j | L_i) P(t | L_{i \rightarrow j}) P(w | L_{i \rightarrow j}) P(s | L_{i \rightarrow j}) P(time | L_{i \rightarrow j})$$

$$P(s \text{ or } time | L_{i \rightarrow j}) = \frac{\text{count}(L_{i \rightarrow j} \cap S_u = s \text{ or } time)}{\text{count}(L_{i \rightarrow j})}$$

最後に、気温 t と天気 w の間の独立性について言及しておく。我々が「晴れ」と聞くと「暖かそう」、「雨」と聞けば「肌寒そう」などとイメージすることからも想像できる通り、気温と天気は厳密に言えば独立ではない。しかし、本研究のような確率モデルを扱う場合においては、独立性を仮定しても特に顕著な悪影響などはなく、うまく動作することが経験的に知られている。そこで本研究でも気温と天気の間独立性を仮定し、推薦モデルを簡潔化した。

3.4.2. 推薦モデルに基づく実際の処理の流れ

前項で導出した推薦モデルを用いて実際に旅行ルートを推薦する場合の、具体的な処理の流れを説明する。

まず前提として、本研究が目指す旅行ルート推薦は、以下のように定義される。

「旅行ルート推薦の定義：旅行者が現在ランドマーク L_i にいて、気温・天気等の各条件が $S_u = (t, w, s, time)$ の場合に、旅行者に次に訪問すべきランドマーク L_j を提示すること」

この問題設定だと、ただ1つのランドマークを提示しているだけであり、旅行「ルート」とはいえないのではないかという疑問を抱かれるかもしれない。しかし、このような「 $L_i \rightarrow L_j$ 」という2つのランドマーク間の移動をつなげていけば1つのルートを生成できるため、必要に応じて「ルート」の生成・提案をすることも可能である。ただ、本稿での問題設定としては、上記の通りとする。なお、旅行の規模については、「1つの都市内で1日でなされたもの」とする。「都市」というのは、画像収集の際に収集の単位とした東京、京都などを指す。また、1日の旅行に焦点をあてているため、日付を跨いでランドマーク間の移動があったとしても、それについては考慮しない。

処理の流れの説明に戻る。まず全体を箇条書きで簡潔にまとめると、以下のような流れになる。

- ・ランドマーク間の移動を検出し、移動の履歴を作る。
- ・ランドマーク間の移動に関する統計をとり、行列を作る。
- ・作成した行列を推薦モデルに当てはめ、実際に旅行ルートの推薦を行う。

各過程を1つずつ説明していく。

- ・ランドマーク間の移動の履歴の作成

前のステップまでの処理により、各画像はランドマークのIDと気温、天気、季節、時間帯それぞれのラベルを持っている。それらの画像をまず、メタデータの1つである「ユーザーID」によって並べ替え、ユーザーごとの画像の集合を作る。次に、ランドマーク間の移動を検出するには、各ユーザーが少なくとも2枚は画像を持っていないといけないため、1枚しか画像をアップロードしていないユーザーは取り除き、2枚以上アップロードしているユーザーの画像集合のみを残す。そして、各ユーザーの画像を、画像集合内で時間順になるように並び替え、本研究では「1日の旅行」を対象とするので、さらに日付ごとに分割する。

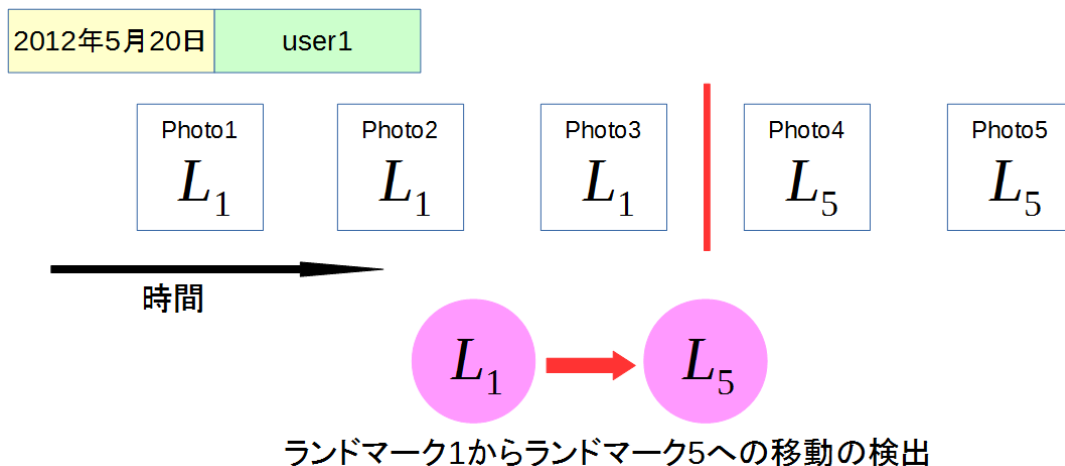


図 15 : ランドマーク間の移動の検出

以上のように画像集合を分割したら、次にランドマーク間の移動の検出をする。ある集合内の時間順に並べられた画像のランドマークの ID を見ていき、例えばランドマーク L_1 で撮影された画像の次にランドマーク L_5 で撮影された画像があれば、「 $L_1 \rightarrow L_5$ 」の移動が検出されるというのが基本操作になる。同一のランドマークで撮影された画像が複数枚続いている場合、それらは 1 まとまりと見做す (図 15)。

・ランドマーク間の移動についての統計の作成

ここでは、「ランドマーク数×ランドマーク数」の大きさの行列を考える。この行列の行は「現在いるランドマーク L_i 」に、列は「次に訪問するランドマーク L_j 」に、それぞれ対応している。そして、例えば先ほども例に挙げた「 $L_1 \rightarrow L_2$ 」の移動が検出された場合には、その行列の「1 行 5 列目」に格納されている値に 1 を足す。要するにこれはランドマーク間の遷移を表現する行列であり、対角成分は 0 となる。見方を変えると、この行列は「現在いるランドマーク」と「次に訪問するランドマーク」の関係性を表しているともいえる。つまり、「次の状態は現在の状態に依存する」とするマルコフモデルに関連し、推薦モデルを表す式の第一項に対応する行列となっている。

この他にも行列を複数作っていく。これらの行列のサイズも全て「ランドマーク数×ランドマーク数」とする。そして、例えば 2 つの行列を用意して、ランドマーク間の移動「 $L_1 \rightarrow L_5$ 」があった時、その時点での天気ラベルを参照して「晴れまたは曇り」であれば行列 1 の (1, 5) 成分に、天気が「雨」であれば行列 2 の (1, 5) 成分に 1 を足す。つまりこれら 2 つの行列は「天気を考慮した際のランドマーク間の遷移」を表しており、推薦モデルの式の第三項に対応する。これらの行列についても対角成分は 0 であり、また、「(晴れまたは曇りの時の $L_m \rightarrow L_n$ の移動の数) + (雨の時の $L_m \rightarrow L_n$ の移動の数) = ($L_m \rightarrow L_n$ の移動の総数)」であるので、2 つの行列の (m, n) 成分の和は、上記の第一項に対応する行列の (m, n) 成分と等しくなる (図 16)。

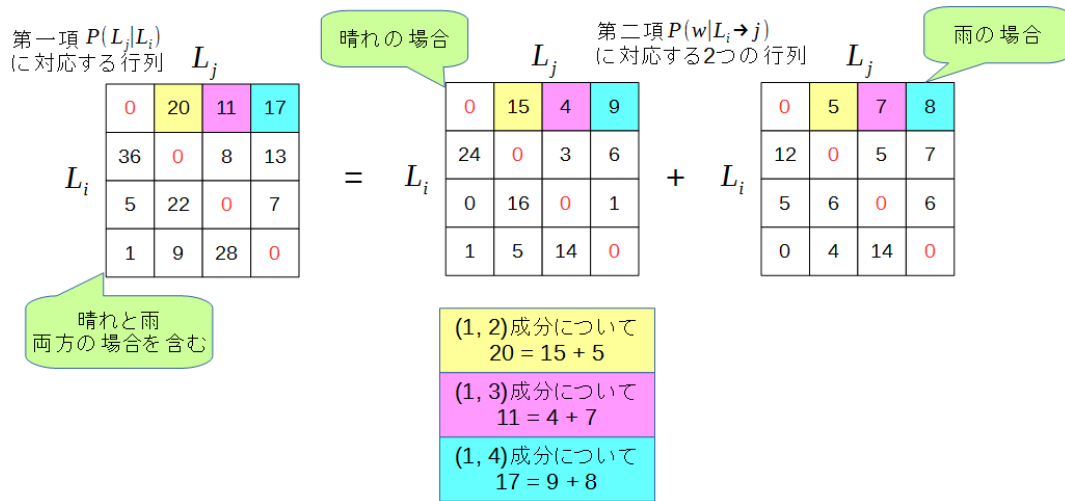


図 16 : 行列同士の対応関係

天気他に気温、季節、時間帯に関して、それぞれラベルの数と同数の行列を用意し、ランドマーク間の遷移に応じた成分に値を足していくことで、推薦モデルの式の第二項、第四項、第五項に対応する行列の組を作ることができる。なお、ここで参照する「移動の時点での気温、天気、季節、時間帯のラベル」とは厳密には、「現在いるランドマーク（先ほどの例では L_1 ）において最後に撮影された画像に付与された各ラベル」を指している。

・推薦モデルと行列を用いた旅行ルート推薦

本稿の定義する旅行ルート推薦とは上述の通り、「旅行者が現在ランドマーク L_i にいて、気温・天気等の各条件が $S_u = (t, w, s, time)$ の場合に、旅行者に次に訪問すべきランドマーク L_j を提示すること」である。したがって、推薦を行う際の入力は $(L_i, t, w, s, time)$ の組であり、出力すなわち推薦結果は L_j となる。

推薦の実際の処理の様子は以下ようになる。まず、入力の組 $(L_i, t, w, s, time)$ が与えられる。現在いるランドマークは L_i なので、先ほど求めたマルコフモデルに対応する行列の i 行目を取り出す。気温は t であるので、気温モデルに対応する4つの行列のうち、ラベル t に応じた行列の i 行目を取り出す。天気、季節、時間帯についても同様に、各ラベルに対応する行列の i 行目を取り出してくる。そして、まず各行列の「 i 行1列目」の要素に着目して、推薦モデルを表す式によって次にランドマーク L_j を訪問する確率を求める。次に各行列の「 i 行2列目」の要素について同様の計算を行い、次にランドマーク L_2 を訪問する確率を求める。以降 i 行目の全ての要素について同様に計算していき、求めた確率が最大となった L_j を次に訪問すべきランドマークとして旅行者に提示する。

以上が、構築した推薦モデルを使い実際に旅行ルート推薦を行う際の処理の流れである。



図 17：気温・天気を考慮した旅行ルート推薦

3.5. ステップ(4)：旅行ルートの推薦

これまで述べてきた推薦モデル・推薦手法に則り，実際にユーザーに旅行ルートを推薦する．気温・天気・季節・時間帯の4つの要素を考慮した本研究の推薦モデルを用いると，例えば 図 17 のように「現在地が上野駅で晴れていて暖かい日ならば，次の目的地として上野動物園を推薦する．しかし，現在地が同じ上野駅であっても雨が降っていて肌寒い日ならば，次の目的地には美術館を推薦する」といった旅行ルート推薦が実現できる．

第四章 実験

この章では、実験について述べる。まず、実験方法や推薦モデルとの比較対象となるベースラインの手法等の実験の設定について解説する。続いて Flickr から収集した画像およびメタデータから成るデータセットの詳細を説明する。そして最後に実験結果を示し、考察を行う。

4.1. 実験の設定

4.1.1 実験方法

まず、実験方法を説明する。前章までに解説した手順により、我々の手元にはランドマーク間の遷移とその遷移が起こった時点での気温・天気・季節・時間帯の条件の組、すなわち $\text{movement} = (L_i, L_j, t, w, s, \text{time})$ が大量にある。ここで、 L_i は現在地のランドマーク、 L_j は目的地のランドマーク、 t は気温のラベル、 w は天気のラベル、 s は季節のラベル、 time は時間帯のラベルをそれぞれ示す。実験では、これらの 6 つのデータのうち、目的地のランドマーク L_j のみを敢えて隠し、残りの 5 つのデータを入力として「現在ランドマーク L_i にいるユーザーが条件 (t, w, s, time) の場合に次に訪問するランドマークはどこであるか」ということを推薦モデルを用いて推測し、推測結果を出力する。そしてその推測結果が実際に訪問したランドマーク L_j と一致しているかどうかを見る (図 18)。一致していれば正解、一致していなければ不正解とみなし、正解率つまり「(正解した数) / (正解した数) + (不正解の数)」を推薦精度として、この値をベースラインの手法と比較する。無論、この推薦精度は高いほど良い。

この種の実験では **cross-validation** (データセットの一部をテスト用、残りをトレーニング用として、トレーニングデータを用いて構築したモデルを使ってテスト用データに対して実験を行う実験手法) を使用するのが望ましい。今回はその中でも、データセットのうちの 1 つをテスト用、残り全てをトレーニング用とし、全てのデータがテストデータとなるようにデータ数と同じ回数だけトレーニングと実験を行う **leave-one-out** という実験手法を用いた。**cross-validation** には他にも、データセットを k 個に分割し、そのうちの 1 つをテスト用、残りをトレーニング用とする **k-fold cross-validation** と呼ばれるものもある。モデルのトレーニングが k 回で済む **k-fold cross-validation** に比べ、データ数と同じ回数のトレーニングを行わなければならない **leave-one-out** 法は計算コストが高いが、その分実験の正確性も高くなる。

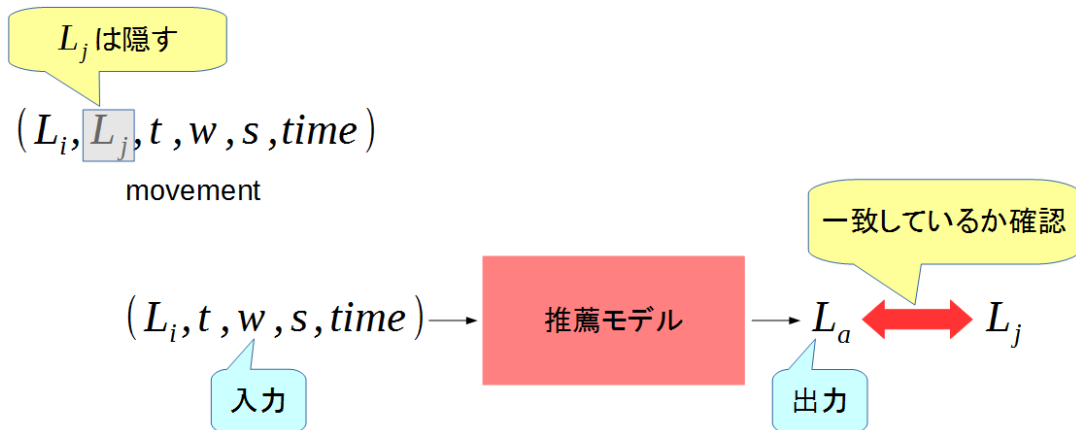


図 18 : 実験の流れ

4.1.2. ベースライン

繰り返しになるが、本研究の提案手法は「気温・天気を考慮した旅行ルート推薦システム」であるが、今回は関連研究 [5] を参考に、季節と時間帯についても考慮している。また、全ての要素を考慮する手法のみだけではなく、個々の要素だけを考慮した場合についても実験することとした。したがって、今回実験により推薦精度の比較を行う手法は、以下の 9 通りとなる。

- (1) 人気順モデル：単純にランドマークの人気順に推薦する手法
- (2) マルコフモデル：マルコフモデルすなわち現在地のみに基づいた手法
- (3) 気温モデル：現在地に加え、気温のみを考慮した手法
- (4) 天気モデル：現在地に加え、天気のみを考慮した手法
- (5) 気温・天気モデル：現在地に加え、気温・天気の 2 つを考慮した手法
- (6) 季節モデル：現在地に加え、季節のみを考慮した手法
- (7) 時間帯モデル：現在地に加え、時間帯のみを考慮した手法
- (8) 季節・時間帯モデル：現在地に加え、季節・時間帯の 2 つを考慮した手法
- (9) 気温・天気・季節・時間帯モデル：現在地に加え、気温・天気・季節・時間帯を全て考慮した手法

これら 9 つの手法のうち、(1) ランドマークの人気順に推薦する手法と (2) マルコフモデルにのみ基づく手法の 2 つがベースラインであり、残りは全て推薦手法またはそれに準ずる手法となる。

ここで、前章で述べた通り、本研究の推薦モデルは以下の式で表される。

$$L^* = \arg \max_{L_j} P(L_j|L_i)P(t|L_{i \rightarrow j})P(w|L_{i \rightarrow j})P(s|L_{i \rightarrow j})P(time|L_{i \rightarrow j})$$

この式と上記の手法 (2)~(9) の対応関係は、それぞれ以下の通りとなる。

- (2) 式中の第一項にのみ対応
- (3) 式中の第一項に加え第二項に対応
- (4) 式中の第一項に加え第三項に対応
- (5) 式中の第一項に加え第二項・第三項に対応
- (6) 式中の第一項に加え第四項に対応
- (7) 式中の第一項に加え第五項に対応
- (8) 式中の第一項に加え第四・第五項に対応
- (9) 式全体に対応

4.1.3. その他の設定

本実験では、mean-shift クラスタリング法によって抽出された全てのランドマークを考慮する基本的な実験の他、各都市で抽出されたランドマークのうち、人気上位 25 位・上位 10 位までのランドマークのみを考慮した場合の実験も行う。前章で解説した通り、本研究では、2つのランドマーク間の移動「 $L_i \rightarrow L_j$ 」を旅行の1つの単位としており、この遷移に関する統計をランドマーク数×ランドマーク数の大きさの行列で保持することとしている。上位 25 位、上位 10 位までのランドマークのみを考慮することは、この行列の大きさを 25×25 、 10×10 にまで縮小することと同義であり、問題の複雑性をある程度緩和することとなる (図 19)。

これを実現するための具体的な手順を、上位 25 位以内のランドマークを考慮する場合を例として説明する (図 20)。図において、赤い円は上位 25 位以内に含まれるランドマーク、青い三角形は上位 25 位以内に含まれないランドマークを表すとする。第三章で述べた処理により、今、全ランドマークを対象としてランドマーク間の移動「 $L_i \rightarrow L_j$ 」が記録されている。図より、「25位以内のランドマーク $L_1 \rightarrow$ 25位外のランドマーク L_2 」という movement1 と、「ランドマーク $L_2 \rightarrow$ 25位以内のランドマーク L_3 」という movement2 が続けて記録されている。この時、2つの movement のそれぞれに紐付けられたユーザーIDと日付を確認し、両者の間でそれらが一致していればランドマーク上位 25 位外のランドマークである L_2 を無視した新たな movement 「 $L_1 \rightarrow L_3$ 」が生成される。これは、ランドマーク L_1 と L_2 の間に訪れた L_3 を「寄り道」とみなすイメージである。この処理によって上位 25 位以内に含まれないランドマークを除外した新たなランドマーク間の移動のデータセットが得られる。それらについて再び気温・天気・季節・時間帯の各条件下における統計を取り、 25×25 の大きさの行列を作成する。後は全てのランドマークを考慮する場合と同様に、提案する推薦モデルに基づいて現在地・気温・天気・季節・時間帯の入力から次に訪問するランドマークを推測すればよい。上位 10 位以内のランドマークのみを考慮する場合も、同様の処理を行う。

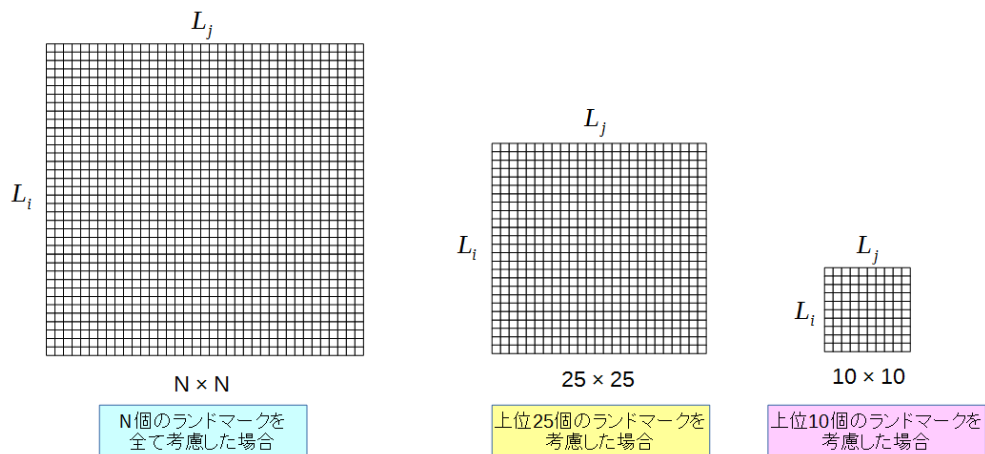


図 19 : 考慮するランドマーク数による行列の大きさの違い

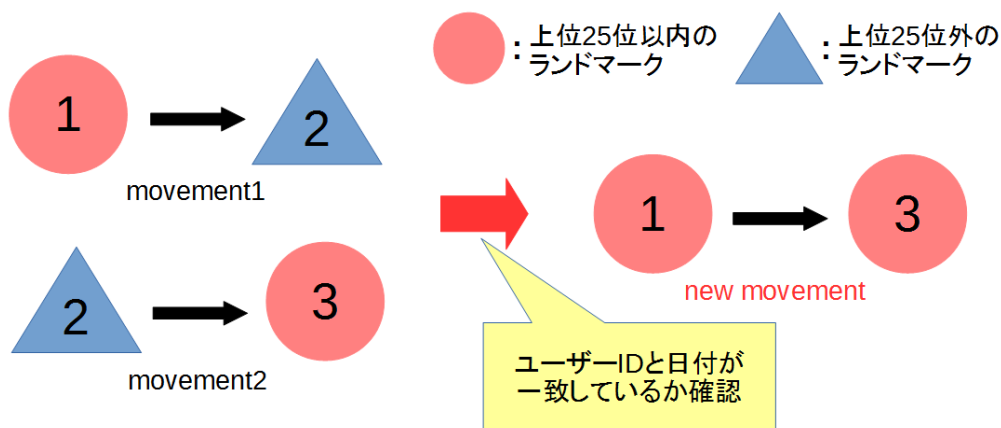


図 20 : 上位 25 位以内のランドマークを考慮する場合の新たな移動の生成

4.2. 実験に用いるデータセット

本研究では、Flickr から Flickr API を用いて取得したデータセットに対し、実験を行う。2.1. で述べた国内 10 都市において収集されたデータの概要は表 8 の通りである。なお、データの収集開始日についてはどの都市においても 2004 年 1 月 1 日としている。これは Flickr がサービスを開始したのが 2004 年のためである。表から分かるように、収集されたデータは 10 都市で合計 7,256,628 枚となった。第一章で述べた通り、SNS サイトの人気の上昇に後押しされ、近年旅行ルート推薦の研究が盛んになっているが、多くの研究はニューヨークやロンドン、パリなど日本以外の国の観光都市に焦点を当てているものが多く、日本の観光都市が取り上げられているものは現在のところほとんどない。そのためこれほどの規模の国内画像のデータセットは我々が知る限りでは存在しておらず、その意味で今回収集した画像とメタデータは純粋にデータセットとしての有用性も高いといえる。また、

表 8 : 国内 10 都市で収集された画像の概要

都市名	収集した画像の枚数	収集期間
北海道	64,755	～2015 年 12 月 31 日
東京	2,338,984	～2016 年 7 月 27 日
千葉	2,101,768	～2016 年 4 月 18 日
神奈川	755,644	～2012 年 5 月 4 日
静岡	99,307	～2016 年 12 月 15 日
大阪	332,719	～2015 年 12 月 31 日
京都	465,101	～2015 年 12 月 31 日
奈良	321,679	～2016 年 12 月 15 日
兵庫	312,287	～2016 年 12 月 15 日
沖縄	464,384	～2015 年 12 月 31 日

表 9 : 海外 3 都市で収集された画像の概要

都市名	収集した画像の枚数	収集期間
ニューヨーク	453,066	～2007 年 10 月 16 日
パリ	439,125	～2008 年 10 月 20 日
ロンドン	550,538	～2007 年 11 月 13 日

本研究で利用したのは画像のメタデータのみだが、Flickr はクリエイティブ・コモンズ・ライセンスを採用しているため、ライセンスの ID さえ確認すれば画像も公開・再利用することができる。

また、今回は気象データの入手が間に合わず実験をすることはできなかったが、海外 3 都市についても画像の収集を行った。その概要は表 9 の通りである。

4.3. 実験条件と実験結果

4.3.1 実験の条件

本稿では、数パターンの条件下で実験を行った。変化させた条件は以下の通り。

- ・ mean-shift クラスタリング法のバンド幅
 - パターン 1 : 0.01
 - パターン 2 : 0.025
- ・ 気温・天気のリベル付け
 - パターン 1 : 気温「10℃未満」「10℃～20℃未満」「20℃～30℃未満」「30℃以上」の 4 ラベル, 天気は「晴れまたは曇り」「雨」の 2 ラベル

表 10 : mean-shift クラスタリング法のバンド幅を 0.01 とした時の移動の検出

都市名	ランドマーク数	移動の総数	旅行したユーザー数
北海道	313	4,067	790
東京	604	135,975	10,451
千葉	572	121,341	9,947
神奈川	471	46,481	4,690
静岡	351	6,680	1,366
大阪	574	18,319	3,057
京都	479	29,207	4,295
奈良	527	16,314	3,187
兵庫	502	19,441	3,167
沖縄	217	4,695	703

表 11 : mean-shift クラスタリング法のバンド幅を 0.025 とした時の移動の検出

都市名	ランドマーク数	移動の総数	旅行したユーザー数
北海道	83	2,381	645
東京	95	84,739	8,833
千葉	91	73,295	8,358
神奈川	96	28,042	3,657
静岡	80	4,895	1,243
大阪	100	11,102	2,396
京都	110	15,512	3,429
奈良	107	8,534	2,273
兵庫	100	10,971	2,340
沖縄	72	3,759	672

ーパターン 2 : 気温「5℃未満」「5℃～10℃未満」「10℃～15℃未満」「15℃～20℃未満」「20℃～25℃未満」「25℃～30℃未満」「30℃以上」の 7 ラベル, 天気は「晴れまたは曇り」「雨」の 2 ラベル

・季節・時間帯のラベル付け

ーパターン 1 : 季節は「春 (3～5 月)」「夏 (6～8 月)」「秋 (9～11 月)」「冬 (12～2 月)」の 4 ラベル, 時間帯は「0～6 時」「6～12 時」「12～18 時」「18～24 時」の 4 ラベル

mean-shift クラスタリング法のバンド幅を 0.01 とした時のランドマーク数, 検出された移動の総数, 移動に寄与したユーザー数を 表 10 に, 0.025 とした時のものを表 11 に示す.

表 12 : 各実験の条件

実験番号	Mean-shift の バンド幅	気温・天気 のラベル	季節・時間帯 のラベル	考慮した ランドマーク
1	0.01	パターン 1	パターン 1	全て
2	0.01	パターン 1	パターン 1	上位 25
3	0.01	パターン 1	パターン 1	上位 10
4	0.025	パターン 1	パターン 1	全て
5	0.025	パターン 1	パターン 1	上位 25
6	0.025	パターン 1	パターン 1	上位 25
7	0.01	パターン 2	パターン 1	全て
8	0.01	パターン 2	パターン 1	上位 25
9	0.01	パターン 2	パターン 1	上位 10

4.3.2. 実験結果

以上述べてきたデータセットと条件の下行った実験の結果を示す。今回は 9 つの実験を行った。各実験の条件は表 12 の通りである。本実験では、4.1.2.で示した 9 つの推薦手法の間で推薦精度に大きな差が見られず、グラフでは差が分かりにくいため結果は表で表すこととした。1 つの実験の結果は 2 つの表で示され、1 つ目の表に北海道、東京、千葉、神奈川、静岡の 5 都市の結果を、2 つ目の表に大阪・京都・奈良・兵庫・沖縄の 5 都市の結果をそれぞれ掲載している。

実験結果全体を通して、今回は残念ながら推薦手法がベースラインの手法の推薦精度を大きく上回ることはなかった。しかし、都市によっては特徴的な結果も見られ、本研究の今後の発展につながる興味深い結果も得られたと考えている。そのような点も含め、次ページ以降に実験結果を示し、それに続いて詳細な考察を述べる。

(実験 1) mean-shift のバンド幅 : 0.01, 気温・天気ラベル : パターン 1, 季節・時間帯ラベル : パターン 1, 全てのランドマークを考慮

表 13 : 実験 1 の結果 (1) (単位 : %)

モデル名	北海道	東京	千葉	神奈川	静岡
人気順	24.072	9.142	9.784	9.180	7.231
マルコフ	28.030	21.097	21.640	22.984	17.515
気温	27.760	20.921	21.650	22.943	17.380
天気	27.785	21.164	21.607	23.113	17.530
気温・天気	27.711	20.898	21.289	22.792	16.991
季節	26.776	20.874	21.253	23.083	16.737
時間帯	27.072	21.780	21.832	23.351	17.021
季節・時間帯	27.072	21.498	21.751	23.222	17.590
気温・天気・ 季節・時間帯	25.227	20.652	21.194	21.770	15.015

表 14 : 実験 1 の結果 (2) (単位 : %)

モデル名	大阪	京都	奈良	兵庫	沖縄
人気順	15.732	19.667	19.762	16.784	17.231
マルコフ	23.549	25.538	25.843	25.354	19.361
気温	23.331	25.374	25.046	24.952	19.212
天気	23.850	25.549	25.775	25.246	20.703
気温・天気	22.971	25.148	25.138	24.644	20.106
季節	23.446	25.251	25.083	25.189	19.425
時間帯	22.976	24.997	25.592	24.901	18.978
季節・時間帯	22.441	25.175	24.648	24.083	19.212
気温・天気・ 季節・時間帯	20.432	24.118	23.458	22.278	18.360

(実験 2) mean-shift のバンド幅 : 0.01, 気温・天気ラベル : パターン 1, 季節・時間帯ラベル : パターン 1, 上位 25 位以内のランドマークを考慮

表 15 : 実験 2 の結果 (1) (単位 : %)

モデル名	北海道	東京	千葉	神奈川	静岡
人気順	38.218	14.249	14.504	15.195	14.198
マルコフ	44.356	26.576	26.553	30.947	30.455
気温	44.911	26.639	26.478	30.689	29.963
天気	44.317	26.565	26.551	31.019	29.994
気温・天気	44.832	26.604	26.484	30.903	30.393
季節	43.248	26.462	26.443	30.836	29.994
時間帯	43.604	27.436	26.786	31.690	30.578
季節・時間帯	42.455	27.617	26.876	31.488	31.377
気温・天気・ 季節・時間帯	40.634	27.225	26.891	30.740	28.427

表 16 : 実験 2 の結果 (2) (単位 : %)

モデル名	大阪	京都	奈良	兵庫	沖縄
人気順	25.825	24.286	28.871	26.454	28.910
マルコフ	33.906	29.003	35.102	34.404	31.207
気温	34.100	29.249	34.590	33.836	29.821
天気	33.586	28.985	35.139	34.146	32.227
気温・天気	33.974	29.080	34.773	34.043	30.587
季節	32.897	29.383	34.910	34.206	30.259
時間帯	33.799	29.508	34.434	33.689	32.483
季節・時間帯	33.198	29.111	34.535	34.025	30.222
気温・天気・ 季節・時間帯	32.324	28.436	33.684	32.638	28.618

(実験 3) mean-shift のバンド幅 : 0.01, 気温・天気ラベル : パターン 1, 季節・時間帯ラベル : パターン 1, 上位 10 位以内のランドマークを考慮

表 17 : 実験 3 の結果 (1) (単位 : %)

モデル名	北海道	東京	千葉	神奈川	静岡
人気順	44.521	18.451	18.392	19.862	19.442
マルコフ	51.769	30.047	29.972	36.627	36.186
気温	51.998	29.651	30.111	35.704	36.372
天気	52.055	30.061	29.919	36.677	36.326
気温・天気	52.283	30.141	30.063	36.702	36.419
季節	50.970	30.229	29.886	36.740	36.372
時間帯	52.055	30.686	30.334	36.734	34.605
季節・時間帯	51.484	30.897	30.756	36.809	37.302
気温・天気・ 季節・時間帯	49.144	30.789	30.670	36.803	34.233

表 18 : 実験 3 の結果 (2) (単位 : %)

モデル名	大阪	京都	奈良	兵庫	沖縄
人気順	31.949	31.481	35.437	32.151	38.554
マルコフ	38.471	32.999	40.149	38.114	40.620
気温	38.510	33.150	39.800	38.114	32.790
天気	38.391	32.920	39.957	38.114	40.511
気温・天気	38.180	32.241	39.849	38.057	39.478
季節	37.333	32.261	40.089	37.839	39.967
時間帯	38.510	32.739	39.428	38.137	37.085
季節・時間帯	37.743	32.376	39.644	37.599	39.152
気温・天気・ 季節・時間帯	37.068	32.140	39.031	37.049	38.717

(実験 4) mean-shift のバンド幅 : 0.025, 気温・天気ラベル : パターン 1, 季節・時間帯ラベル : パターン 1, 全てのランドマークを考慮

表 19 : 実験 4 の結果 (1) (単位 : %)

モデル名	北海道	東京	千葉	神奈川	静岡
人気順	36.119	33.901	37.337	40.097	16.099
マルコフ	37.211	38.254	41.087	44.840	31.187
気温	36.833	38.363	41.075	45.674	30.547
天気	37.421	38.268	40.976	45.813	31.620
気温・天気	37.925	38.226	41.063	45.482	30.361
季節	36.623	38.235	41.049	45.642	30.650
時間帯	38.597	38.392	41.332	46.013	30.506
季節・時間帯	38.009	38.351	41.278	45.721	31.393
気温・天気・ 季節・時間帯	35.615	38.004	40.864	44.986	28.627

表 20 : 実験 4 の結果 (2) (単位 : %)

モデル名	大阪	京都	奈良	兵庫	沖縄
人気順	29.787	51.192	31.029	27.126	32.908
マルコフ	35.957	53.804	35.857	35.430	38.654
気温	34.958	53.752	35.458	35.129	37.483
天気	35.957	53.623	35.353	35.430	38.574
気温・天気	34.597	53.617	34.767	35.229	38.521
季節	35.408	53.546	35.775	35.484	38.414
時間帯	34.228	53.733	36.056	35.439	38.734
季節・時間帯	34.940	52.972	34.931	34.737	38.228
気温・天気・ 季節・時間帯	33.219	52.585	32.658	33.105	37.430

(実験 5) mean-shift のバンド幅 : 0.025, 気温・天気ラベル : パターン 1, 季節・時間帯ラベル : パターン 1, 上位 25 位以内のランドマークを考慮

表 21 : 実験 5 の結果 (1) (単位 : %)

モデル名	北海道	東京	千葉	神奈川	静岡
人気順	43.974	38.863	39.999	45.717	20.293
マルコフ	45.120	42.874	43.344	50.716	36.596
気温	44.572	42.820	43.543	50.910	36.011
天気	45.418	42.856	43.400	50.291	37.128
気温・天気	45.767	42.808	43.500	50.811	35.638
季節	43.625	42.796	43.437	51.088	35.532
時間帯	46.713	42.928	43.497	51.360	35.931
季節・時間帯	46.614	42.971	43.528	51.311	38.324
気温・天気・ 季節・時間帯	43.376	42.719	43.410	50.886	35.984

表 22 : 実験 5 の結果 (2) (単位 : %)

モデル名	大阪	京都	奈良	兵庫	沖縄
人気順	35.562	56.237	38.130	32.483	35.820
マルコフ	42.249	57.840	42.902	41.476	42.267
気温	41.474	57.664	42.783	41.216	41.168
天気	42.261	57.706	42.544	41.340	42.122
気温・天気	41.274	57.678	42.305	40.957	40.619
季節	42.003	57.531	42.962	41.137	41.515
時間帯	40.122	57.727	42.723	41.036	42.729
季節・時間帯	42.390	57.130	42.753	40.449	41.602
気温・天気・ 季節・時間帯	41.062	56.778	40.710	39.535	40.908

(実験 6) mean-shift のバンド幅 : 0.025, 気温・天気ラベル : パターン 1, 季節・時間帯ラベル : パターン 1, 上位 10 位以内のランドマークを考慮

表 23 : 実験 6 の結果 (1) (単位 : %)

モデル名	北海道	東京	千葉	神奈川	静岡
人気順	49.772	48.645	49.045	56.820	29.651
マルコフ	51.207	50.336	50.683	60.499	46.612
気温	51.011	50.369	50.704	60.595	46.776
天気	51.402	50.336	50.687	60.290	46.982
気温・天気	52.968	50.369	50.711	60.333	47.187
季節	53.294	50.414	50.747	59.957	46.571
時間帯	54.664	50.403	50.755	60.601	41.930
季節・時間帯	53.816	50.288	50.772	60.601	46.817
気温・天気・ 季節・時間帯	52.250	50.398	50.679	60.451	45.092

表 24 : 実験 6 の結果 (2) (単位 : %)

モデル名	大阪	京都	奈良	兵庫	沖縄
人気順	43.868	61.759	44.023	43.409	49.718
マルコフ	50.819	63.100	49.186	50.348	55.184
気温	50.819	63.100	49.166	50.383	56.052
天気	50.767	63.092	49.026	50.417	54.837
気温・天気	50.767	63.084	49.166	50.174	52.972
季節	50.819	63.100	49.588	50.400	56.616
時間帯	48.275	63.140	47.519	50.017	55.228
季節・時間帯	51.307	62.785	50.492	49.635	56.182
気温・天気・ 季節・時間帯	50.749	62.785	49.206	49.878	55.184

(実験 7) mean-shift のバンド幅 : 0.01, 気温・天気ラベル : パターン 2, 季節・時間帯ラベル : パターン 1, 全てのランドマークを考慮

表 25 : 実験 7 の結果 (1) (単位 : %)

モデル名	北海道	東京	千葉	神奈川	静岡
人気順	24.072	9.142	9.784	9.180	7.231
マルコフ	28.030	21.097	21.640	22.984	17.515
気温	27.440	20.973	21.300	22.919	16.811
天気	27.785	21.164	21.607	23.113	17.530
気温・天気	27.342	20.879	21.069	22.549	17.156
季節	26.776	20.874	21.253	23.083	16.737
時間帯	27.072	21.780	21.832	23.351	17.021
季節・時間帯	27.072	21.498	21.751	23.222	17.590
気温・天気・ 季節・時間帯	23.949	20.147	20.707	21.045	15.988

表 26 : 実験 7 の結果 (2) (単位 : %)

モデル名	大阪	京都	奈良	兵庫	沖縄
人気順	15.732	19.667	19.762	16.784	17.231
マルコフ	23.549	25.538	25.843	25.354	19.361
気温	22.381	24.898	24.617	24.603	20.767
天気	23.850	25.549	25.775	25.246	20.703
気温・天気	22.163	24.922	24.408	24.325	20.596
季節	23.446	25.251	25.083	25.189	19.425
時間帯	22.976	24.997	25.592	24.901	18.978
季節・時間帯	22.441	25.175	24.648	24.083	19.212
気温・天気・ 季節・時間帯	20.039	23.960	22.894	21.264	18.062

(実験 8) mean-shift のバンド幅 : 0.01, 気温・天気ラベル : パターン 2, 季節・時間帯ラベル : パターン 1, 上位 25 位以内のランドマークを考慮

表 27 : 実験 8 の結果 (1) (単位 : %)

モデル名	北海道	東京	千葉	神奈川	静岡
人気順	38.218	14.249	14.504	15.195	14.198
マルコフ	44.356	26.576	26.553	30.947	30.455
気温	44.356	26.732	26.106	30.951	29.871
天気	44.317	26.565	26.551	31.019	29.994
気温・天気	43.762	26.703	26.235	31.062	30.731
季節	43.248	26.462	26.443	30.836	29.994
時間帯	43.604	27.436	26.786	31.690	30.578
季節・時間帯	42.455	27.617	26.876	31.488	31.377
気温・天気・ 季節・時間帯	40.238	27.178	26.783	30.665	28.888

表 28 : 実験 8 の結果 (2) (単位 : %)

モデル名	大阪	京都	奈良	兵庫	沖縄
人気順	25.825	24.286	28.871	26.454	28.910
マルコフ	33.906	29.003	35.102	34.404	31.207
気温	32.771	28.825	34.709	34.043	31.571
天気	33.586	28.985	35.139	34.146	32.227
気温・天気	32.402	28.877	34.462	33.862	30.478
季節	32.897	29.383	34.910	34.206	30.259
時間帯	33.799	29.508	34.434	33.689	32.483
季節・時間帯	33.198	29.111	34.535	34.025	30.222
気温・天気・ 季節・時間帯	32.043	28.994	32.604	31.889	28.363

(実験 9) mean-shift の幅 : 0.01, 気温・天気ラベル : パターン 2, 季節・時間帯ラベル : パターン 1, 上位 10 位以内のランドマークを考慮

表 29 : 実験 9 の結果 (1) (単位 : %)

モデル名	北海道	東京	千葉	神奈川	静岡
人気順	44.521	18.451	18.392	19.862	19.442
マルコフ	51.769	30.047	29.972	36.627	36.186
気温	48.687	30.221	30.147	36.168	33.814
天気	52.055	30.061	29.919	36.677	36.326
気温・天気	51.427	30.219	29.704	36.570	34.791
季節	50.970	30.229	29.886	36.740	36.372
時間帯	52.055	30.686	30.334	36.734	34.605
季節・時間帯	51.484	30.897	30.756	36.809	37.302
気温・天気・ 季節・時間帯	48.573	30.644	30.684	36.256	34.930

表 30 : 実験 9 の結果 (2) (単位 : %)

モデル名	大阪	京都	奈良	兵庫	沖縄
人気順	31.949	31.481	35.437	32.151	38.554
マルコフ	38.471	32.999	40.149	38.114	40.620
気温	38.444	32.860	39.428	38.125	40.131
天気	38.391	32.920	39.957	38.114	40.511
気温・天気	38.153	33.319	39.284	38.171	38.934
季節	37.333	32.261	40.089	37.839	39.967
時間帯	38.510	32.739	39.428	38.137	37.085
季節・時間帯	37.743	32.376	39.644	37.599	39.152
気温・天気・ 季節・時間帯	37.029	32.467	38.106	36.592	37.792

4.4. 考察

まず全体として、前述の通り今回の実験では、推薦手法は 2 つのベースラインの手法のうち、人気順に推薦するモデルには勝っているものの、マルコフモデルに基づいた手法の推薦精度を大きく上回ることはなかった。このことから、本実験の対象とした国内の 10 都市についてはマルコフモデルが支配的、すなわち次に訪問するランドマークを決める際には現在地が最も大きな影響を与えると考えられる。しかし、その中でも実験結果を見るといくつかの知見を得ることができ、今後システムを改良・発展させるためのヒントを得ることができる。ここでは、実験から得られた知見について、「ランドマークの特徴」と「データ数」の 2 つの視点を中心に述べる。また、最後にシステムの今後の将来性についても触れる。

4.4.1. ランドマークの特徴から捉える実験結果

始めに、今回実験対象とした国内の 10 都市において、mean-shift クラスタリング手法によって抽出されたランドマークのうち人気上位 10 個のものを表 31, 表 32 に示す。なお、表 31 において東京と千葉のランドマークが一致しているのは誤植ではなく、画像収集の際に設定した収集領域に重複している部分があるためである。収集領域が重ならないように中心地点を設定することも考えたが、千葉にある東京ディズニーランドを中心にするると興味深いデータが得られる可能性があるかと予測し、領域の重なりが大きくなっても敢えて今回の設定にした。大阪・奈良・兵庫に関しても同様に、収集領域の重複のためランドマークの多くが一致している。

2 つの表を見ると、上位 10 位以内のランドマークには、駅や空港などの交通に関する場所が多いことが分かる。これらは、推薦システムにとってノイズになると考えられる。交通機関は気温や天気等の条件によらず、どこかに出かける際には利用しなければならないものだからである。特に空港に関しては、飛行機は搭乗するためには事前に予約が必要なため、予定を変更して「空港に行く・行かない」をその時の状況で決めるのは難しい。また、鉄道の駅に関しては、旅行者だけではなくその土地に住んでいる人も利用する（むしろそちらの方が多くであろう）ため、Sun et al. [22] も指摘した「居住者による旅行とは関係のない画像」によるノイズを引き起こしやすいと考えられる。推薦手法がうまく動作しなかった理由の一端が、この点にもあるかもしれない。

その一方で、北海道や静岡、沖縄に関しては、比較して公園など比較的気候の影響を受けやすいランドマークが多く含まれているように見受けられる。実験結果を見てみると、この 3 都市に関しては実験の設定によっては複数の推薦手法がマルコフモデルよりも推薦精度が高くなっている。例えば、北海道の実験 3・4・5・6 や静岡の実験 3・5・6・9、沖縄の 1・6・7・8 などがそれに当たる。特に、実験 3・6・8・9 は、考慮するランドマークを人気上位のものに絞った実験であり、気温・天気等の要素により影響を受けるランドマーク（例

表 31 : 国内 10 都市のランドマーク人気上位 10 個 (1)

順位	北海道	東京	千葉	神奈川	静岡
1	札幌時計台	渋谷駅	渋谷駅	銀座駅	恩賜箱根公園
2	小樽	東京駅	東京駅	渋谷駅	彫刻の森 美術館
3	桑園駅	新宿駅	新宿駅	新宿駅	箱根湯本駅
4	サッポロビール 博物館	六本木駅	六本木駅	神保町駅	桃源台駅
5	札幌サンプラザ	秋葉原駅	秋葉原駅	六本木駅	大涌谷 黒たまご館
6	丸山公園	東京タワー	東京タワー	秋葉原駅	小田原城跡
7	ウイングベイ 小樽	東京ドーム	浅草寺	東京タワー	箱根園水族館
8	豊平公園	浅草寺	東京ドーム	代官山	熱海駅
9	白い恋人パーク	上野公園	上野公園	横浜駅	星の王子さま ミュージアム
10	平岸駅	品川駅	品川駅	横浜赤レンガ倉庫	根布川駅

表 32 : 国内 10 都市のランドマーク人気上位 10 個 (2)

順位	大阪	京都	奈良	兵庫	沖縄
1	大阪駅	清水寺	道頓堀	大阪駅	国際通り
2	道頓堀	二条城	大阪駅	道頓堀	奥武山公園
3	大阪天満宮	京都駅	大阪天満宮	三ノ宮駅	北谷公園
4	通天閣	平安神宮	通天閣	大阪天満宮	沖縄県立博物館
5	大阪城	京都御所	大阪城	通天閣	那覇空港
6	三ノ宮駅	嵐山	東大寺	大阪城	首里城
7	東大寺	金閣寺	新大阪駅	USJ	沖縄アウトレット モールあしびなー
8	USJ	安楽寺	伏見稲荷大社	新大阪駅	コザ運動公園
9	新大阪駅	伏見稲荷大社	大阪上本町駅	関西国際空港	残波岬
10	京セラドーム ム	出町柳駅	平城宮跡	京セラドーム	宜野湾海浜公園

えば公園など)が多いほど上手くいくことが期待される実験である。

以上より、検出されたランドマークの性質にまで目を向け適切なノイズ処理等を行えば、推薦システムの推薦精度がさらに向上する可能性はある。

4.4.2. データ数から捉える実験結果

次に、推薦手法である「気温・天気・季節・時間帯の4つを考慮した手法」と、その他の全ての手法(推薦手法に準ずるものも含めて)との差について考える。東京・千葉に関しては、推薦手法とその他の手法の推薦精度の差は小さい(全ての実験を通して概ね1%未満)であるのに対し、北海道・静岡ではその差はかなり大きい。これは、データ数の差によるものと考えられる。表8に示した通り、東京・千葉では200万枚以上の画像を収集したのに対し、北海道では約6万5千枚、静岡ではおよそ10万枚しか収集できなかった。画像が少ないと、1つ大きな問題が生じてしまう。それは、ランドマーク間の移動に関する統計の行列が疎になってしまうということである。全てのランドマークを考慮する実験の場合、行列の大きさはmean-shiftのバンド幅を0.01とした場合だと北海道で 313×313 、静岡で 351×351 である。つまり、いずれの行列にも9万個以上のセルがあるということになる。それに対し、この条件下で検出されたランドマーク間の移動は北海道で4,067、静岡で6,680であった(表10)。これでは移動に関する統計の行列は非常に疎になってしまう。さらに、例えば気温モデルでは、気温を表すラベルによって移動を4つないしは7つの行列に振り分けるため、(ラベルによる振り分けが行われない)マルコフモデルの行列と比較して気温モデルの行列はさらに疎になる。そのため、気温モデル、天気モデル、季節モデル、時間帯モデルの各モデルは一種の「不安定性」のようなものがあり、4つの要素を全て考慮した気温・天気・季節・時間帯モデルは4つの疎な行列に基づいているため、さらに不安定性が増し、北海道および静岡ではこの手法が他の手法と比べて推薦精度が低くなってしまった可能性があると考えられる。

この不安定性を解消するためにはデータの数を増やすとよいが、表8において各都市で収集した画像の枚数と取得期間を見れば分かる通り、国内ではあまり多くの画像の収集は期待できない。それに対し、表9から分かるように海外の観光都市では国内都市に比べかなり多くの画像の取得が期待できるので、今回は気象情報の入手の関係で断念したが、海外の都市について実験を行えばより良い推薦精度が得られる可能性がある。

4.4.3. 推薦手法の将来性

4.1.1. において今回のような実験ではcross-validationを用いるべきだと述べたが、敢えて自データを含んだ状態で実験を行ってみると、図21~図30のような結果になった。なお、この実験は自データを含んでいること以外は実験1と同様の条件で行った。

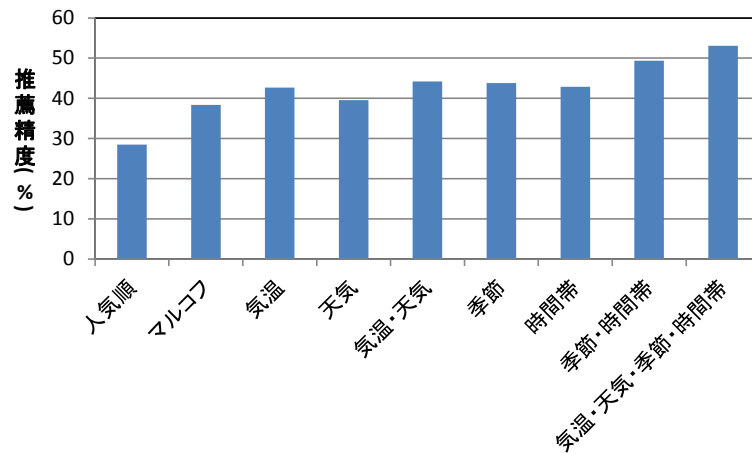


図 21：自データを含んだ状態での実験結果（北海道）

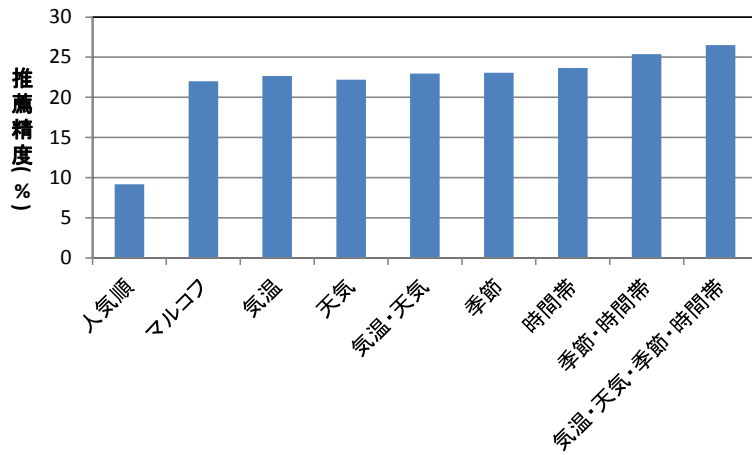


図 22：自データを含んだ状態での実験結果（東京）

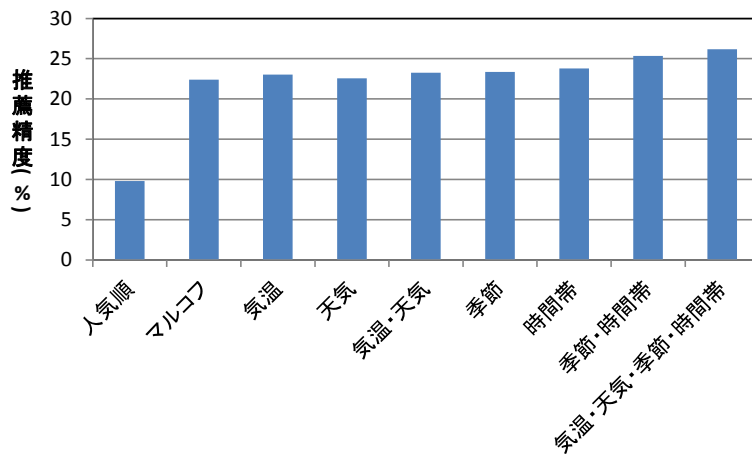


図 23：自データを含んだ状態での実験結果（千葉）

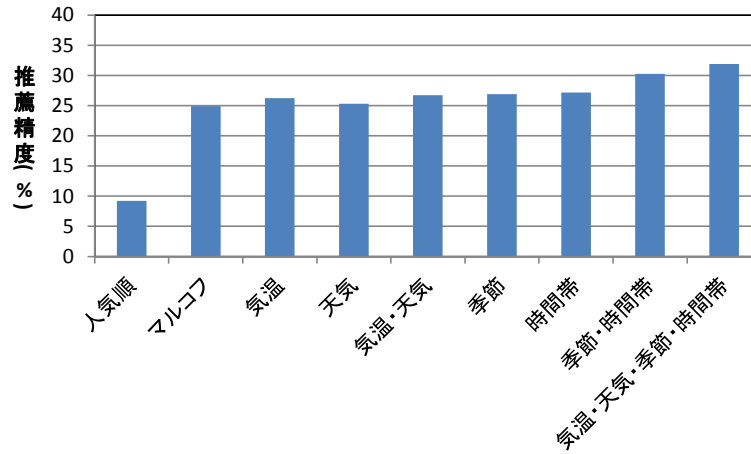


図 24：自データを含んだ状態での実験結果（神奈川）

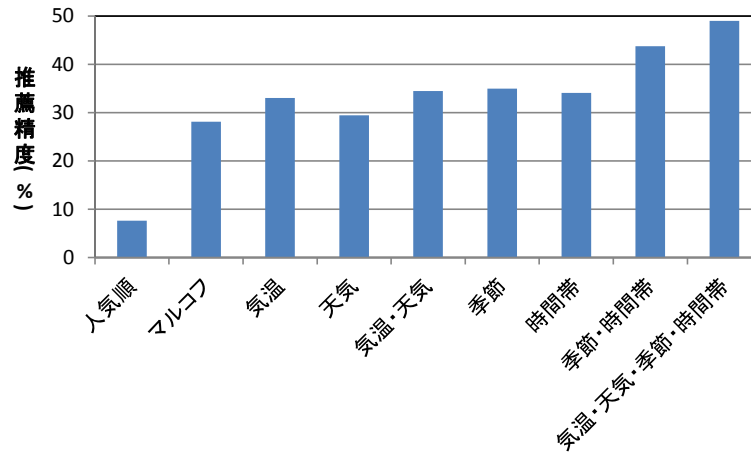


図 25：自データを含んだ状態での実験結果（静岡）

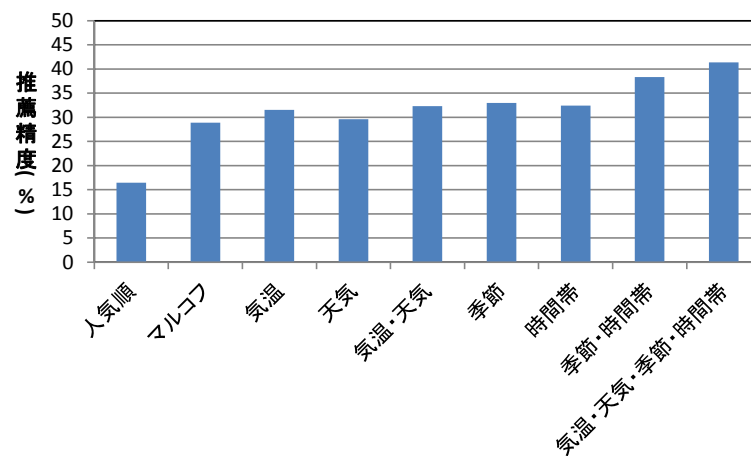


図 26：自データを含んだ状態での実験結果（大阪）

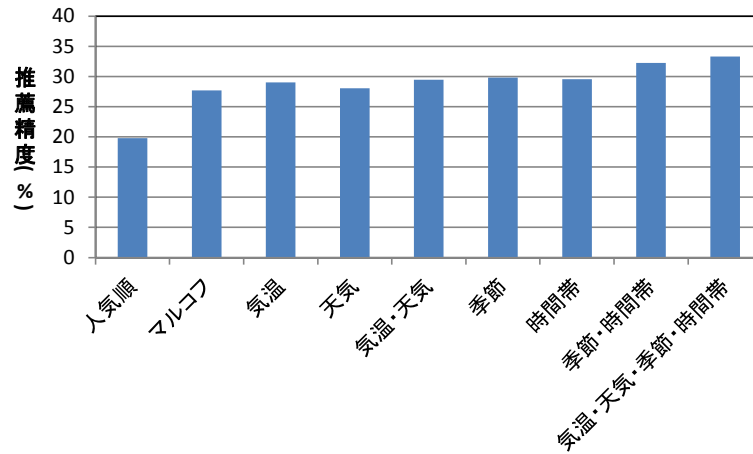


図 27：自データを含んだ状態での実験結果（京都）

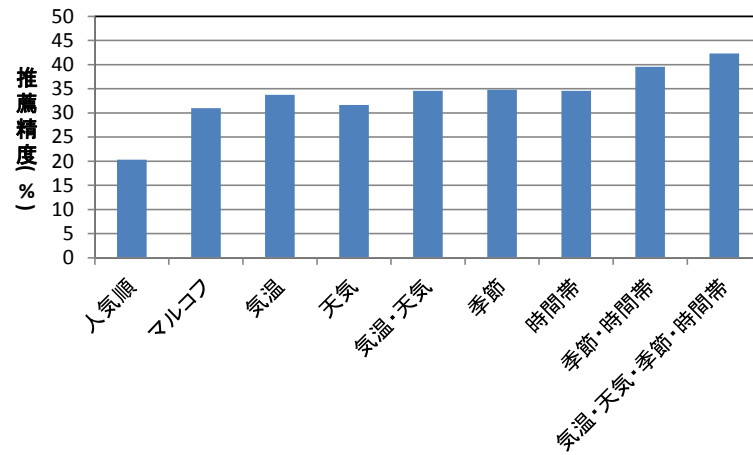


図 28：自データを含んだ状態での実験結果（奈良）

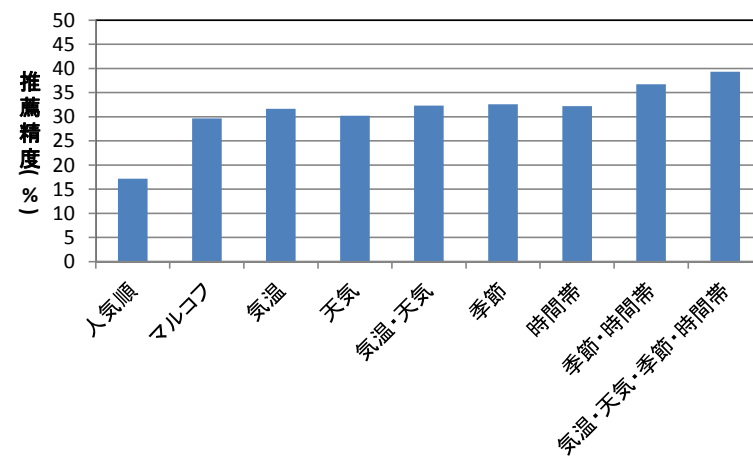


図 29：自データを含んだ状態での実験結果（兵庫）

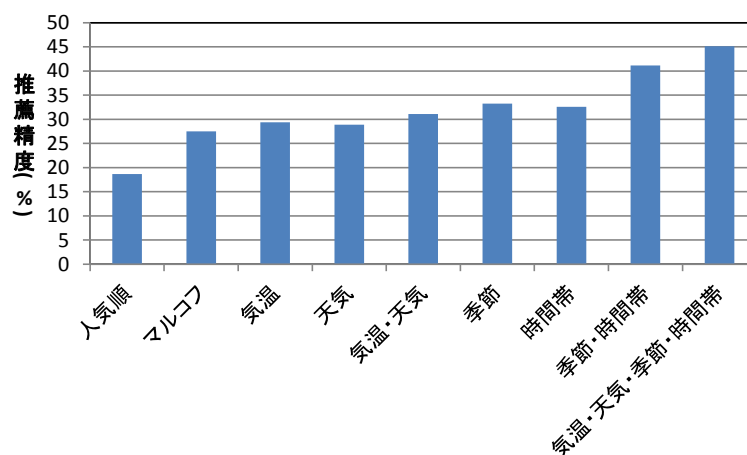


図 30 : 自データを含んだ状態での実験結果 (沖縄)

いずれの都市においても、推薦手法の推薦精度はベースラインの手法の推薦精度を大きく上回っている。これらは、自データを含んで行った実験であり、実験手法としては適切とはいえない。しかし、「自データを含んでいる」すなわち「正解を示すデータが1つ増える」という条件はベースラインの手法でも平等であり、その状態でこのように推薦手法がベースラインの手法の推薦精度に大きく勝っている。したがって、上で述べたノイズ除去やデータ数の確保等の適切な処理を行えば、推薦手法は将来的にはこの程度の推薦精度を出す可能性があるかと期待できる。

第五章 結論

この章で本稿を締めくくる。まず本稿の内容を総括し、それから本研究の今後の展望について言及していく。

5.1. まとめ

本稿では、SNS サイトの人気の高まりに伴って成長した Flickr や Instagram 等の画像共有サイト上の大規模な画像とメタデータの利用法として、旅行ルート推薦の個人化を取り上げた。個人の趣味・嗜好や人の属性など様々な要素を考慮した関連研究がある中、本研究が提案したのは「気温・天気を考慮した旅行ルート推薦」である。また、気温と天気に加え、関連研究 [5] を参考に季節と時間帯についても考慮した。実験に際し Flickr から国内 10 都市で撮影された総計 7,256,628 枚の画像を取得し、そのデータセットとしての有用性にも触れた。第四章の実験においては現段階では望ましい結果が得られなかったものの、提案システムを改善・発展させるための有用な知見を得ることができた。今後は次節で述べる展望を踏まえ、推薦精度の向上を目指す。

5.2. 今後の展望

前章で述べた提案システムの改良策を含め、以下のような項目を実現すれば、推薦精度の向上が期待できる。

- ・データ数を増やす
- ・海外の観光都市のデータも利用する
- ・ランドマークの特徴・性質の精査
- ・様々なノイズの除去
- ・マルコフモデルに基づく関連研究 [4][26] との融合

参考文献

- [1] <https://www.flickr.com/>
- [2] <https://www.instagram.com/>
- [3] Robert Barta, Christina Feilmayr, Birgit Proll, Christoph Grun and Hannes Werthner, "Covering the semantic space of tourism," New York, 2009.
- [4] T. Kurashima, T. Iwata, G. Irie and K. Fujimura "Travel route recommendation using geotags in photo sharing sites," CIKM, pp.579-588, 2010.
- [5] T. Yamasaki, A. Gallagher, T. CHEN, K. Aizawa, "Geotag-based travel route recommendation featuring seasonal and temporal popularity," GeoMM, 2013.
- [6] Michael F. Goodchild, "Citizens as sensors: The world of volunteered geography," *GeoJournal*, Vol. 69, No. 4, pp211-221.
- [7] Xiaoyuan Su and Taghi M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, Vol. 2009: 1-20.a, 2009.
- [8] Greg Linden, Brent Smith and Jeremy York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *Internet Computing*, IEEE 7 (1), 76-80, 2003.
- [9] Marco Mamei, Alberto Rosi, Franco Zambonelli, "Automatic Analysis of Geotagged Photos for Intelligent Tourist Services," *Sixth International Conference on Intelligent Environments*, IEEE, 2010.
- [10] Vincent W. Zheng, Yu Zheng, Xing Xie, Qiang Yang, "Collaborative Location and Activity Recommendations with GPS History Data," *International World Wide Web Conference Committee*, 2010.
- [11] J. Zahalka, S. Rudinac and M. Worring, "New Yorker mélange: interactive brew of personalized venue recommendations," *ACMMM*, 2014.
- [12] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *NIPS*, 2012.
- [13] M. Hoffman, F. R. Bach and D. M. Blei, "Online learning for latent dirichlet allocation," *NIPS*, 2010.
- [14] Kai Jiang, Peng Wang and Nenghai Yu, "ContextRank: Personalized Tourism Recommendation by Exploiting Context Information of Geotagged Web Photos," *Sixth International Conference on Image and Graphics*, IEEE, 2011.
- [15] Kevin Meehan, Tom Lunney, Kevin Curran and Aiden McCaughey, "Context-Aware Intelligent Recommendation System for Tourism," *Work in Progress session at PerCom 2013*, IEEE, 2013.
- [16] Rongrong Ji, Ling-Yu Duan, Jie Chen, Shuang Yang, Hongxun Yao, Tiejun Huang

and Wen Gao, "Learning The Trip Suggestion from Landmark Photos on The Web," 18th IEEE International Conference on Image Processing, IEEE, 2011.

[17] Y. Arase, X. Xie, T. Hara and S. Nishio, "Mining people's trips from large scale geo-tagged photos," ACM MM, 2010.

[18] F. Giannotti, M. Nanni and D. Pedreschi, "Efficient mining of temporally annotated sequences," In Proc. of SDM, 2006.

[19] Junge Shen, Cheng Deng and Xinbo Gao, "Attraction recommendation: Towards personalized tourism via collective intelligence," Neurocomputing, Vol. 173, pp789-798, 2016.

[20] Tripadvisor (<https://tripadvisor.com>).

[21] L. Gomez-Chova, G. Gamp-Valls, J. Munoz-Mari and J. Calpe, "Semisupervised image classification with laplacian support vector machines," IEEE Geoscience and Remote Sensing Letters, Vol. 5, No. 3, pp.336-340, 2008.

[22] Yeran Sun, Hongchao Fan, Mohamed Bakillah and Alexander Zipf, "Road-based travel recommendation using geo-tagged images," Computers, Environment and Urban Systems, Vol. 53, pp110-122, 2015.

[23] Martin Ester, Hans-Peter Kriegel, Jorg Sander and Xiaowei Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," Proceedings of the second international conference on knowledge discovery and data mining, AAAI, 1996.

[24] Yu Zheng, Lizhu Zhang, Xing Xie and Wei-Ying Ma, "Mining interesting locations and travel sequences from CPS trajectories." Proceedings of the 18th international conference on world wide web, 2009.

[25] T. Hofmann, "Collaborative filtering via Gaussian probabilistic latent semantic analysis," Annual ACM SIGIR Conference (SIGIR), pp.259-266, 2003.

[26] A.J. Cheng, Y.Y. Chen, Y.T. Huang, W.H. Hsu and H.Y.M. Liao, " Personalized travel recommendation by mining people attributes from community-contributed photos," ACM MM, pp.83-92, 2011.

[27] T. M. Cover and J. A. Thomas, "Elements of information theory," Wiley, 1991.

[28] J. Hays and A. A. Efros, "IM2GPS: estimating geographic information from a single image," CVPR, 2008.

[29] <http://www.data.jma.go.jp/gmd/risk/obsdl/index.php>

発表文献

国内会議

- [1] 小澤真里奈, 山崎俊彦, 相澤清晴, “大規模メタデータ処理に基づく気温・天気を考慮した旅行ルート推薦,” 映像情報メディア学会年次大会, 2015.
- [2] 小澤真里奈, 山崎俊彦, 相澤清晴, “気温・天気を考慮した大規模位置履歴からの旅行ルート推薦,” 画像符号化シンポジウム (PCSJ) / 映像メディア処理シンポジウム (IMPS), 2016.

謝辞

1度休学していた時期を含めて3年間の修士学生生活の間、私は多くの人に支えられてきました。

指導教員である山崎俊彦准教授は、いつもお忙しいにも関わらず週に一度個別ミーティングの時間を設けてくださり、研究のことだけでなく語学留学や就職活動などの相談にものっていただきました。普段の研究だけでなく学会での発表の準備や自動車の教習所通いなど、色々なことが重なって大変な時期でも乗り切ることができたのは、学生を暖かく見守ってくださる山崎先生のおかげだと思います。

相澤清晴教授には、山崎先生のように直接指導を賜る機会はありませんでしたが、ミーティングの発表の際にはいつも的確なご助言をいただきました。

秘書の松林さんは、学生にはややこしい提出書類のこともわかりやすく説明して下さり、また、いつも研究室が過ごしやすい環境になるように心配りをしてくださいました。何度かご一緒させていただいた女子ランチも美味しく楽しかったです。

また、卒論生時代の指導教員だった田浦健次朗教授には、今でも感謝しております。とても教育熱心な先生で、右も左も分からなかった私に研究の基礎を教えてくださいました。学生の間では怖いイメージを抱かれがちですが、本当はとても優しい先生です。

もちろん、今まで私を育ててくれた両親を始め、家族や親戚、友人にも感謝しています。この場を借りて皆様に心よりお礼申し上げます。

2017年2月3日

小澤 真里奈