

ジオタグ付き写真の被写体を考慮した意味的な移動軌跡の分析

瀧本 祥章[†] 杉浦 健人[†] 石川 佳治[†]

[†] 名古屋大学大学院情報科学研究科 〒464-8601 愛知県名古屋市千種区不老町

E-mail: [†] {takimoto, sugiura}@db.ss.is.nagoya-u.ac.jp, ishikawa@is.nagoya-u.ac.jp

あらまし GPS を搭載したモバイルデバイスの普及に伴い、ジオタグ付き写真などにより構成される意味的な移動軌跡が増加している。従来の移動軌跡はいつ、どこにユーザがいたのかのみを表す。これに対し、意味的な移動軌跡はその地点でユーザが何に注目しているのかを表し、関連する研究が盛んに行われている。本研究では、重み付きタグ集合により構成される意味的な移動軌跡に対して頻出パターン抽出を行う。まず、意味的な移動軌跡から多くのユーザが訪問し、同じものに注目した領域を抽出する。次に、頻出する領域間の移動を抽出する。最後に、実データを用いた実験により、提案手法の有用性を示す。

キーワード 時空間分析, 意味的な移動軌跡, Flickr, 軌跡マイニング, 時空間データマイニング

1. はじめに

GPS を搭載したモバイルデバイスの普及に伴い、位置情報に関連するサービスが増加している。例えば Twitter [5] であれば、ツイートが「どこで」つぶやかれたかをジオタグ付きツイートで表している。他にも、Flickr [2] であれば、写真が「どこで」撮影されたのか、Swarm [4] であればユーザが「どこに」チェックインしたのかを表す。

このような位置情報付きのデータは、意味的な移動軌跡として近年注目されている [19]。従来の移動軌跡では、ユーザが「いつ」・「どこに」いたのかのみが表されるのに対し、意味的な移動軌跡はその地点に関する追加情報を持つ。例として Flickr のジオタグ付き写真を考える。あるユーザが観光地で写真を撮影し、Flickr にアップロードしたとする。このとき、アップロードされた写真を時刻順に並べることで、図 1 のような写真に関する意味的な移動軌跡が生成できる。つまり、この例における追加情報は写真であり、ユーザが注目しているものを示す。このような意味的な移動軌跡を解析することにより、ユーザが「いつ」・「どこで」・「何に」注目していたのかを把握できる。

意味的な移動軌跡を活用するために、著者らの研究グループでは頻出パターン抽出 [14] を行っている。つまり、複数のユーザが同じように移動し、同じ何かに注目した軌跡をパターンとして抽出している。図 1 の例であれば、軌跡 T_1 と軌跡 T_2 が同じように移動し、 $\{p_2, p_6\}$ では鳥居、 $\{p_3, p_7\}$ では寺と、同じものに注目している。そのため、 $\{p_2, p_6\}$ から $\{p_3, p_7\}$ へ移動するパターンを抽出できる。

しかし、上述の文献 [14] には 2 つの問題がある。1 つ目の問題は、適用できる意味的な移動軌跡が制限されることである。文献 [14] では、追加情報としてタグを持つ意味的な移動軌跡を対象とする。例えば、チェックインデータであればチェックインした POI (Point Of Interest) のカテゴリ情報をタグとして利用できる。しかし、ツイートや写真により構成される意味的

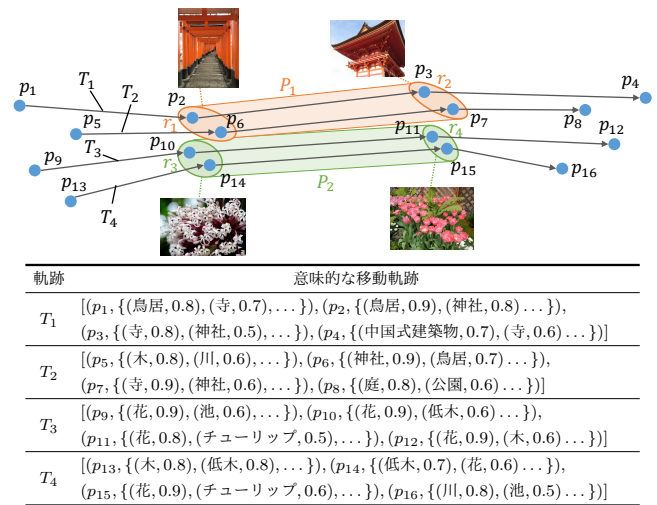


図 1: パターン例 (注1)

な移動軌跡はタグを持たないため、適用できない。また、人手によるタグ付けも考えられるが、データ量によっては現実的ではない。2 つ目の問題は、前提条件として、注目すべき地点が全て既知、即ち有名な観光地や施設であるとする点である。そのため、「境内で紫陽花が綺麗に咲く場所」、「漫画のシーンの元になった場所」など、興味を持つ人が一定数いるが有名ではない場所に対応できない。

そこで本研究では、各点の特徴を重み付きタグ集合で表し、注目すべき場所である ROI (Region Of Interest) の動的な抽出により、上述の問題を解決する。まず 1 つ目の問題は、機械学習による追加情報の重み付きタグ集合への変換により解決する。例えば、ツイートであればトピック分析、写真であれば Google Cloud Vision API [3] (以降、Vision API と呼ぶ) などの一般物体認識技術が利用できる。これにより、機械的に処理できるためビッグデータにも対応可能である。次に 2 つ目の問題は、観光地などの背景知識を用いずに、機械学習の分析結果のみから ROI を抽出することにより解決する。これにより、上述のような有名でない場所も ROI として抽出できる。

(注1): 写真は Flickr から収集したデータセット中の画像 (以下同様)

本研究では、追加情報として重み付きタグ集合を持つ意味的な移動軌跡を対象とし、それらから頻出パターンを抽出する手法を提案する。まず、各意味的な移動軌跡の位置情報と重み付きタグ集合から、多くのユーザが訪問し同じものに注目した領域である ROI を抽出する。次に、抽出した ROI 間の移動から頻出するパターンの抽出を行う。例えば、図 1 のような意味的な移動軌跡 $\{T_1, T_2, T_3, T_4\}$ と最小支持度 2 が与えられたとする。位置情報のみに注目すると、各移動軌跡は同じ移動パターンを持つように見える。しかし実際には、各ユーザは異なるものに注目しており、2 つの移動パターンが得られる。まず、点 p_2 と点 p_6 は鳥居など同じタグを持ち、近傍に存在する点であることから、ROI r_1 として抽出できる。同様に、 p_3 と p_7 から r_2 、 p_{10} と p_{14} から r_3 、 p_{11} と p_{15} から r_4 を抽出できる。次に、ROI 間の移動の内、最小支持度以上のものを抽出する。この例では最小支持度が 2 であるため、 $P_1 = [r_1, r_2]$ と $P_2 = [r_3, r_4]$ を抽出できる。

本稿の構成は以下のとおりである。まず 2. で関連研究として、移動軌跡からのパターン抽出などについて述べる。次に 3. で提案手法の概要について述べ、ROI の抽出について詳細を 4. で述べる。5. では、実データを用いた実験について述べ、最後に、6. で本稿のまとめと今後の課題について述べる。

2. 関連研究

本章ではまず、移動軌跡からパターンを抽出する研究を紹介する。次に機械学習により、タグを付与するツールを紹介する。

2.1 移動軌跡からのパターン抽出

移動軌跡からパターン抽出を行う研究は、対象の移動軌跡の種類によって以下の 2 種類にわけられる。

- 従来の移動軌跡を対象とする研究
移動軌跡の各点は位置情報及び時刻を持つ。
- 意味的な移動軌跡を対象とする研究
移動軌跡の各点は位置情報、時刻だけでなく、訪問した POI の情報やテキスト、画像などの意味的な情報を持つ。

これらの移動軌跡を対象とする研究について順に説明する。

2.1.1 従来の移動軌跡を対象とするパターン抽出

従来の移動軌跡を対象とするパターン抽出の代表的な研究に T-Pattern [10] がある。T-Pattern では $ROI_0 \xrightarrow{\Delta t_1} ROI_1 \xrightarrow{\Delta t_2} \dots \xrightarrow{\Delta t_n} ROI_n$ と表されるパターンを抽出する。ここで、 ROI_i は多くの軌跡が通過する領域であり、 Δt_i は ROI 間の遷移時間を表す。他にも一定時間以上の間一定距離以内を共に移動するパターンである flock [16] や、swarm [17], convoy [13], gathering [22], CACT [12] などさまざまなパターンが提案されている。また、ストリームデータからリアルタイムにパターンを抽出する研究 [8] も行われている。

2.1.2 意味的な移動軌跡を対象とするパターン抽出

意味的な移動軌跡を対象とするパターン抽出は Alvares ら [6] が初めに行った。Alvares らは従来の移動軌跡からユーザの滞在地を抽出し、その属性を地図から識別する。その後、各滞

在地をアイテムとしてとらえ、シーケンスパターンとして抽出する。しかし、滞在地ベースのパターンは、近傍に存在するほぼ同一の滞在地を区別してしまうという問題があった。

これに対し、Zhang らや Jiang らは近傍に存在する同一の滞在地をまとめることにより、この問題を解決した [14, 21]。これらの研究では、ユーザが訪れた POI のカテゴリ情報を追加情報とする意味的な移動軌跡を対象とし、近隣に存在する同一カテゴリを持つ POI をクラスタリングにより、1 つのグループとしてまとめてシーケンスパターンを抽出した。なお、クラスタリングは、Zhang らが mean-shift [7] ベース、Jiang らが DBSCAN [9] ベースで行っている。しかしこれらの研究には、POI の一覧とそのカテゴリ情報が既知であり、ユーザが訪れた POI も明らかであることを前提とするという問題がある。これに対し我々の研究では、意味的な移動軌跡から ROI の特定及び、特徴を抽出するため、未知の ROI に対応可能であり、事前情報を必要としない。一方、Ying ら [20] はユーザの位置予測のために意味的なパターンの抽出を行った。しかし、パターン抽出の際には位置情報に注目せず、ラベルのみに注目している。そのため、得られるパターンは「学校 → 公園 → レストラン」などのラベルで表され、位置情報は無視される。これに対し、我々の研究では、位置情報も活用してパターンの抽出を行う。

Kim らの TOPTRAC [15] はテキストを追加情報として持つ意味的な移動軌跡からパターンの抽出を行った。具体的には、同一トピックのテキストが多く投稿される領域を特定し、その領域間の遷移の内、頻出するものをパターンとして抽出する。しかし与えるパラメータに、領域数やトピック数が必要であるが、これらの値を分析前に適切な値を設定することは困難である。これに対し本研究では、与えるパラメータは全て閾値であり、抽出したいパターンの性質から設定できる。

2.2 機械学習によるタグの付与

本節では実験で用いる分析ツールである Clarifai API [1] や Vision API [3] について述べる。

Clarifai API はアメリカの Clarifai Inc. が提供しているサービスであり、一般物体認識や職場閲覧不適切画像判定を行える。一方、Vision API は Google Inc. が提供しているサービスであり、一般物体認識や OCR などを行える。両サービスは画像と実行する分析を入力として受け取り、分析結果を出力する。各分析結果はその分析の確からしさを表す確信度が付加されている。例えば、両サービスを用いて伏見稲荷大社の鳥居の写真を一般物体認識すると、図 2 のように複数のタグと確信度を得られる。

3. 提案手法の概観

提案手法は「ROI 系列への変換」と「意味的な軌跡パターンの抽出」の 2 ステップに分けられる。本章では、これらについて順に述べる。また、表 1 に本稿で用いる記号の一覧を示す。

3.1 ROI 系列への変換

このステップでは、意味的な移動軌跡の集合 $Traj$ を入力として受け取り、多くのユーザが同一のものに注目する領域である ROI を抽出する。その後、意味的な移動軌跡をその軌跡が



(a) Clarifai API

タグ	確信度
people	0.986
festival	0.966
tourist	0.961
group	0.953
temple	0.949
⋮	⋮

(b) Vision API

タグ	確信度
chinese architecture	0.982
torii	0.906
shinto shrine	0.851
temple	0.848
building	0.814
⋮	⋮

図 2: 分析ツールによる写真の分析結果例

表 1: 本稿で用いる記号一覧

記号	意味
$Traj$	意味的な移動軌跡の集合（データセット）
p_i	点
$traj$	意味的な移動軌跡
$pattern$	意味的な軌跡パターン
r_i	ROI
t_j	タグ
$c_{t_j}^{p_i}$	点 p_i のタグ t_j の確信度
ε	距離閾値
τ	類似度閾値
$MinPts$	密度閾値
$MinSup$	最小支持度

通過する ROI の系列に置き換える。以下では、入力である意味的な移動軌跡の定義及び、抽出する ROI について述べる。

まず、意味的な移動軌跡を以下のように定義する。

[定義 1] 意味的な移動軌跡 $traj = [p_1, \dots, p_n]$ は同一ユーザが生成した点を時系列順に並べた系列である。各意味的な移動軌跡の長さ n は異なってもよいものとする。

ここで、軌跡上の各点 $p_i \in traj$ は $(p_i.loc, p_i.tags, p_i.con)$ のタプルで定義される。 $p_i.loc$ は位置を表す二次元の座標であり、 $p_i.tags$ はタグ集合である。また、 $p_i.con$ は各タグの確信度を表す。タグ集合及び確信度の詳細な定義は後述する。なお、本稿では移動軌跡の時刻情報を各点の前後関係を定めるために用いるため、この定義では省略している。本稿では以降、意味的な移動軌跡を単に軌跡と呼ぶ。

各点が保持するタグ集合及び確信度を以下のように定義する。

[定義 2] タグ集合 $p_i.tags = \{t_1, \dots, t_l\}$ は各点 p_i が保持するタグ t_j の集合である。

[定義 3] 確信度 $p_i.con = \{c_{t_1}^{p_i}, \dots, c_{t_l}^{p_i}\}$ は確信度の集合である。各確信度は点 p_i が保持するタグ t_j の確からしさを表し、 $0 \leq c_{t_j}^{p_i} \leq 1$ である。なお、タグ t_j がタグ集合 $p_i.tags$ に含まれない場合、その確信度は 0 である。

次に、本稿で抽出する ROI を以下のように定義する。

[定義 4] ROI r_i は近傍に存在する類似した点の集合である。

ただし、近傍や類似の基準はユーザやシナリオにより異なり、ROI を抽出する際にはこれらを柔軟に変化させる必要がある。本稿における具体的な定義及び導出は 4. で述べる。

最後に、以下の手順で各軌跡を通過する ROI の系列に置き換える。まず、各点 p_i をそれが属する ROI r_j に置換する。ここで、点 p_i がいずれの ROI にも属さない場合は削除する。次に、連続する同一の ROI を削除をする。例えば、ROI の系列 $[r_1, r_1, r_1, r_2, r_2, r_3, r_1]$ は $[r_1, r_2, r_3, r_1]$ になる。

3.2 意味的な軌跡パターンの抽出

このステップでは、入力として ROI 系列の集合と最小支持度 $MinSup$ を受け取り、頻出する ROI 系列を意味的な軌跡パターンとして出力する。本稿における意味的な軌跡パターンの定義は以下のとおりである。

[定義 5] 意味的な軌跡パターン $pattern = [r_1, \dots, r_m]$ は与えられた最小支持度 $MinSup$ 以上の軌跡が順序どおりに通過する ROI の系列である。

本稿では以降、意味的な軌跡パターンを単にパターンと呼ぶ。

本稿では PrefixSpan [11] により ROI 系列からパターンを抽出する。PrefixSpan ではまず、長さ 1 の頻出するパターンを抽出する。次に、発見済みの頻出パターンの末尾に長さ 1 の頻出パターンを追加したパターンについて、各々頻出であるか調べる。そして、頻出であったパターンに対し、更に長さ 1 の頻出パターンを追加して頻出であるか調べる。以後、同様の処理を頻出パターンが発見できなくなるまで繰り返す。

4. ROI の抽出

本章では、ROI の抽出について述べる。ROI は多くのユーザが同一のものに興味を持つ領域である。そのため、軌跡中のどの点が同一のものを表し、同領域に含まれることをどのように判断するかが重要となる。本稿では、各点の特徴はタグ集合により表されるため、タグ集合間の類似度を定め、分析者が与える閾値以上の類似度を持つ点を同一とみなす。また、同領域の判断には Jiang らと同様に DBSCAN [9] を用いる。ただし、DBSCAN では類似度を考慮することができないため、類似度を考慮するように拡張した SimDBSCAN を提案する。最後に ROI の特徴を特定するため、ROI に含まれる点の重み付きタグ集合からトピックを抽出する。

4.1 点間の類似度

各点は重み付きタグ集合を持っているため、2 点間の類似度をタグ集合間の類似度と定義する。タグ集合間の類似度には Jaccard 係数を用いることが考えられるが、Jaccard 係数では確信度と各タグの価値の違いを考慮できないため、以下の拡張を行う。

各タグの確信度

各点が保持するタグ集合は機械学習により統一的な基準で付加されたものであるため、類似する点は同一のタグを同程度の

確信度で保持すると予想される。しかし、Jaccard 係数はタグの有無のみを用いて定義され、確信度の情報が失われる。そこで、同程度の確信度であるときに類似度が大きくなるようにタグの確信度を重みとして使用する。この拡張を行った重み付き Jaccard 係数の式は以下のようになる。

$$Jaccard'(p_i, p_j) = \frac{\sum_{t_k} \min(c_{t_k}^{p_i}, c_{t_k}^{p_j})}{\sum_{t_k} \max(c_{t_k}^{p_i}, c_{t_k}^{p_j})} \quad (1)$$

各タグの価値の違い

タグ集合に含まれるタグには、多くの点が共通して保持するものと、そうでないものが存在すると予想され、実際に本稿の実験で用いたデータセットにもそのような傾向がみられた (5.1 節 参照)。前者のタグを共通して持っている 2 点と後者のタグを共通して持っている 2 点では、後者の方がより類似していると考えられる。そこで、各タグ t の逆文書頻度 [18] を以下のように定義し、重みとして使用する。

$$idf_t = \log \frac{N}{|\{p_i | t \in p_i.tags\}|} \quad (2)$$

ただし、 N はデータセット中の点の数である。

これらの拡張から、2 点 p_i, p_j 間の類似度を以下のように定義する。

$$\text{sim}(p_i, p_j) = \frac{\sum_{t_k} \min(c_{t_k}^{p_i}, c_{t_k}^{p_j}) \times idf_{t_k}}{\sum_{t_k} \max(c_{t_k}^{p_i}, c_{t_k}^{p_j}) \times idf_{t_k}} \quad (3)$$

なお、 $\text{sim}(p_i, p_j)$ は Jaccard 係数と同様に $0 \leq \text{sim}(p_i, p_j) \leq 1$ を満たし、その値が大きいほど、点 p_i, p_j は類似する。

4.2 密度と類似度を考慮したクラスタリング

本節ではまず、密度と類似度を考慮したクラスタリングである SimDBSCAN に必要となる概念を順に定義する。その後、SimDBSCAN のアルゴリズムを示す。

4.2.1 概念の定義

SimDBSCAN に必要となる概念の定義について述べる。

DBSCAN では密集した点をクラスタにするため、周囲に点が密集している点をコアポイントとして定義している。SimDBSCAN では拡張を行い、周囲に類似した点が密集している点をコアポイントとして以下のように定義する。

[定義 6] (ϵ, τ) -近傍 $Neighbor_{\epsilon, \tau}(p_i) = \{p_j | \text{dist}(p_i, p_j) \leq \epsilon, \tau \leq \text{sim}(p_i, p_j), i \neq j\}$ は点 p_i から距離 ϵ 以内であり、点 p_i との類似度が τ 以上の点の集合である。なお、 $\text{dist}(p_i, p_j)$ は点 p_i, p_j 間のユークリッド距離である。

[定義 7] (ϵ, τ) -近傍の点が $MinPts$ 以上ある点 p_i をコアポイントという。

例えば図 3 のように点があるとき、点 p_2 から距離 ϵ 内に存在しかつ類似度が 0.3 以上である点は p_1 と p_3 であるため、点 p_2 の $(\epsilon, 0.3)$ -近傍は $\{p_1, p_3\}$ である。一方、類似度が 0.5 以上の点は p_3 のみであるため、 $(\epsilon, 0.5)$ -近傍は $\{p_3\}$ となる。また、 $(\epsilon, \tau, MinSup) = (\epsilon, 0.3, 2)$ の場合、コアポイントは p_2, p_3 であり、 $(\epsilon, \tau, MinSup) = (\epsilon, 0.5, 2)$ の場合、コアポイントは p_3

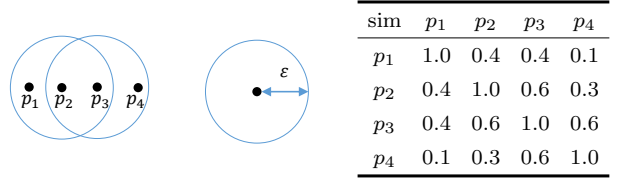


図 3: SimDBSCAN の例 (表は 2 点間の類似度を表す)

である。

次に DBSCAN では、密度が大きい部分の繋がりを調べるため、直接密度到達可能や密度到達可能、密度接続を定義している。類似する点が多く存在する領域の繋がりを調べるため、SimDBSCAN ではこれらに拡張を行い、直接類似到達可能、類似到達可能、類似接続をそれぞれ以下のように定義する。

まず、コアポイントとその周囲の関係を表す直接類似到達可能を以下のように定義する。

[定義 8] 点 p_i, p_j 及び、閾値 $\epsilon, \tau, MinPts$ が与えられ、以下の条件を全て満たすとき、点 p_i は点 p_j に $(\epsilon, \tau, MinPts)$ に関して直接類似到達可能である。

- i) $p_j \in Neighbor_{\epsilon, \tau}(p_i)$
- ii) $MinPts \leq |Neighbor_{\epsilon, \tau}(p_i)|$

例えば、図 3 中の点 p_2 は点 p_1 及び点 p_3 に $(\epsilon, 0.3, 2)$ に関して直接類似到達可能であるが、 $(\epsilon, 0.5, 2)$ に関しては点 p_2 がコアポイントとならないため、直接類似到達可能でない。

次に、直接類似到達可能の推移的關係である類似到達可能を以下のように定義する。

[定義 9] 一連の点 $p_1, \dots, p_i, \dots, p_k$ があり、 p_{i+1} は $p_i (1 \leq i < k)$ に $(\epsilon, \tau, MinPts)$ に関して直接類似到達可能であるとき、点 p_1 は点 p_k に $(\epsilon, \tau, MinPts)$ に関して類似到達可能である。

例えば、図 3 中の点 p_3 は点 p_1 に $(\epsilon, 0.3, 2)$ に関して類似到達可能であるが、 $(\epsilon, 0.5, 2)$ に関しては類似到達可能でない。ここで、点 p_3 から点 p_1 は $(\epsilon, 0.3, 2)$ に関して類似到達可能であるが、点 p_1 から点 p_3 は類似到達可能でない。このように類似到達可能は対称性を持たない。そこで、DBSCAN と同様に、対称性を持つ類似接続を以下のように定義する。

[定義 10] 点 p_o, p_i, p_j があり、点 p_o が点 p_i 及び、点 p_j に $(\epsilon, \tau, MinPts)$ に関して類似到達可能であるとき、点 p_i と点 p_j は $(\epsilon, \tau, MinPts)$ に関して類似接続するという。

例えば、図 3 中の点 p_1, p_4 は $(\epsilon, 0.3, 2)$ に関して類似接続するが、 $(\epsilon, 0.5, 2)$ に関しては類似接続しない。

最後に SimDBSCAN では ROI (クラスタ) を以下のように定義する。

[定義 11] **ROI** $r = \{p_1, \dots, p_k\}$ は以下の条件を全て満たす点の集合である。

- i) r 中の点 p_i に $(\epsilon, \tau, MinPts)$ に関して類似接続する点 p_j は r に属する。

- ii) r 中の任意の点の組 p_i, p_j は $(\varepsilon, \tau, MinPts)$ に関して類似接続する。

例えば、図 3 のように点があり、 $(\varepsilon, \tau, MinSup) = (\varepsilon, 0.3, 2)$ としたときには、全ての点が類似接続するため 1 つの ROI $r_1 = \{p_1, p_2, p_3, p_4\}$ が得られる。これに対し、 $(\varepsilon, \tau, MinSup) = (\varepsilon, 0.5, 2)$ としたとき、 p_2 がコアポイントでなくなるため、 p_1 が他の点に類似接続しない。そのため、得られる ROI は $r_2 = \{p_2, p_3, p_4\}$ となる。

ここで、SimDBSCAN で得られる ROI は類似接続した点の集合であるため、ROI 内の任意の 2 点間の類似度が小さくなる恐れがある。実際、前述の ROI r_1 中の 2 点 p_1, p_4 の類似度 $\text{sim}(p_1, p_4)$ は 0.1 であり、類似度閾値 τ 未満になってしまう。このように SimDBSCAN によるクラスタリングは ROI 中の任意の 2 点間の類似度の大きさを保証しない。この問題については、5.2 節の実験により検証する。

4.2.2 アルゴリズム

SimDBSCAN のアルゴリズムをアルゴリズム 1 に示す。SimDBSCAN は入力として、軌跡上の全ての点の集合 $\mathcal{D}$ 及び、距離閾値 ε 、類似度閾値 τ 、密度閾値 $MinPts$ を受け取り、クラスタ (ROI) の集合 $\mathcal{C}$ を抽出し出力する。処理の流れは次のとおりである。まず、データセット内の各点について順に処理済み (is_visited) か確認する。処理済みである場合には処理を行わず (3-4 行目)、次の点を調べる。未処理の場合には、処理済みのマーキングを行い、点がコアポイントであるか確認する。コアポイントでない場合はその点はノイズとなる (6-8 行目)。コアポイントである場合は関数 `expandCluster` (アルゴリズム 2) により同一の ROI に含まれる点を全て抽出する (10-12 行目)。関数 `expandCluster` (アルゴリズム 2) は入力として、点 p 、点 p の (ε, τ) -近傍 $Neighbors$ 、ROI r 、距離閾値 ε 、類似度閾値 τ 、密度閾値 $MinPts$ を受け取り、次のように処理を行う。まず、点 p を ROI r に追加する (1 行目)。次に $Neighbors$ 内の未処理 (`is_visited`) の点 p' を処理済みとし、コアポイントであるか調べる (5-6 行目)。コアポイントである場合は点 p' の (ε, τ) -近傍である点を $Neighbors$ に追加する (7 行目)。更に、 p' が既に ROI に割り当てられていない場合に ROI r に追加する (8-9 行目)。この処理を $Neighbors$ が空になるまで行う。

各 ROI の特徴を明らかにするため、ROI 内の各点のタグを分析する。まず、ROI r_i の各タグ t_j のスコア ($score_{ij}$) を以下の式で計算し、値が大きい k 個のタグを r_i のトピックとする。

$$score_{ij} = \sum_{p_i \in r_i} c_{t_j}^{p_i} \times idf_{t_j} \quad (4)$$

5. 実験

本章では、提案手法によるパターン抽出手法の有効性を確認するため、実データを用いた実験について紹介する。まず、実験に用いたデータについて説明する。その後、抽出した ROI やパターンについて述べる。

5.1 実験データ

本稿では、写真から構成される意味的な移動軌跡に対して、

アルゴリズム 1: SimDBSCAN

Input: $\mathcal{D}, \varepsilon, \tau, MinPts$ // $\mathcal{D}$ は軌跡上の全ての点の集合
Output: $\mathcal{C}$ // ROI の集合

```

1  $\mathcal{C} \leftarrow \emptyset$  // 抽出した ROI 集合
2 foreach  $p \in \mathcal{D}$  do
3   if is_visited( $p$ ) then
4     continue
5    $p$  を visited とマーキング
6    $Neighbors \leftarrow \text{rangeQuery}(p.loc, \varepsilon, \tau)$  //  $(\varepsilon, \tau)$ -近傍の点
7   if sizeOf( $Neighbors$ ) <  $MinPts$  then
8      $p$  をノイズとマーキング // コアポイントでない
9   else
10     $r \leftarrow \emptyset$  // 新しい ROI
11    expandCluster( $p, Neighbors, p, \varepsilon, MinPts, \tau$ )
12     $r$  を  $\mathcal{C}$  に追加

```

アルゴリズム 2: `expandCluster`

Input: $p, Neighbors, r, \varepsilon, \tau, MinPts$

```

1  $p$  を ROI  $r$  に追加
2 foreach  $p' \in Neighbors$  do
3   if is_visited( $p'$ ) then
4      $p'$  を visited とマーキング
5      $Neighbors' \leftarrow \text{rangeQuery}(p'.loc, \varepsilon, \tau)$ 
6     if sizeOf( $Neighbors'$ )  $\geq MinPts$  then
7        $Neighbors$  に  $Neighbors'$  の要素を追加
8   if  $p'$  が未割当の場合 then
9      $p'$  を  $r$  に追加

```

表 2: 地域別のデータ数

	京都	神奈川
点の数	41866	33594
軌跡数	7533	6283
軌跡の長さ (mean $\pm$ SD)	5.56 $\pm$ 9.91	5.35 $\pm$ 8.48
ユーザ数	1139	797
ユーザごとの点の数 (mean $\pm$ SD)	36.8 $\pm$ 101.7	42.2 $\pm$ 241.5

各写真の被写体を分析して得られた結果をデータセットとして用いる。まず、Flickr から 2015 年に京都及び神奈川で撮られた写真を収集した。次に各写真に対し、Clarifai API 及び、Vision API を用いて分析を行い、被写体を表すタグと確信度を求めた。これにより、4 つのデータセットを得た。各データセットに対し、ユーザごと、撮影日ごとに点を分け、意味的な移動軌跡とした。地域ごとの写真数や軌跡数は表 2 のとおりである。軌跡の最大の長さは京都が 230 個、神奈川が 240 個であり、ユーザごとの点の数の最大数は京都が 1495 個であり、神奈川が 6009 個である。また、各データセットのタグと確信度について、タグの種類の総数、1 点当たりのタグ数などは、表 3 のようになった。Clarifai API と Vision API を比較すると Clarifai API の方が各タグの確信度が高く、1 点当たりの平均タグ数が 20 個と多い。その一方、京都と神奈川で比較すると点の数が京都の方が多くにもかかわらず、神奈川の方がタグ

表 3: タグの数と確信度

	京都 (Clarifai)	神奈川 (Clarifai)	京都 (Vision)	神奈川 (Vision)
タグの種類数	2905	3198	2744	3254
タグの総数	841400	673840	375609	321418
1 点当たりのタグ数	20.0 ± 0.0	20.0 ± 0.0	9.0 ± 5.0	9.5 ± 5.2
確信度 (mean $\pm$ SD)	0.92 ± 0.06	0.93 ± 0.06	0.72 ± 0.13	0.73 ± 0.13

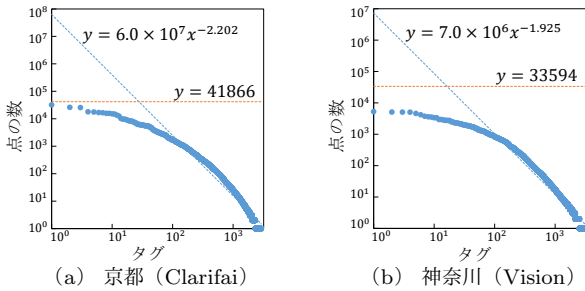


図 4: タグごとの保持する点の数

の種類数が多く、多様な点を含んでいるといえる。

次に、4.1 節で予想したタグごとの保持される点の数について確認を行った結果、各タグとそのタグを保持する点の数の関係は図 4 のようになった。図 4 の y 軸は各タグを保持する点の数を表し、 x 軸は保持する点の数で降順に並べたタグの ID を表す両対数グラフである。青い点は実際の点の数を、青い点線はデータを対数変換して直線近似した関数、橙の線はデータセットの写真数を表す。図 4 から確かに 4.1 節で予想したように、多くの点が共通して保持するタグと、一部の点のみが保持するタグが存在することがわかる。この傾向は京都 (Vision) や神奈川 (Clarifai) でもみられた。

5.2 ROI の抽出結果

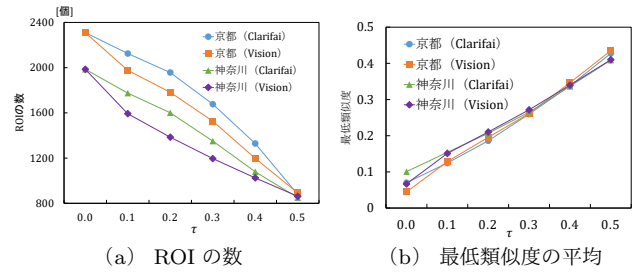
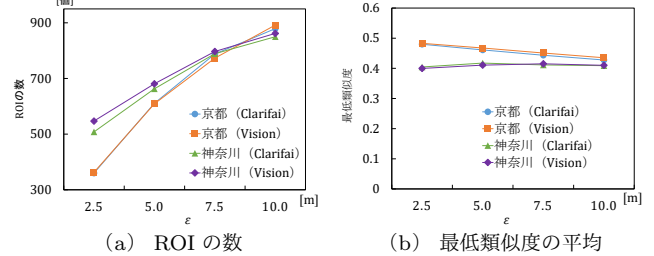
4. で提案した ROI の抽出方法の評価を行うため、各データセットに対し、SimDBSCAN による ROI の抽出を行った。このとき、パラメータの変化により、抽出できる ROI の数などの変化を観察した。また、実際に抽出した ROI の例を示す。

5.2.1 パラメータの影響

距離閾値 ε を 10m に固定し、類似度閾値 τ の値を変化させたときに抽出できる ROI の数の変化を計測した。また 4.2.1 項で述べた、SimDBSCAN が ROI 中の 2 点間の類似度の大きさを保証しない問題について検証するため、各 ROI 中に含まれる点間の最低類似度の平均の変化を計測した。同様に、類似度閾値 τ を 0.5 に固定し、距離閾値 ε を変化させ、計測を行った。

図 5 から全てのデータセットで、類似度閾値 τ が大きくなるにつれ、抽出できる ROI の数が減り、最低類似度の平均が上昇することが観察された。これは類似度閾値 τ が大きくなることにより、直接類似到達可能な点が少なくなり、より精練された ROI のみが抽出されるためである。

一方、図 6 では距離閾値 ε が大きくなるにつれ、抽出できる ROI の数が増加している。これは距離閾値 ε が大きくなることにより、直接類似到達可能な点が多くなるためである。一方で、最低類似度の平均は京都ではゆるやかに減少しており、神奈川ではほぼ横ばいである。そのため、距離閾値 ε を大きくし、ROI に多くの点が含まれるようにしても、最低類似度には殆ど

図 5: 類似度閾値 τ の影響図 6: 距離閾値 ε の影響

影響がないことがわかる。以上のことから 4.2.1 項で述べた、SimDBSCAN の ROI 内の任意の 2 点間の最小類似度が小さくなる恐れについて、実データセット上では問題ないといえる。

5.2.2 ROI の例

パラメータを $(\varepsilon, MinPts) = (10, 3)$ に設定し、類似度閾値 τ を 0, 0.3, 0.5 と変化させたときの実行結果の変化を観察する。

図 7 は、神奈川 (Vision) に提案手法を用いたときに長谷寺付近で得られた ROI の分布を表し、表 4 は図 7 内のいくつかの ROI のトピックである。図 7 の各図内の色付きの多角形は ROI の凸包を表しており、一部の ROI は重なって存在する。

類似度を考慮せずに ROI の抽出を行った図 7 (a) 中では、ROI r_1 のように非常に大きな ROI が長谷寺を覆っている。その一方、類似度を考慮して ROI の抽出を行った図 7 (b) では、 r_1 がいくつかの ROI にその被写体ごとに分かれている。例えば、ROI r_2 に含まれる写真の多くは紫陽花を写したものであり、ROI r_3 に含まれる写真の多くは観音堂を写したものである。しかし、 r_2 は石造釈迦如来坐像の写真など紫陽花以外の写真を含んでいる。これに対し、より厳しい条件で類似度を考慮している図 7 (c) では、ROI r_4 に含まれる写真は紫陽花の写真のみであるなど、より正確に被写体別に分かれた ROI を得られる。また、ROI r_5 では紫陽花が撮影され、ROI r_6 では電車が撮影されていた。これらの領域は有名なスポットではないため、未知の ROI といえる。以上のように、長谷寺などの有名な既知の ROI 内では、より細かな ROI が発見できること、 r_5 や r_6 のような未知の ROI を発見できることが確認できた。

5.3 パターンの抽出結果

まず、SimDBSCAN のパラメータにより、得られるパターン数などの変化を観察する。その後、各分析ツールによって得られたパターンの違いを観察する。

5.3.1 パラメータの影響

距離閾値 ε を 10m に固定し、類似度閾値 τ の値を変化させたときに得られるパターンの数とその長さを計測した。ただし、1 人のユーザの軌跡のみが要因となるパターンは、複数のユー

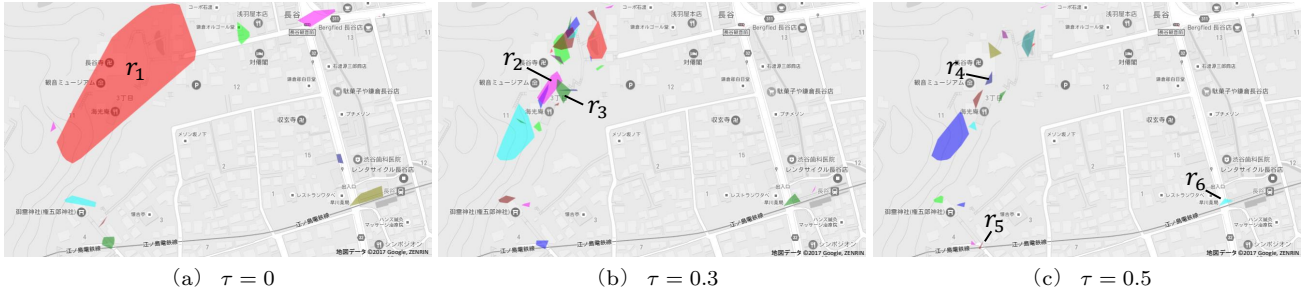


図 7: 長谷寺付近の ROI の分布

表 4: 図 7 中の ROI のトピック

ROI	トピック	score	ROI	トピック	score
r_1	flower	398.8	r_4	hydrangea	18.6
	plant	361.9		nature	16.0
	botany	354.1		land plant	15.9
r_2	garden	45.4	r_5	hydrangea	9.7
	botany	38.5		hydrangeaceae	8.9
	flower	36.6		flora	6.8
r_3	chinese architecture	26.8	r_6	tram	19.2
	temple	23.5		land vehicle	17.7
	historic site	20.6		electricity	13.5

表 5: 閾値のパターン数への影響

(a) 類似度閾値 τ

類似度 閾値 τ	パターンの長さ				
	2	3	4	5	6
0.0	1056	323	83	15	2
0.1	324	81	11	0	0
0.2	345	96	19	1	0
0.3	191	35	4	0	0
0.4	102	25	4	0	0
0.5	34	3	0	0	0

(b) 距離閾値 ε

距離 閾値 ε	パターンの長さ		
	2	3	4
2.5	2	0	0
5.0	32	0	0
7.5	139	20	1
10.0	191	35	4

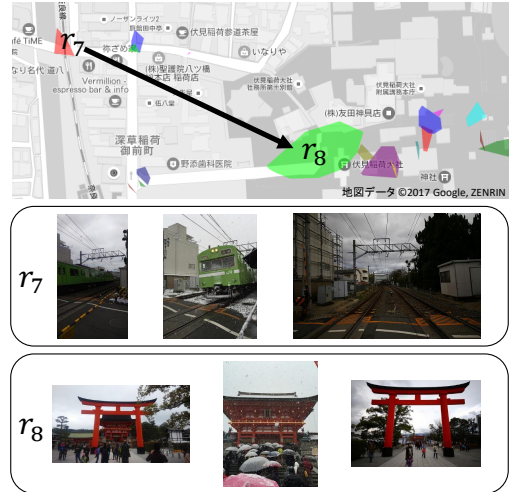


図 8: 京都 (Vision) のパターン例

ザの軌跡が要因となるパターンより、価値が低いと考えられるため除外した。同様に、類似度閾値 τ を 0.3 に固定し、距離閾値 ε を変化させたときについても計測を行った。

表 5 (a), (b) は京都 (Vision) から得られたパターン数とその長さを表す。これらの表から得られるパターン数は類似度閾値 τ が大きいほど少なく、距離閾値 ε が大きいほど多くなり、その影響を大きく受けることがわかる。この傾向は他のデータセットにおいてもみられた。これは閾値が変化することにより、抽出できる ROI 数が変化することに起因する。

5.3.2 分析ツールの違い

パラメータを $(MinPts, \varepsilon, \tau, MinSup) = (3, 10, 0.5, 3)$ に設定し、京都 (Clarifai), 京都 (Vision) に対して提案手法を実行した際に得られるパターンの違いを観察した。

図 8 は京都 (Vision) でのみ得られたパターン例である。パターン中の各 ROI の top-3 のトピックは表 6 のとおりである。ROI r_7 は JR の奈良線の電車や稲荷駅が撮影されている ROI である。一方、ROI r_8 は比較的広範囲の ROI となっているが、ROI 中の写真は全て伏見稲荷大社の楼門や本殿を撮影しており、

妥当な ROI である。京都 (Vision) では伏見稲荷大社付近に別のパターンを複数発見することができたが、京都 (Clarifai) では 1 つのパターンしか発見できなかった。

一方、図 9 は京都 (Clarifai) でのみ得られたパターン例 (ROI r_{10} は円中の小さな三角形) であり、二条城内を移動するパターンである。パターン中の各 ROI の top-3 のトピックは表 7 のとおりである。このパターン中の ROI は各々異なる ROI と領域が重なっている。ここで、 r_{10} の撮影対象が二の丸御殿であるのに対し、 r'_{10} の撮影対象は二の丸庭園であり、その撮影対象は大きく異なっている。また、ROI r_{11} と ROI r'_{11} についても同様に r_{11} が建物、 r'_{11} が庭園と撮影対象が大きく異なっている。以上のことから、このパターンを持つユーザは、特に建築物に興味を持ったことが示唆される。各 ROI のトピックもそれを指し示す。京都 (Clarifai) では二条城内に別のパターンを複数発見することができたが、京都 (Vision) では 1 つのパターンしか発見できなかった。

このように分析ツールによって、発見できるパターンが変化することがわかった。これはツールによって、識別できる被写体の得手不得手があり、求められるタグが変化し、タグ間の類似度がその影響を受けるためと考えられる。また、京都 (Clarifai) から得られるパターンのように、ユーザの嗜好を推定できるパターンが存在することがわかった。

6. まとめ

本稿では、重み付きタグ集合を持つ意味的な移動軌跡からの

表 6: 図 8 中の ROI のトピック

ROI	トピック	score
r_7	rail transport	32.2
	train	30.9
	track	28.5
r_8	chinese architecture	247.9
	shinto shrine	179.2
	temple	166.4

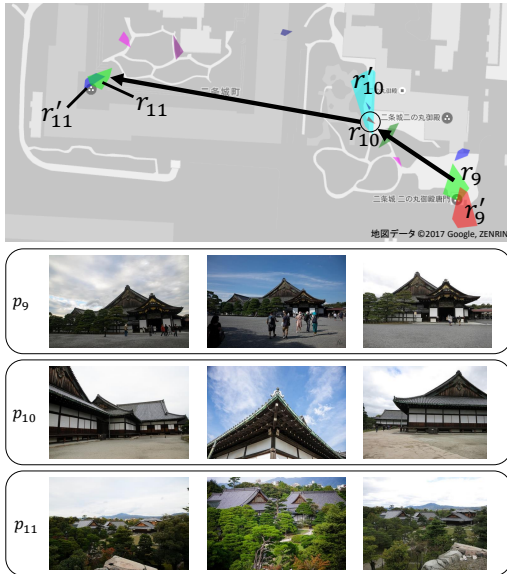


図 9: 京都 (Clarifai) のパターン例

表 7: 図 9 中の ROI のトピック

ROI	トピック	score	ROI	トピック	score
r_9	castle	45.1	r'_9	castle	117.5
	marquee	43.8		ancient	111.4
	temple	43.4		temple	108.9
r_{10}	eaves	20.4	r'_{10}	pool	71.5
	expression	12.6		stone	61.6
	construction	12.6		lake	60.7
r_{11}	roof	42.5	r'_{11}	lake	25.0
	mountain	40.0		river	23.9
	home	36.2		bridge	19.0

意味的な軌跡パターンの抽出手法を提案した。まず、同様の点が多く存在する ROI を明らかにするため、タグ集合間の類似度を定義し、類似度と密度に基づくクラスタリング手法を提案した。提案手法の有効性を確認するために、写真の分析結果を用いた評価実験では、提案手法により、被写体別に ROI を分類できることを確認した。更に得られたパターンは、ユーザの嗜好を示唆する場合があることを確認できた。

本研究ではデータセットに対して、一律のパラメータで分析を行った。しかし、現実には都市部と郊外など、多くの点が存在する地点とそうでない地点が存在する。このばらつきを考慮した適応的なクラスタリング手法については今後の課題としたい。また他の課題として、より大規模なデータセットや、異なる性質を持つデータセットに対する評価実験が挙げられる。

謝辞 本研究の一部は、科研費 (16H01722, 26540043) と

び文部科学省「実社会ビッグデータ利活用のためのデータ統合・解析技術の研究開発」による。

文 献

- [1] Clarifai API. <https://www.clarifai.com/api>.
- [2] Flickr. <https://www.flickr.com/>.
- [3] Google cloud vision api. <https://cloud.google.com/vision/>.
- [4] Swarm. <https://www.swarmapp.com/>.
- [5] Twitter. <https://twitter.com/>.
- [6] L. O. Alvares, V. Bogorny, B. Kuijpers, B. Moelans, J. A. Fern, E. Macedo, and A. T. Palma. Towards semantic trajectory knowledge discovery. *Data Mining and Knowledge Discovery*, 2007.
- [7] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [8] T. L. C. da Silva, K. Zeitouni, J. A. de Macêdo, and M. A. Casanova. A framework for online mobility pattern discovery from trajectory data streams. In *International Conference on Mobile Data Management (MDM 2016)*, volume 1, pages 365–368, 2016.
- [9] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.
- [10] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *KDD*, pages 330–339, 2007.
- [11] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *ICDE*, pages 215–224, 2001.
- [12] C.-C. Hung, W.-C. Peng, and W.-C. Lee. Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. *The VLDB Journal*, 24(2):169–192, 2015.
- [13] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen. Discovery of convoys in trajectory databases. *Proceedings of the VLDB Endowment*, 1(1):1068–1080, 2008.
- [14] R. Jiang, J. Zhao, T. Dong, Y. Ishikawa, C. Xiao, and Y. Sasaki. A density-based approach for mining movement patterns from semantic trajectories. In *TENCON 2015-2015 IEEE Region 10 Conference*, pages 1–6. IEEE, 2015.
- [15] Y. Kim, J. Han, and C. Yuan. Toptrac: Topical trajectory pattern mining. In *KDD*, pages 587–596, 2015.
- [16] P. Laube and S. Imfeld. Analyzing relative motion within groups of trackable moving point objects. In *International Conference on Geographic Information Science*, pages 132–144. Springer, 2002.
- [17] Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. *Proceedings of the VLDB Endowment*, 3(1-2):723–734, 2010.
- [18] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to Information Retrieval*, volume 1. Cambridge university press Cambridge, 2008.
- [19] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis, Y. Theodoridis, and Z. Yan. Semantic trajectories modeling and analysis. *ACM Computing Surveys*, 45(4), 2013.
- [20] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng. Semantic trajectory mining for location prediction. In *ACM SIGSPATIAL GIS*, pages 34–43, 2011.
- [21] C. Zhang, J. Han, L. Shou, J. Lu, and T. La Porta. Splitter: Mining fine-grained sequential patterns in semantic trajectories. *PVLDB*, 7(9):769–780, 2014.
- [22] K. Zheng, Y. Zheng, N. J. Yuan, and S. Shang. On discovery of gathering patterns from trajectories. In *ICDE*, pages 242–253, 2013.