

ソーシャルストリームからのイベント検出とユーザ位置推定の統合

山口 祐人^{†,††} 伊川 洋平^{†††} 天笠 俊之^{††††} 北川 博之^{††††}

† 筑波大学大学院システム情報工学研究科 〒 305-8573 茨城県つくば市天王台 1-1-1

†† 日本学術振興会特別研究員 DC1 〒 102-0083 東京都千代田区麹町 5-3-1

††† 日本アイ・ビー・エム株式会社 東京基礎研究所 〒 135-8511 東京都江東区豊洲 5-6-52

†††† 筑波大学システム情報系 〒 305-8573 茨城県つくば市天王台 1-1-1

E-mail: †yutoymgc@kde.cs.tsukuba.ac.jp, ††yikawa@jp.ibm.com, †††{amagasa,kitagawa}@cs.tsukuba.ac.jp

あらまし ソーシャルメディアの普及に伴い、多くのユーザが時々刻々と大量の情報を投稿し、ソーシャルストリームと呼ばれる即時性の高い情報源を形成している。ソーシャルストリームを用いると地震や火事などの地理的な局所性を持つ実世界イベントを検出することが可能であり、重要視されている。しかし、ソーシャルストリームからのイベント検出にはユーザの位置情報が必要であるが、多くのユーザは自らの位置情報を公開していない。そこで本研究では、イベント検出と位置推定を統合し、それらを継続的に行う手法を提案する。提案手法では、多くのイベントが検出されれば多くのユーザの位置推定が可能であり、多くのユーザの位置情報が分かれば多くのイベントの検出が可能である。実施した評価実験により、妥当なイベントが検出され、既存の位置推定手法より高精度な位置推定が可能であることが示された。

キーワード イベント検出, 位置推定, ソーシャルストリーム, ソーシャルメディア, 時空間データマイニング

Yuto YAMAGUCHI^{†,††}, Yohei IKAWA^{†††}, Toshiyuki AMAGASA^{††††}, and Hiroyuki

KITAGAWA^{††††}

† Graduate School of Systems and Information Engineering, University of Tsukuba
Tennodai 1-1-1, Tsukuba, Ibaraki

†† Research Fellow of the Japan Society for the Promotion of Science
Kojimachi 5-3-1, Chiyoda-ku, Tokyo

††† IBM Research-Tokyo, IBM Japan Ltd. Toyosu 5-6-52, Koto-ku, Tokyo

†††† Faculty of Engineering, Information and Systems, University of Tsukuba
Tennodai 1-1-1, Tsukuba, Ibaraki

E-mail: †yutoymgc@kde.cs.tsukuba.ac.jp, ††yikawa@jp.ibm.com, †††{amagasa,kitagawa}@cs.tsukuba.ac.jp

1. はじめに

ウェブ上では様々なソーシャルメディアが登場し、多くの人々がそれらを利用して多様な情報を発信している。例えば、写真や動画を投稿するメディアや、自らの意見や主張などをテキストとして投稿するメディア、レストランや本などのレビューを投稿するメディアなどがある。その中でも、マイクロブログなどの即時性の高いソーシャルメディアからは、それぞれのユーザがテキストなどを投稿するという形で時々刻々と大量の情報が発信されている。本稿では、ソーシャルメディアから時々刻々とストリームのように発信される、即時性の高い情報源のことをソーシャルストリームと呼ぶ。

即時性の高い情報が時々刻々と発信されるという性質を生かして、ソーシャルストリームを用いて実世界のイベントを

検出するという研究が多く行われている [2], [5], [7]~[9], [11]~[13], [16]。全世界のユーザがそれぞれの状況や意見を自由に投稿するため、ソーシャルストリームをモニタリングすることで、実世界の様子をある程度捉えることが出来る。例えば、東京で地震が発生した時には、東京にいるユーザが一斉に「地震だ」のような投稿をするため、ソーシャルストリームをモニタリングしていれば東京で地震が発生したということを検出することが可能である。ソーシャルストリームを用いた実世界イベントの検出には以下の三つの利点がある。

- 人々の投稿を用いるため、様々な種類のイベント（例えば、地震などの災害、歌手のゲリラライブ、通り魔などの犯罪）の検出が可能である。
- 人々の投稿の内容を見ることにより、イベントの検出だけでなく、そのイベントに対する意見やより詳細な状況など

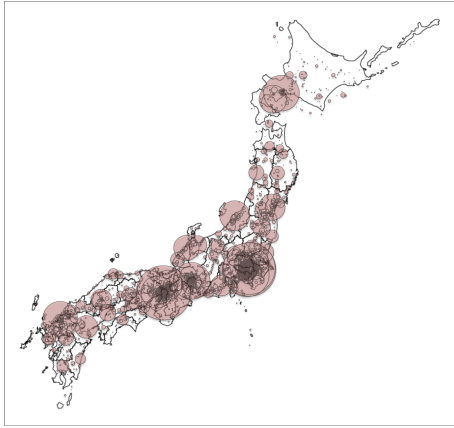


図 1 ある期間に投稿されたツイートの地理的な分布. 円の大きさは同じ位置から投稿されたツイートの数を示す. 東京や大阪などの大都市からの投稿が多いことが分かる.

を把握することが可能である.

- 新聞やテレビ, ブログなどの従来のメディアとは異なり, ソーシャルストリームは即時性の高い情報源であるため, イベントの発生を直ちに検出することが可能である.

本研究では, 選挙やクリスマスのような地理的局所性を持たないイベントではなく, 地震などの地理的局所性のある**ローカルイベント**を対象とする.

ソーシャルストリームを用いたローカルイベントの検出には, それぞれの投稿がどこから行われたかという位置情報が必要である. しかし, それぞれのメディアにおいて位置情報を公開する機能があまり普及していないことや, プライバシーの観点などから多くのユーザは自らの位置情報を公開していない. Cheng ら [4] によると, Twitter ユーザの約 76 %が居住地情報(テキスト情報)を公開しておらず, またすべての投稿のうち約 99.5 %には位置情報(GPS タグ)が付与されていないことが明らかになっている. 本研究で実施した予備調査によると, やはり同様に Twitter ユーザの約 75 %が居住地情報を公開していないことが示された(詳細は 5.2 節). また, Backstorm ら [1] は, facebook においては約 94%ものユーザが居住地情報を公開していないと報告している. したがって, ソーシャルストリームを用いたイベント検出を行うには, ユーザの位置情報を推定し, それぞれの投稿がどこから行われたものかを知ることが必要である. ユーザの位置推定を行う研究は, 投稿されたテキストの内容を用いるもの [3], [4], [10] や, ソーシャルグラフを用いるもの [1], [6], [10], [14] など, 様々なものが提案されている.

本研究では, ソーシャルストリームを用いたローカルイベントの検出とユーザの位置推定を統合的かつ継続的に行う手法を提案する. 提案手法は以下のアイデアに基づいている.

(1) 類似した内容を持つ投稿がある地域から局所的に多く投稿された時, その地域で何らかのローカルイベントが発生している可能性が高い.

(2) ある地域でローカルイベントが検出された時, そのイベントに関する投稿をしたユーザはその地域にいる可能性が高い.

一つ目のアイデアは多くのイベント検出手法に採用されている考え方であるが, 二つ目のアイデアに基づく位置推定手法は筆者の知る限りこれまで提案されていない. 図 1 はある期間に

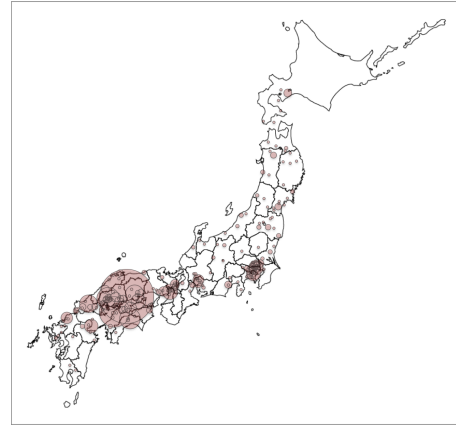


図 2 広島で地震が発生した時に投稿された「地震」という単語を含むツイートの地理的な分布. 円の大きさは同じ位置から投稿されたツイート数を示す. 普段の地理的な分布(図 1)とは異なり, 広島付近からの投稿が多くなっていることが分かる. 一方で, 東京や大阪などからの投稿もある程度数を保っている事が分かる.

Twitter に投稿されたツイートの地理的な分布^(注1)を表し, 図 2 は広島で地震が起きた際に投稿された「地震」という単語を含むツイートの地理的な分布を表す. 円の大きさは同じ位置から投稿されたツイート数を表す. これらの図によると, あるローカルイベントが発生した時にそれに関する投稿をしたユーザは, その付近に位置することが多いことが分かる. すなわち, 既知であるユーザの位置情報を用いてローカルイベントの検出をすることが可能であり, 検出されたローカルイベントの位置情報を用いて未知のユーザの位置情報を推定することが可能であると考えられる.

本研究の貢献は以下に示す三点である.

(1) イベント検出とユーザ位置推定を統合的に行う手法を提案する.

(2) ユーザの位置情報を逐次的に推定, 更新する手法を提案する.

(3) Twitter データを用いた評価実験により, 提案手法のイベント検出及びユーザ位置推定における有効性を示す.

提案手法では, 多くの妥当なイベントが検出されれば多くのユーザの位置を精度よく推定することが可能であり, 多くのユーザの位置情報が分かれば多くの妥当なイベントの検出が可能である. 本研究では, ユーザの位置情報として**居住地**の推定を目的とする. これは, ユーザの投稿の多くはそのユーザの居住地付近から行われるという仮定に基づいている. また, 提案手法はイベントを検出する度にユーザの位置情報を推定し, 逐次更新することが出来る. これにより, 引越などによりユーザの居住地が変わった場合にも対応することが可能である.

5. 節で議論する評価実験によると, 提案手法によるユーザ位置推定の精度は約 76 %であり, これは既存の手法の精度と比較すると約 33 %から約 121 %の向上である. また, どの程度の規模のイベントがどれだけ発生したかにも依存するが, 5. 節の評価実験によると, 提案手法を二週間分の Twitter ソーシャルストリームに適用すると約 20,000 のユーザの位置推定が可能であった. さらに, 位置情報が既知であるユーザ数が増加するほど多くの妥当なイベントが検出されることが明らかになった.

(注1): ツイートを投稿したユーザの居住地の地理的な分布

本稿の以降の構成は以下のとおりである。まず、2. 節でイベントを検出する関連研究及びユーザ位置推定を行う関連研究を概観し、3. 節で本研究で扱う問題とそれに関する用語を定義する。次に、4. 節で本研究の提案手法について述べ、5. 節で提案手法の有効性を、イベント検出の妥当性、ユーザ位置推定の精度及びどれだけの数のユーザの位置を推定できたかという効率性の観点から検証する。最後に、6. 節で本稿をまとめる。

2. 関連研究

本節では、ソーシャルストリームからのイベント検出及びソーシャルメディアにおけるユーザ位置推定に関する研究について概観する。

2.1 イベント検出

Sakaki ら [15] は、Twitter におけるソーシャルストリームを用いて地震や台風などのイベントを検出する手法を提案した。Sakaki らの提案手法では、パーティクルフィルタを用いて台風などの移動するイベントの移動軌跡を推定した。Lee ら [8] は、DBSCAN を用いてツイートをクラスタリングし、ツイートを投稿したユーザのタイムゾーンからイベントの位置を推定した。Lappas ら [7] は、地理的に分散した複数のストリーム情報源からデータが流れてくる状況において、時間的、地理的にバーストしている単語を検出する手法を提案した。Ritter ら [13] は、様々な種類のイベントを検出し、そのカテゴリ分けをする手法を提案した。Watanabe ら [16] は、GPS による位置情報タグの付けられたツイートを用いて地理的な粒度の小さいイベントを検出する手法を提案した。これらの研究はイベント検出のみに注目した研究であり、ユーザ位置推定を対象としていないため、本研究とは異なる。

2.2 ユーザ位置推定

ユーザ位置推定を行う手法として、ユーザの投稿を用いるものが提案されている。Cheng ら [4] は、Twitter において地理的な局所性を持つ単語を用いてユーザ位置推定を行う手法を提案した。Cheng らの手法では、まずツイートに含まれる単語とそのツイートを投稿したユーザのロケーションとを用いて、単語の地理的な分布を作る。次に、その地理的な分布から、ある地域から偏って投稿されている単語を抽出する。例えば、*rockets* という単語は、アメリカのヒューストン付近から多く投稿されると報告されている。そして、抽出された単語が推定したいユーザのツイートに含まれていれば、それらの単語の地理的な分布からユーザのロケーションを推定する。Cheng らは、ユーザが多くのツイートを投稿していれば、比較的高い精度での推定が可能であると主張している。Chang ら [3] は、混合正規分布を用いて単語の分布をモデル化し、ユーザ位置推定を行う手法を提案した。これらの手法は定常的な単語の地理的な分布を扱っており、時間に依存するイベントの発生を考慮していないため、本研究とは異なる。

また、ユーザ間の関係を表すソーシャルグラフを用いたユーザ位置推定も提案されている。Clodoveu ら [6] は、友人同士であるユーザのロケーションは近い可能性が高いとし、ユーザ位置推定手法を提案した。具体的には、ある Twitter ユーザ u のロケーションを推定するとき、 u と相互にフォローしているユーザ群のロケーションを調べ、最も多いロケーションを u のロケーションであると推定した。Clodoveu らは、友人数が少なすぎる場合には手がかりとなる情報が少なく、また友人数が多すぎる場合にはそのユーザは有名人やボットであり、ロケーション情報は意味をなさないため、友人数が 20 から 200

程度の場合にうまくいくと主張している。Backstorm ら [1] は Facebook のデータを用いて同様にソーシャルグラフから位置推定を行う手法を提案した。Sadilek ら [14] は、Twitter における GPS による移動軌跡と、ユーザ間の友人関係は互いに関係するとして、ユーザの移動軌跡の推定とユーザ間の友人関係を予測するリンク予測という二つのタスクを行う手法を提案した。Sadilek らは、たとえユーザが現在の位置情報を公開していなくても、一緒にいるユーザがそれを公開してしまうことで自身の位置情報が知れ渡ってしまうと主張している。Li ら [10] は、ユーザは複数のロケーションを持つとし、ツイートに含まれる地名とソーシャルグラフを用いてそれらのロケーションを推定する手法を提案した。これらの手法はソーシャルグラフに基づくものであるため、本研究とは異なる。

3. 問題定義

本節では、本研究で用いる用語の定義及びイベント検出とユーザ位置推定の問題定義を行う。

それぞれの投稿におけるタイムスタンプ s 、テキスト t 、投稿された位置 l (以下、ロケーション) の三つ組を **ポスト** $p = (s, t, l)$ と呼び、ポストの列を **ソーシャルストリーム** $SS = (p_1, p_2, \dots)$ と定義する。ここで、ロケーションが未知の場合は $l = NULL$ とする。すべてのロケーション l はあらかじめ与えられたロケーション集合 L に属するものとする。また、**イベント** e とは、次のような性質を持つポストの集合と定義する。すべての $p_i \in e$ について、タイムスタンプ s_i が時刻の区間として定義された同一の **タイムウィンドウ** W_j に属し、互いのテキスト t_i が十分に類似し、かつ互いのロケーション l_i 間の距離が互いに十分に近い。例えば、広島付近から「地震」という単語を含むポストが短期間に多く投稿された場合、それらをイベントとみなすことが出来ると考えられる。このとき、イベント検出を以下のように定義する。

[イベント検出] 与えられたタイムウィンドウ W_j に属すポストの集合からイベントの集合 $E_j = \{e_1, e_2, \dots, e_n\}$ を検出する。一つもイベントが検出されない場合は E_j は空集合となる。ただし、一つのポストは必ず一つ以下のイベントに属する。

ソーシャルストリームに投稿をし得るユーザ $u \in U$ の位置情報は離散確率分布 $P_u(l)$ として定義する。これをユーザの **ロケーション分布** と呼ぶ。 u がポストを投稿する際に、最も確率値の大きいロケーション $\hat{l}_u = \operatorname{argmax}_l P_u(l)$ が実現値として与えられるものとする。このとき、ユーザ位置推定を以下のように定義する。

[ユーザ位置推定] 検出されたイベント e が与えられた時、 $\hat{l}_u = \operatorname{argmax}_l P_u(l)$ がユーザ u の実際のロケーション l_u となるように、 e を用いてユーザのロケーション分布 $P_u(l)$ を更新する。次のイベント検出にロケーション分布を用いるため、一つのイベントが検出されるたびにロケーション分布を逐次更新する。

4. 提案手法

本節では提案手法について説明する。提案手法はソーシャルストリームを入力として受け取り続け、継続的にイベント検出とユーザ位置推定を行う。提案手法を用いるための初期条件として、あらかじめある程度のユーザのロケーション分布が既知であることが必要となる。

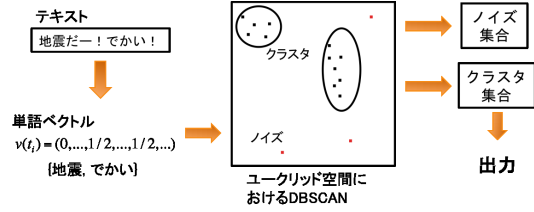


図3 DBSCANによるポストのクラスタリング. DBSCANによりノイズとみなされたポストは破棄し、クラスタの集合のみを出力する。

4.1 イベント検出

提案手法はまずこれまでに既知となっているユーザーのロケーションとソーシャルストリームを用いてイベント検出を行う。イベント検出はテキストの内容を用いて類似するポストをクラスタリングする Content Clustering と、それぞれのクラスタに含まれるポストのロケーションが地理的な局所性を持つか判別し、局所性を持たないクラスタをフィルタリングする Spatial Filtering というフェーズに分かれる。以下ではそれぞれのフェーズについての説明を与える。

4.1.1 Content Clustering

ここでは、入力としてあるタイムウィンドウ W_j に属するポストの集合が与えられるものとする。タイムウィンドウの幅 $WindowSize$ は、例えば10分というように時間で与えるパラメータである。それぞれのポスト p のテキスト t を、bag-of-words モデルを用いて単語ベクトル $\mathbf{v}(t)$ で表す。ベクトルの次元数は対象とする語彙に含まれる単語の種類数であり、ベクトルの各次元には対応する単語が t に含まれていれば $\frac{1}{|T|}$ 、そうでなければ0が入る。ここで、 T は t に含まれる単語の集合である。

単語ベクトル $\mathbf{v}(t)$ を用いて、DBSCANによるポストのクラスタリングを行う。距離関数として単語ベクトル間のユークリッド距離 $dist_t(\mathbf{v}(t_i), \mathbf{v}(t_j))$ を用いる。また、DBSCANはパラメータとして $MinPts$ と Eps を取る。これらはそれぞれクラスタとして許す最小のポスト数と、類似するポストとして許す最大の距離を表す。DBSCANを用いる理由は密度の大きい部分のみがクラスタとして出力され、密度の小さい部分にあるポストはノイズとして捨てられるためである(図3)。これにより、短期間に類似したポストが多く投稿された場合にのみイベントとして扱われる。

Content Clustering の出力はDBSCANによって出力されたクラスタの集合 $C_j = \{c_1, c_2, \dots, c_n\}$ である。入力として与えられたタイムウィンドウ W_j に対して一つのクラスタ集合 C_j が出力される。

4.1.2 Spatial Filtering

ここでは、入力として Content Clustering の出力であるクラスタ集合 C_j が与えられるものとする。それぞれのクラスタ $c \in C_j$ に対して地理的な局所性を持つかどうか判別し、局所性を持つもののみをイベント e として出力する。ポスト $p \in c$ のロケーションを緯度、経度のペアとして $l = (lat, long)$ のように表す^(注2)。

クラスタの地理的な局所性を図る指標として、dispersion という指標を次のように導入する。

$$dispersion(c) = \frac{1}{|c|} \sum_{p_i \in c} dist_l(l_i, m_c) \quad (1)$$

(注2)：実際にはジオコーディングなどを用いて緯度経度情報に変換する。

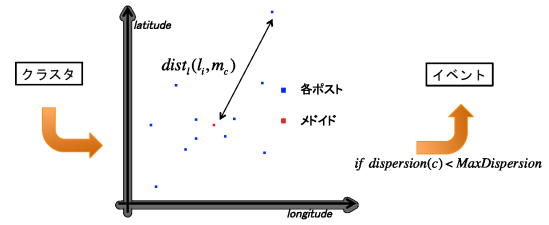


図4 Spatial Filteringによる地理的な局所性の判別. 算出された $dispersion(c)$ がパラメータ $MaxDispersion$ を超えなければ、クラスタ c をイベントとして検出する。

ただし、 $dist_l(\cdot, \cdot)$ は二点間のユークリッド距離を表し、 m_c はクラスタ c の中心点として、 c に含まれるポスト集合のメドイドを表す。メドイドとは、ポスト集合のうち、他のすべてのポストとの距離の合計を最小にするようなポストのことである。 $dispersion(c)$ が小さいほどクラスタ c の地理的な局所性が高いということになる。

算出した $dispersion(c)$ がパラメータとして与えられた $MaxDispersion$ を超えないクラスタ c をイベントとして検出し、イベントの集合 $E_j = \{e_1, e_2, \dots, e_m\}$ を出力する(図4)。

4.2 ユーザ位置推定

前節のイベント検出によってイベント e が検出されるたびに、それを用いてユーザーのロケーション分布を更新する。直感的には、あるイベントが発生したときそのイベントに関する投稿をしたユーザーは、イベントが発生した場所の付近にいた可能性が高い。提案手法はこの考え方に基づき、イベント e に含まれるポスト p を投稿したユーザーのロケーション分布を、イベントが発生した場所付近の確率値が大きくなるように更新する。これにより、推定されたユーザーのロケーション $\hat{l}_u$ は、実際のロケーション l_u に近づいていくと考えられる。以下ではまずユーザー位置推定を行うためのモデルを提案し、次に提案したモデルを用いた分布の更新について説明する。

4.2.1 ロケーションモデル

取りうるロケーションの集合 L について、ユーザーのロケーション分布 $P_u(l)$ を以下を満たすパラメータベクトル $\theta_u = (\theta_{u,1}, \dots, \theta_{u,|L|})$ で表す。

$$P_u(l) = \theta_{u,l}, \quad \sum_{l \in L} \theta_{u,l} = 1, \quad 0 \leq \theta_{u,l} \leq 1 \quad (2)$$

提案手法では、それぞれのユーザー u のロケーション分布、すなわち u のロケーションが l である確率を表すパラメータベクトル θ_u を推定する。ユーザーのロケーション $\hat{l}_u$ は、推定されたパラメータベクトル θ_u を用いて以下のように与えられる。

$$\hat{l}_u = \arg \max_{l \in L} \theta_{u,l} \quad (3)$$

パラメータベクトル θ_u は MAP 推定

$$\theta_u = \arg \max_{\theta} P(\theta | E_u) \quad (4)$$

によって得られる。ここで、 E_u は u が e に含まれるポスト p を投稿したことを $mention(u, e)$ と書くと、以下で与えられる。

$$E_u = \{e \in E \mid mention(u, e)\} \quad (5)$$

ただし、 E はこれまでに検出されたすべてのイベントの集合である。

4.2.2 パラメータベクトルの MAP 推定

MAP 推定は、ベイズの定理により以下のように書ける。

$$\arg \max_{\theta} P(\theta|E_u) = \arg \max_{\theta} P(E_u|\theta)P(\theta) \quad (6)$$

さらに、検出されたイベントは i.i.d であると仮定すると、イベント集合 E_u の尤度関数 $P(E_u|\theta)$ は次を満たす。

$$P(E_u|\theta) = \prod_{e \in E_u} P(e|\theta) \quad (7)$$

ここで、 $P(e|\theta)$ はパラメータ θ に従ってイベント e が生成される尤度を表す。以上より、求めるパラメータの MAP 推定値は以下の式により得られる。

$$\theta_u = \arg \max_{\theta} P(\theta) \prod_{e \in E_u} P(e|\theta) \quad (8)$$

以下では、イベント e の尤度関数 $P(e|\theta)$ 及びパラメータの事前分布 $P(\theta)$ を求める。

イベント e を、 e に属すポストのロケーションの集合であると考え、bag-of-words モデルと同様の考え方により、ベクトル $\mathbf{v}(e) = (n_{e,1}, n_{e,2}, \dots, n_{e,|L|})$ で表現する。ここで、 $n_{e,k}$ はポスト $p \in e$ のうち、ロケーションが l_k であるものの数を表す。すると、イベント e を表すロケーションベクトル $\mathbf{v}(e)$ はパラメータ θ の多項分布に従って次のように生成されると考えられる。

$$P(e|\theta) = \text{Multi}(\mathbf{v}(e); \theta) = \frac{N_e!}{\prod_k n_{e,k}!} \prod_k \theta_k^{n_{e,k}} \quad (9)$$

ただし、 $N_e = \sum_k n_{e,k}$ である。

多項分布の共役事前分布はディリクレ分布であるため、ここではパラメータ θ の事前分布としてディリクレ分布を採用する。

$$P(\theta) = \text{Dir}(\theta; \alpha) = \frac{\Gamma(A)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1} \quad (10)$$

ただし、 $\Gamma(\cdot)$ はガンマ関数、 α はディリクレ分布のパラメータ、 $A = \sum_k \alpha_k$ である。

ここまでで求めた尤度関数及び事前分布を用いると、MAP 推定は以下のように求められる^(注3)。

$$\begin{aligned} \theta_u &= \arg \max_{\theta} P(\theta) \prod_{e \in E_u} P(e|\theta) \\ &= \arg \max_{\theta} \text{Dir}(\theta; \alpha) \prod_{e \in E_u} \text{Multi}(\mathbf{v}(e); \theta) \\ &= \arg \max_{\theta} \text{Dir}(\theta; \alpha + \sum_{e \in E_u} \mathbf{v}(e)) \end{aligned} \quad (11)$$

これを解くと

$$\theta_{u,k} = \frac{\sum_{e \in E_u} n_{e,k} + \alpha_k - 1}{\sum_{e \in E_u} N_e + A + |L|} \quad (12)$$

となり、パラメータベクトル θ_u が得られる。

4.2.3 パラメータベクトルの逐次更新

新たなイベント e' が検出された時、パラメータベクトルを更新することが可能である。これまでの事後分布 $P(\theta|E_u)$ を事前分布 $P(\theta)$ とし、以下のように再度 MAP 推定を行う。

$$\theta'_u = \arg \max_{\theta} P(\theta|e') = \arg \max_{\theta} P(e'|\theta)P(\theta) \quad (13)$$

さらなるイベント e'' が検出された時も同様にパラメータベクトルを更新する。これにより、イベントが検出されるたびにパラメータを逐次更新することが可能であり、それはすなわちユーザのロケーションが逐次更新されていくことを意味する。

5. 評価実験

提案手法の有効性を以下の三つの観点から検証する。

- (1) 検出されたイベントの妥当性
- (2) ユーザ位置推定の精度
- (3) ユーザ位置推定の効率性

ユーザ位置推定の効率性とは、ある一定の期間にどれだけの数のユーザのロケーションを推定できたかという観点である。

5.1 節では提案手法を実装したプロトタイプシステムについて説明し、5.2 節では本実験に用いた Twitter データセットについて説明する。5.3 節以下ではそれぞれの実験結果を示し、考察する。

5.1 プロトタイプシステム

提案手法を Twitter に適用し、イベント検出及びユーザ位置推定を行うプロトタイプシステムを実装した^(注4)。本システムの概要図を図 5 に示す。実線の枠で囲まれた部分が本システムの範囲である。本システムはツイートクローラによって継続的にツイートを収集する。実際には、検出したいイベントに関連すると思われる複数のキーワードを指定して、それらのうちのいずれかが含まれるツイートをリアルタイムに収集する^(注5)。収集したツイートを投稿したユーザのロケーション情報が必要となるため、ユーザロケーションクローラによって Twitter からユーザ情報を収集する。しかし、Twitter から収集されたユーザのロケーション情報はテキストであるため、Yahoo! Geocoder を用いて緯度経度情報に変換する。収集したロケーション情報をユーザ DB に格納する。ただし、ロケーション情報が未知のユーザについては NULL 値を格納する。この未知のロケーション情報を推定するのが本研究の一つの目的である。

収集したツイート及びユーザのロケーション情報を逐次イベント検出コンポーネントに入力する。ここで、収集したツイートはツイート DB に格納されるが、これはバックアップのためであり、本質的なものではない。イベント検出コンポーネントによって検出されたイベントは一旦イベント DB に格納され、ユーザ位置推定コンポーネントに入力される。そして、ロケーション分布、すなわちパラメータベクトル θ_u を推定し、ユーザ DB に格納されているパラメータベクトルを更新する。推定されたロケーション情報は以後のイベント検出に用いられる。

5.2 データセット

ツイートクローラ及びユーザロケーションクローラによって収集されたデータについて説明する。本実験は 2012/11/5 から 2012/11/19 までの二週間に五つのキーワード群について行

(注4)：システムのコードは次の URL から取得可能である。

<https://github.com/yamaguchiyuto/bagel>

(注5)：キーワードを指定せずにランダムサンプリングされたツイートを収集する API も存在するが、収集できるツイート数が少ないため今回は用いなかった。

(注3)：2 から 3 行目への変換は多項分布とディリクレ分布の共役性により導かれる。

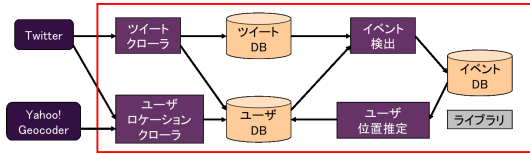


図 5 実験に用いたプロトタイプシステムの概要。Twitter から継続的にデータを収集するコンポーネントと、イベント検出及びユーザ位置推定を行うコンポーネントに分かれる。

表 1 データセット

	All	Earthquake	Weather
# of users	508,824	141,978	307,474
# of seed users	130,391 (26%)	36,613 (26%)	78,582 (26%)
# of tweets	1,018,164	317,982	519,803
keywords	-	地震	大雨, 洪水, 台風, 雷
	Tornado	Emergency	
# of users	15,868	81,592	
# of seed users	3412 (22%)	20,043 (25%)	
# of tweets	12,073	103,933	
keywords	竜巻	パトカー, 救急車, 消防車, サイレン	

われた。実験期間中に収集されたデータを表 1 に示す。各列がひとつのデータセット、すなわち一つのキーワード群について収集されたデータを示している。All は他の四つすべてのデータセットを統合して扱うデータセットである。ロケーション情報が既知である seed user はいずれのデータセットにおいても 25%前後であった。

5.3 イベント検出の妥当性

本実験では、提案手法によって検出されたイベントについて、以下の項目を検証した。

- 提案手法により検出されたイベントのうちどれだけのものが妥当であるか。
- 多くのユーザ位置情報が分かっているならば多くの妥当なイベントが検出できるか。

データセットは All 以外の四つを用い、パラメータはそれぞれ $WindowSize = 600s$, $MinPts = 15$, $Eps = 0.2$, $MaxDispersion = 200km$ とした。また、既知の位置情報の割合が提案手法によるイベント検出へ影響するかを検証するために、既知の位置情報のうちそれぞれ 100%, 50%, 25% を用いた場合について比較した。

本実験の詳細について示す。まず、提案手法によってイベント検出を行い、検出されたイベントに属す全ツイートを 5 名の被験者に提示する。被験者は提示されたイベントが以下の評価基準を満たすとき、そのイベントが妥当であるという判断を下す。

- イベントに属す全ツイートのうち半数以上が同一のイベントについて言及している。
- それらのツイートに言及されているイベントは地理的な局所性がある。

そして、妥当であるとされたイベントを正解とし、適合率を算出する。適合率を図 6、検出されたイベントの数を図 7 に示す。

図 6 によると、データセットによってその適合率が大きく異なっている。Earthquake, Weather, Tornado においては 0.7 から 0.8 程度の適合率を示しているが、Emergency においては 0.1 程度と、非常に低い適合率を示している。これは、実

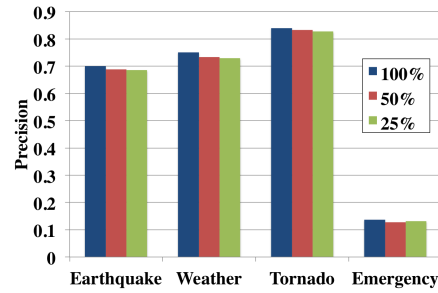


図 6 提案手法により検出されたイベントの適合率。実験期間中に多くのイベントが発生したデータセットにおける適合率が高くなっている。イベント検出に用いる位置情報の割合を大きくするとわずかに適合率が上昇している。

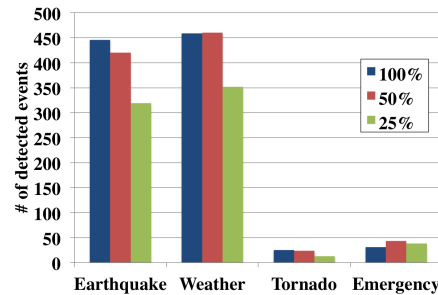


図 7 提案手法により検出されたイベントの数。実験期間中に多くのイベントが発生したデータセットにおいては多くのイベントが検出できている。イベント検出に用いる位置情報の割合を大きくすることで多くのイベントが検出されている。

験期間中に地震や雷、竜巻が発生しており、それに関する言及が Twitter において多く投稿されていたためであると考えられる。実際、それらのイベントが発生した際に付近にいたユーザが状況を発信しているツイートが多く見られた。一方で、Emergency データセットに属すツイートを確認したところ、イベントへの言及はほとんど見られなかった。このような場合、何も検出されないのが理想であるが、被験者によってイベントと見なされなかったものが検出されてしまったため、適合率が下がってしまったと考えられる。

イベント検出に用いる位置情報の割合を変化させたところ、割合を大きくするに連れて適合率はわずかに上昇し、検出されたイベント数はある程度増加した。本実験では、実際に位置推定を行い増加した位置情報を用いているわけではないが、位置情報の増加がイベント検出の妥当性向上に繋がることが期待される。位置情報を増やすことにより適合率が向上するのは、dispersion の算出がより正確になるためであると考えられる。ただし、Content Clustering の段階で内容が類似しているクラスタのみを検出しているため、たとえ dispersion が正確でなくてもある程度のイベントの妥当性は保証されている。これにより、位置情報の増加に伴う適合率の向上が数%にとどまったと考えられる。正確な dispersion を算出するためには、Content Clustering によって検出されたクラスタにおいて、高い割合のユーザの位置情報が既知であることが望まれる。クラスタ内に位置情報が既知であるユーザがふたり以上いない場合は dispersion を算出できないため、提案手法ではイベントではないとして破棄している。そのため、位置情報を増やすことによって検出されたイベント数が増加したと考えられる。

5.4 ユーザ位置推定の精度

提案手法によるユーザ位置推定の精度を、既存手法である Cheng らの手法 [4]^(注6)、Clodoveu らの手法 [6]、及びランダムにロケーションを割り当てる手法 (Random) と比較した。以下、実験の詳細について説明する。

提案手法は五つのデータセットのうち Weather データセットを用いて、ロケーション情報が既知である seed user のロケーションを一つ抜き交差検定 (Leave-One-Out Cross Validation) で推定し、精度を算出した。提案手法のパラメータはそれぞれ $WindowSize = 600s$, $MinPts = 15$, $Eps = 0.5$, $MaxDispersion = 150km$ である。これらは実験的に決定した。

Cheng ら、及び Clodoveu らの手法については、提案手法によって推定された seed user の中から 1,000 ユーザをランダムサンプリングし、それらのユーザに対して位置推定を行い、精度を算出した。Cheng らの手法は単語のロケーション分布を学習データ (ツイート) を用いて学習する必要があるため、All データセットのユーザから 10,000 ユーザをランダムサンプリングし^(注7)、それぞれのユーザの直近 100 ツイートを取得して学習データとした。提案手法に用いた全ツイート数が約 1,000,000 であるため、Cheng に用いる学習データも 1,000,000 のツイートとした。推定の対象とする 1,000 ユーザそれぞれの直近 3,200 ツイートを収集し、それらに含まれる単語と事前に学習した単語のロケーション分布を用いてユーザの位置推定を行った。また、Clodoveu らの手法は推定の対象とするユーザと相互にフォローしているユーザのロケーション情報を用いるため、それらのデータも収集した。

図 8、表 2 に結果を示す。図 8 の横軸は許容する推定誤差の値を示し、縦軸はそれに対応する精度を示している。表 2 において、*Mean E.D.* は推定誤差の平均値を表し、*Median E.D.* は推定誤差の中央値を表す。結果から、提案手法が最も高い精度を示していることが分かる。推定誤差 160km での提案手法の精度を Cheng の精度と比較すると約 122%、Clodoveu らの手法と比較すると約 33% の向上となった。また、推定誤差の平均値及び中央値に関しても提案手法が最も良い結果を示した。

提案手法では Weather データセットを用いたが、データの収集期間に日本各地で大雨や雷が多く発生していたため、強い局所性を持つイベントが多く検出され、精度が高くなったと考えられる。また、Cheng の精度は元論文中 [4] における精度よりも大きく低下している。元論文で報告されていた精度は 0.510 であった。これは、元論文ではアメリカ全土における位置推定を行なっているのに対し、本実験では日本を対象として位置推定を行なっているためであると考えられる。元論文での実験と本稿での実験は扱っている言語が異なるため、ストップワードなどのチューニングを行うことで Cheng らの手法は精度が向上すると考えられる。

提案手法は精度では比較手法を上回ったが、任意のユーザのロケーションを推定できないという点で比較手法と異なっている。Cheng 及び Clodoveu は任意のユーザのツイート集合もしくは相互にフォローしているユーザ集合が与えられれば位置推定が可能であるが、提案手法は検出されたイベントに関するポストを投稿していないユーザの位置推定は出来ない。しかし、この違いは、未知のユーザのロケーションを推定し、既知の

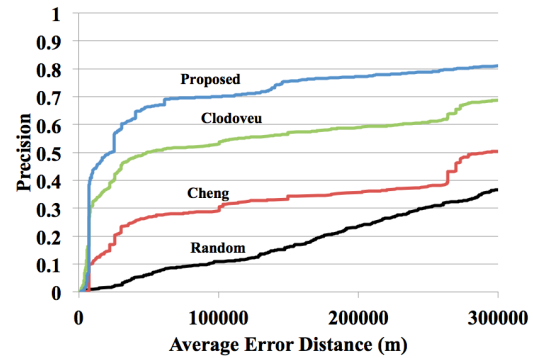


図 8 ユーザ位置推定の精度の比較結果. 推定誤差 160km での提案手法の精度を Cheng と比較すると約 122%、Clodoveu と比較すると約 33% の精度向上を示している。

表 2 精度及び平均推定誤差の具体的な値

	Proposed	Cheng	Clodoveu	Random
<i>Pre@160km</i>	0.761	0.344	0.573	0.170
<i>Pre@80km</i>	0.696	0.284	0.520	0.092
<i>Mean E.D. (m)</i>	134,114	336,489	227,100	457,666
<i>Median E.D. (m)</i>	22,862	290,465	49,118	416,106

ユーザのロケーション情報を増やし、イベント検出に役立てるという本研究の目的を達成する上では問題にならない。また、推定に用いている情報がそれぞれ異なるため、提案手法と比較手法とを組み合わせることが可能であると考えられる。

5.5 ユーザ位置推定の効率性

ある一定の期間にどれだけの数のユーザのロケーションを推定できたかという効率性について検証する。本節では、五つのデータセットによるそれぞれの結果の比較、パラメータ $MaxDispersion$ の変化による結果の比較を行う。パラメータはそれぞれ $WindowSize = 600s$, $MinPts = 15$, $Eps = 0.5$, $MaxDispersion = 200km$ である。

5.5.1 位置推定されたユーザ数の時間推移

五つのデータセットそれぞれに対して提案手法を適用し、6 時間ごとにそれまでにロケーションを推定できたユーザの数を記録した。結果を図 9 に示す。横軸は経過時間を表し、縦軸はそれまでにロケーションが推定されたユーザ数、すなわち効率性を表す。

All, Earthquake, Weather データセットでは効率性が高くなっているが、Tornado, Emergency データセットでは効率性が低いことが分かる。これは、効率性の高い三つのデータセットでは多くのイベントが検出されたためである。また、図 9 において、推定されたユーザ数が階段上に増加している部分があるが、これは比較的大きなイベントが検出されたためである。従って、イベントが多く発生するようなソーシャルストリームを用いれば効率良くユーザ位置推定が行えると考えられる。

5.5.2 $MaxDispersion$ の変化による影響

パラメータ $MaxDispersion$ を変化させることによって、それぞれのデータセットにおける精度と効率性の関係を検証した。結果を図 10 に示す。横軸は効率性を表し、縦軸は精度を表す。 $MaxDispersion$ は 50km から 50km 刻みで 300km までと、Spatial Filtering を行わない、すなわち $MaxDispersion$ が無限である NSF について比較した。

図 10 によると、 $MaxDispersion$ を大きくするにしたがって、ロケーションが推定されたユーザ数は単調増加していることが分かる。これは、 $MaxDispersion$ が大きいと地理的な局所性を

(注6)：ただし、論文中 [4] であまり効果がないとされていたスムージングは行っていない。

(注7)：推定の対象とする 1,000 ユーザとの重複は許さない。

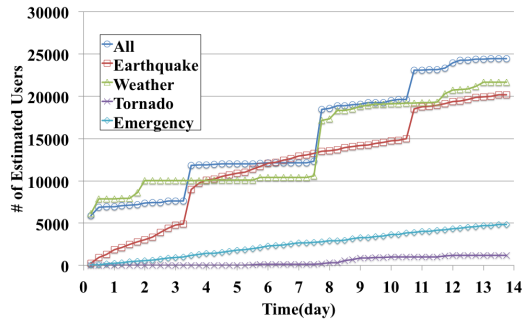


図9 提案手法によってロケーションを推定されたユーザ数の時間推移. 多くのイベントが検出されたデータセットにおいては推定されたユーザ数が大きいのが分かる. また, 推定されたユーザ数が階段上に増加しているのは比較的大きなイベントが検出されたからであると考えられる.

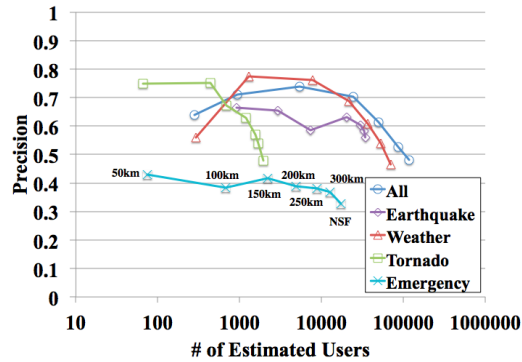


図10 パラメータ $MaxDispersion$ を変化させた時の精度及び効率性への影響. 精度と効率性はトレードオフの関係にあることが分かる. また, $MaxDispersion$ がデータセットに対して小さすぎる場合は精度, 効率性共に小さくなっていることが分かる.

あまり持たないクラスタであってもイベントとして検出され, それに伴い多くのユーザがロケーションを推定されるためである. 一方で, 精度は $MaxDispersion$ を大きくするに従って下がる傾向にあることが分かる. これは, 局所性をあまり持たないイベントではうまくユーザのロケーションを推定できないためであると考えられる. また, All, Weather において見られるが, $MaxDispersion$ が小さすぎると精度も下がる傾向にある. これは, データセットに対して小さすぎる $MaxDispersion$ を与えると, 妥当なイベントを検出できなくなるためであると考えられる. 実際に, これらのデータセットにおいては $Dispersion$ の値が $50km$ を下回るような妥当なイベントはあまり見られなかった.

6. 結 論

本研究では, ソーシャルストリームからのイベント検出とユーザ位置推定を統合的かつ継続的に行う手法を提案した. 提案手法は, 地理的な局所性を持つローカルイベントに関するポストを投稿したユーザはそのローカルイベントが発生した場所にいる可能性が高いというアイデアに基づいている. また, Twitter におけるソーシャルストリームを用いて評価実験を行い, 検出されたイベントの妥当性, ユーザ位置推定の精度及び効率性という観点から提案手法の有効性を示した.

今後の課題として, 提案手法を長期的に稼働させる実験をする必要がある. 長期的なスパンで, イベント検出とユーザ位置

推定の結果の相互作用を調査することで, より適切なアルゴリズム及びパラメータが検討できると考えられる. また, イベントの情報のみではなくソーシャルグラフなどを考慮することにより, より広い範囲のユーザの位置推定が可能になると考えられる. さらに, 本稿ではユーザの位置情報を離散確率分布によってモデル化したが, 正規分布などの連続確率分布によるモデル化による精度の向上も考えられる.

謝辞 本研究に関して議論していただいた日本アイ・ビー・エム株式会社東京基礎研究所の皆様へ深く感謝する.

文 献

- [1] Backstrom, L., Sun, E. and Marlow, C.: Find me if you can: improving geographical prediction with social and spatial proximity, *WWW*, pp. 61–70 (2010).
- [2] Becker, H., Naaman, M. and Gravano, L.: Beyond Trending Topics: Real-World Event Identification on Twitter, *ICWSM* (2011).
- [3] Chang, H.-W., Lee, D., Eltaher, M. and Lee, J.: @Phillies Tweeting from Philly? Predicting Twitter User Locations with Spatial Word Usage, *ASONAM*, pp. 111–118 (2012).
- [4] Cheng, Z., Caverlee, J. and Lee, K.: You are where you tweet: a content-based approach to geo-locating twitter users, *CIKM*, pp. 759–768 (2010).
- [5] Cui, A., Zhang, M., Liu, Y., Ma, S. and Zhang, K.: Discover breaking events with popular hashtags in twitter, *CIKM*, pp. 1794–1798 (2012).
- [6] Jr., C. A. D., Pappa, G. L., de Oliveira, D. R. R. and de Lima Arcanjo, F.: Inferring the Location of Twitter Messages Based on User Relationships, *T. GIS*, Vol. 15, No. 6, pp. 735–751 (2011).
- [7] Lappas, T., Vieira, M. R., Gunopulos, D. and Tsotras, V. J.: On The Spatiotemporal Burstiness of Terms, *PVLDB*, Vol. 5, No. 9, pp. 836–847 (2012).
- [8] Lee, C.-H.: Mining spatio-temporal information on microblogging streams using a density-based online clustering method, *Expert Syst. Appl.*, Vol. 39, No. 10, pp. 9623–9641 (2012).
- [9] Li, C., Sun, A. and Datta, A.: Twevent: segment-based event detection from tweets, *CIKM*, pp. 155–164 (2012).
- [10] Li, R., Wang, S. and Chang, K. C.-C.: Multiple Location Profiling for Users and Relationships from Social Network and Content, *PVLDB*, Vol. 5, No. 11, pp. 1603–1614 (2012).
- [11] Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S. and Miller, R. C.: Twitinfo: aggregating and visualizing microblogs for event exploration, *CHI*, pp. 227–236 (2011).
- [12] Mathioudakis, M., Bansal, N. and Koudas, N.: Identifying, Attributing and Describing Spatial Bursts, *PVLDB*, Vol. 3, No. 1, pp. 1091–1102 (2010).
- [13] Ritter, A., Mausam, Etzioni, O. and Clark, S.: Open domain event extraction from twitter, *KDD*, pp. 1104–1112 (2012).
- [14] Sadilek, A., Kautz, H. A. and Bigham, J. P.: Finding your friends and following them to where you are, *WSDM*, pp. 723–732 (2012).
- [15] Sakaki, T., Okazaki, M. and Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors, *WWW*, pp. 851–860 (2010).
- [16] Watanabe, K., Ochi, M., Okabe, M. and Onai, R.: Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs, *CIKM*, pp. 2541–2544 (2011).