情報処理学会論文誌　データベース　Vol.6　No.5　23–37 (Dec. 2013)

ソーシャルメディアにおけるローカルイベントを用いたユーザ位置推定手法

山口 祐人[1,2,a)]　伊川 洋平[3,b)]　天笠 俊之[4,c)]　北川 博之[4,d)]

受付日 2013年6月21日, 採録日 2013年10月4日

概要：ソーシャルメディアの普及にともない，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．また，ソーシャルメディアのユーザは個々人が居住地などの特定の位置情報と結びついているという特徴がある．これらの情報を用いると，たとえば特定の地域のユーザへ新しいレストランを推薦したり，災害情報を提供したりと，様々なサービスの提供が可能であると考えられる．しかし，多くのユーザは自らの居住地情報などの位置情報を公開していないという現状がある．そのため，ユーザ位置情報の推定は，ソーシャルメディア分析における重要なタスクとなっている．一方，ソーシャルメディアへのユーザの投稿を用いると，地震や火事などの地理的な局所性を持つローカルイベントを検出することが可能である．そこで本研究では，あるローカルイベントに関する投稿をしたユーザはそのイベントが発生した地域にいる可能性が高いという考え方に基づき，ユーザ位置を推定する手法を提案する．提案手法では，まず位置情報が既知であるユーザの投稿を用いてローカルイベントの検出を行い，検出されたローカルイベントに関する投稿をした，位置情報が未知であるユーザの位置情報を推定する．評価実験により，提案手法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：ユーザ位置推定，ソーシャルストリーム，ローカルイベント検出，ソーシャルメディア，Twitter

User Location Inference Using Local Events in Social Media

Yuto Yamaguchi[1,2,a)]　Yohei Ikawa[3,b)]　Toshiyuki Amagasa[4,c)]　Hiroyuki Kitagawa[4,d)]

Received: June 21, 2013, Accepted: October 4, 2013

Abstract: People using social media transmit vast quantities of information in real time, which forms real-time information sources called social streams. An important characteristics of such media is that one can disclose their home location information to other users. By using such information, we can provide several services such as recommending restaurants or providing disaster-related information to users who live in a certain area. However, due to the fact that not many users publicize their home locations, there is a lack of information to provide such services. For this reason, there is a strong demand for inferring users' home locations. Meanwhile, by monitoring social streams, we can detect local events (e.g., earthquakes, fires, etc.) because people all over the world may post messages about those local events instantly. In this paper, we propose a method for user location inference using local events detected from social streams. Our method is based on the assumption that users who post about a local event likely to live near the event. Specifically, the method first detects local events using messages posted by location-known users, and then infer home locations of location-unknown users who post about the detected event. Experimental results show that our method can properly detect local events and infer user locations more precisely than other existing location inference methods.

Keywords: user location inference, social streams, local event detection, social media, Twitter

[1] 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
[2] 日本学術振興会特別研究員 DC1
Research Fellow of the Japan Society for the Promotion of Science, Chiyoda, Tokyo 102–0083, Japan
[3] 日本アイ・ビー・エム株式会社東京基礎研究所
IBM Research-Tokyo, IBM Japan Ltd., Koto, Tokyo 135–8511, Japan
[4] 筑波大学システム情報系
Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
a) yuto_ymgc@kde.cs.tsukuba.ac.jp
b) yikawa@jp.ibm.com
c) amagasa@cs.tsukuba.ac.jp
d) 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．たとえば，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．

ソーシャルメディアには，ユーザそれぞれに位置情報が結びついているという特徴がある．ソーシャルメディアにおけるユーザの位置情報としては**居住地**と**現在地**の 2 つが考えられる．居住地はユーザが定常的にいる場所であり，ユーザのプロファイルなどから得られる．一方，現在地はある時点でユーザがいる場所であり，それぞれの投稿に付与された GPS タグなどから得られる．これらの位置情報を用いると，たとえば特定の地域のユーザに新しいレストランなどを推薦する [13]，または災害情報を提供するというようなサービスの実現が可能であると考えられる．また，特定の地域のユーザからの投稿を用いて災害について分析する研究も行われている [21], [24]．さらに，企業活動の観点からも，特定の地域のユーザに自社製品の広告を提供するなど，有益なアプリケーションが考えられる．このように，ソーシャルメディアにおけるユーザ位置情報は重要であるといえる．本研究では，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き，ユーザの位置情報として**居住地**に焦点を当てる．

しかし，それぞれのメディアにおいてユーザの居住地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 [5] 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，Backstrom 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，上記のアプリケーションなどを実現するには，ユーザの居住地を高い精度で推定することが重要である．ユーザの居住地推定を行う手法は，投稿されたテキストの内容を用いるもの [4], [5], [15] や，ソーシャルグラフを用いるもの [1], [8], [19] など，様々なものが提案されている．

一方，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ことを目的とした研究が多く行われている [2], [6], [11], [14], [18], [23]．全世界のユーザがそれぞれの状況や意見を自由に投稿しているため，ソーシャルストリームをモニタリングすることで，実世界の様子をとらえることができる．たとえば，東京で地震が発生したときには，東京にいるユーザがいっせいに「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 [20]．

本研究では，上記のイベント検出のアイデアとは逆の考え方により，ユーザの位置推定を行う．たとえば，東京で地震が発生したときに「地震だ」のような投稿をしたユーザは東京にいる可能性が高いと考えることができる．**図 1** はある期間に Twitter に投稿されたツイートの地理的な分布[*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の数を表す．これらの図によると，あるローカルイベントが発生したとき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示されている．なお，本研究ではイベントをユーザ位置推定に応用することを想定しているため，株価の急激な変動やクリスマスなどの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．

本研究の貢献は以下に示す 3 点である．

(1) 検出されたローカルイベントを用いてユーザの居住地

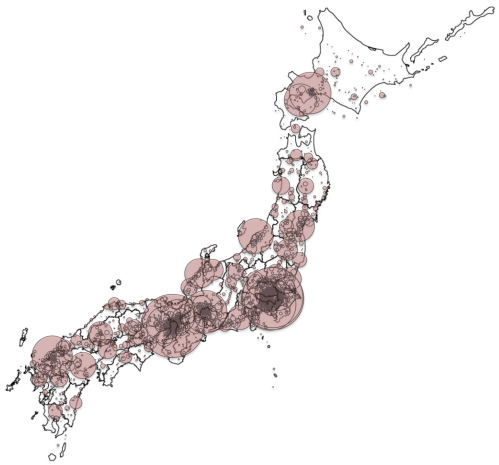

図 1　ある期間に投稿されたツイートの地理的な分布．円の大きさは同じ位置から投稿されたツイートの数を示す．東京や大阪などの大都市からの投稿が多いことが分かる

Fig. 1　A geographical distribution of tweets posted in a certain time period. The size of circles illustrates the number of tweets posted from the corresponding location. We can see that there are a lot of tweets from metropolises such as Tokyo and Osaka.

[*1] ツイートを投稿したユーザの居住地の地理的な分布．

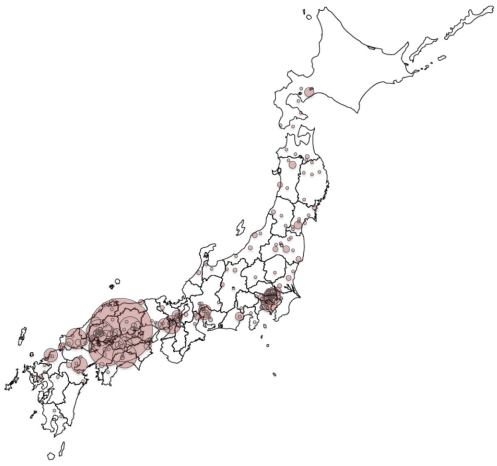

図 2　広島で地震が発生したときに投稿された「地震」という単語を含むツイートの地理的な分布．円の大きさは同じ位置から投稿されたツイートの数を示す．普段の地理的な分布（図 1）とは異なり，広島付近からの投稿が多くなっていることが分かる．一方で，東京や大阪などからの投稿もある程度の数を保っていることが分かる

Fig. 2　A geographical distribution of tweets that contain the word “earthquake” when an earthquake happened at Hiroshima prefecture. The size of circles illustrates the number of tweets posted from the corresponding location. Compared to the usual distribution (Fig. 1), we can see that the number of tweets from Hiroshima increases. Meanwhile, tweets from metropolises retain a certain number.

推定を行う手法を提案する．

(2) イベントが検出されるたびにユーザの居住地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，提案手法によるユーザ居住地推定の有効性を示す．

提案手法はまず居住地情報が既知であるユーザの投稿を用いてローカルイベントの検出を行う．そして，検出されたイベントを用いて，それに関する投稿をした居住地情報が未知であるユーザの居住地推定を行う．なお，本稿の以降では単にユーザの位置といえばユーザの居住地を示すものとする．

5 章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 章の評価実験によると，提案手法を 2 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．

本稿の以降の構成は以下のとおりである．まず，2 章でイベントを検出する関連研究およびユーザ位置推定を行う関連研究を概観し，3 章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 章で本研究の提案手法について述べ，5 章で提案手法の有効性を，検出されたイベントの妥当性，ユーザ位置推定の精度およ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 章で本稿をまとめる．

2. 関連研究

本章では，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，およびソーシャルストリームからのローカルイベント検出に関する研究について概観する．

2.1 ユーザ位置推定

ソーシャルメディアにおいてユーザの位置推定を行う手法は，1) ユーザが投稿したコンテンツを用いる手法，2) ソーシャルグラフ上でのユーザ間の関係を用いる手法，3) コンテンツとソーシャルグラフの両方を用いる手法の 3 つのカテゴリに分けることができる．

2.1.1 コンテンツを用いた位置推定手法

Cheng ら [5] は，**ローカルワード**を用いて Twitter ユーザの居住地を推定する手法を提案した．ローカルワードとは，その単語を投稿したユーザの居住地が比較的狭い範囲に偏っているような単語のことをいう．たとえば，*rockets* という単語は，アメリカのヒューストンに住むユーザから頻繁に投稿されるため，ローカルワードであると報告されている．Cheng らはこのようなローカルワードの地理的な分布を用いて，それらを含むツイートを投稿したユーザの居住地を推定した．

Chang ら [4] は，単語の地理的な分布を混合正規分布を用いてモデル化し，それを用いてユーザの位置を推定する手法を提案した．Chang らの手法は Cheng らの手法とは異なり，ローカルワードの抽出に人手で作成した教師データを必要としない．実験結果によると，Chang らの手法は比較的少ないローカルワードを用いるだけで Cheng らの手法と同等の精度を達成した．

Kinsella ら [9] は，GPS によるジオタグが付与されたツイートを用いて各都市の言語モデルを構築し，このモデルを用いてユーザの位置推定を行った．Kinsella らは，ジオタグを用いた手法は，Cheng らや Chang らのようにユーザのプロファイルのみを用いる手法よりもユーザの移動に頑強な言語モデルを構築できると報告している．

Chandra ら [3] は，ユーザ同士の会話に注目した．同じ会話に属するツイートは同じトピックについての内容であるとし，この考え方に基づいて言語モデルを構築した．実験結果によると会話の構造を用いた言語モデルによるユーザ位置推定手法のほうが，会話を用いないものよりも精度が高いことが報告されている．

2.1.2 ソーシャルグラフを用いた位置推定手法

Clodoveu ら [8] 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程度の場合に良い結果が得られると主張している．

Backstrom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．Backstromらの手法は，あるユーザの居住地は，その友人とのエッジが張られる尤度を最大にするような場所であるとした．実験結果によると，IPアドレスを用いたアプローチよりも高い精度が示されたと報告されている．

Sadilekら[19]は，ユーザの居住地ではなく，GPSによるTwitterユーザの移動軌跡を推定する手法を提案した．提案手法では，移動軌跡の推定とソーシャルグラフにおけるリンク推定は相互補完的な関係にあるとし，2つのタスクを同時に行った．また，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，一緒に行動している友人が自らの位置情報を公開するだけで，それが推定されてしまうと主張している．

2.1.3　コンテンツとソーシャルグラフの両方を用いた手法

Liら[16]は，unified discriminative influence model（UDI）というモデルを提案し，ユーザの居住地推定を行った．UDIはユーザが投稿したツイートとソーシャルグラフ上でのユーザ間の関係を，ユーザと地名をノードとする異種グラフ（heterogeneous graph）としてモデル化する．また，それぞれのノードは異なる影響力を持つとした．強い影響力を持つユーザ（たとえば，レディー・ガガなど）は世界中のユーザからフォローされるため，そのようなユーザをフォローしていても位置推定の手がかりにはならない．Liらの手法はこの考え方に基づき，異種グラフが得られる尤度を最も大きくするようなロケーションをそれぞれのユーザの居住地であるとして推定した．Liら[15]はまた，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複数のロケーションを推定する手法も提案した．

以上のユーザ位置推定に関する研究は，手がかりとしている情報が異なるという点で本研究とは異なる．コンテンツを用いた位置推定手法は本研究と類似しているが，単語の定常的な地理的分布のみを考慮し，イベントという時間的な側面を考慮していないため本研究とは異なる．

2.2　ローカルイベント検出

ソーシャルメディアに投稿された情報を用いて，実世界のイベントの検出を行う研究が多く行われている．Rattenburyら[17]は，Flickrに投稿された位置情報付きの写真を用いて，時間的，もしくは地理的なバーストを発見する手法を提案した．Rattenburyらの手法では，写真に付けられているタグ（New York，World Cup，dogなど）の時間的，地理的な分布を分析し，有意な偏りを持つタグはイベントを表すとした．

Lappasら[10]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．

近年では，Twitterのリアルタイムな性質を用いたローカルイベント検出の研究がさかんである．Sakakiら[20]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．

Waltherら[22]は，地理空間上におけるイベントを検出するシステムを構築した．また，Waltherらはどのような特徴量を用いれば精度良くイベント検出を行えるかを議論し，あるロケーションから投稿するユーザの数とその投稿のトピックを分析すれば良い結果が得られると報告している．

Leeら[12]も同様にTwitterを用いて地理空間上におけるイベントを検出する手法を提案した．Leeらの手法は地理空間を四分木を用いて領域分割し，それぞれの領域においてツイートの投稿数が通常の状態から予測される数よりも多いときにその領域内でイベントが発生しているとした．

Watanabeら[23]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．

Ritterら[18]は，Twitterのデータを用いて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．

Leeら[11]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．

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．

3.　問題定義

本章では，本研究で用いる用語の定義およびローカル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**$SS=(p_1,p_2,\ldots)$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．ポストpを投稿したユーザをu_pと表す．また，イ

ベント e とは，次の条件をすべて満たすポストの集合であると定義する．

- すべての $p_i \in e$ のタイムスタンプ 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，
- 互いのロケーション l_i 間の距離が互いに十分に小さく，
- イベント e に属するポストの数が定められた値より大きい．

たとえば，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見なすことができると考えられる．このとき，イベント検出を以下のように定義する．

問題 1（ローカルイベント検出）　与えられたタイムウィンドウ W_j に属するポストの集合からイベントの集合 $E_j = \{e_1, e_2, \ldots, e_n\}$ を検出する．ここで，それぞれのイベントはイベントである条件を満たすポストの**極大集合**であるとする．ただし，1 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．

ソーシャルストリームに投稿をしうるユーザ $u \in U$ 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 $\hat{l_u} = \mathrm{argmax}_l P_u(l)$ 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．

問題 2（ユーザ位置推定）　検出されたイベント e が与えられたとき，e に属するポスト $p \in e$ を投稿したそれぞれのユーザ u_p に対して，$\hat{l_{u_p}} = \mathrm{argmax}_l P_{u_p}(l)$ がユーザ u_p の実際のロケーション l_{u_p} となるように，e を用いてユーザのロケーション分布 $P_{u_p}(l)$ 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，1 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

本章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．

4.1 ローカルイベント検出

提案手法はまずこれまでに既知となっているユーザのロケーションとソーシャルストリームを用いてローカル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，たとえば 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，単語ベクトル $\boldsymbol{v(t)}$ 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．

本手法では，クラスタリング手法として，ノイズに対して頑強な手法を用いる．ノイズに対して頑強な手法とは，必ずしもすべてのデータ点がクラスタに属するのではなく，自らと類似するデータ点の少ないデータ点はノイズと見なされる手法である．これにより，短期間に類似したポストが多く投稿されたときにのみイベントが検出されることが期待される．

本稿では，クラスタリング手法として密度ベース手法である DBSCAN [7] を用いる．距離関数として単語ベクトル間のユークリッド距離 $dist_t(\boldsymbol{v(t_i)}, \boldsymbol{v(t_j)})$ を用いる．また，DBSCAN はパラメータとして $MinPts$ と Eps をとる．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．

Content Clustering の出力は採用したクラスタリング手法によって出力されたクラスタの集合 $C_j = \{c_1, c_2, \ldots, c_n\}$ である．入力として与えられたタイムウィンドウ W_j に対してクラスタ集合 C_j が出力される．Content Clustering の処理の流れを**図 3** に示す．

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

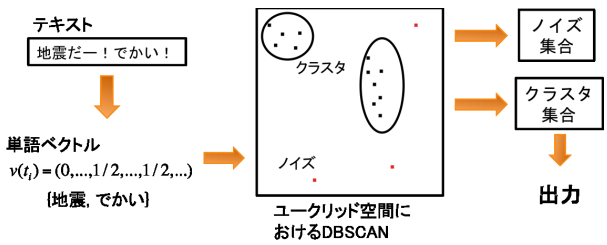

図 3　DBSCAN によるポストのクラスタリング．DBSCAN によりノイズと見なされたポストは破棄し，クラスタの集合のみを出力する

Fig. 3　The procedure of clustering of posts by DBSCAN. Posts regarded as noises by DBSCAN are disposed. The output is a set of clusters.

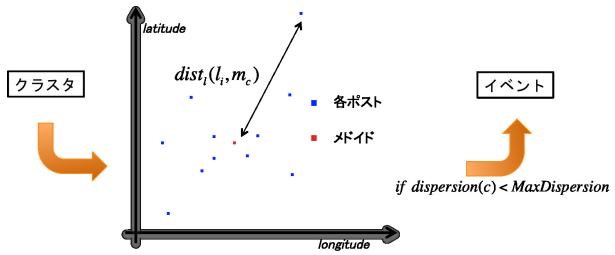

図 4　Spatial Filtering による地理的な局所性の判別．算出された $dispersion(c)$ がパラメータ $MaxDispersion$ を超えなければ，クラスタ c をイベントとして検出する

Fig. 4　The procedure of filtering clustering with respect to the geographical locality by Spatial Filtering. Cluster c is regarded as an event if the value of $dispersion(c)$ is lower than the parameter $MaxDispersion$.

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[*2]．ただし，本提案手法では，ポスト p のロケーション l はポストのジオタグではなく，p を投稿したユーザの居住地を用いていることに注意されたい．

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．

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

ただし，$dist_l(\cdot, \cdot)$ は 2 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大きくなる．

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 $E_j = \{e_1, e_2, \ldots, e_m\}$ を出力する（図 4）．ここで，パラメータ $MaxDispersion$ は位置推定のためのイベントの最大粒度を表す．

4.1.3　提案手法のイベント検出における妥当性

本研究で提案するイベント検出手法は Content Clustering と Spatial Filtering の 2 ステップによってローカルイベントを検出する．これにより，内容が類似しており，かつ地理的に局所性のあるポストの集合がイベントとして抽出される．また，Content Clustering はノイズに頑強なクラスタリング手法を用いることを要請しているため，一種のバースト検出として働いていると考えられる．これは，内容が類似しているポストが複数存在していても，その数が $MinPts$ より小さければノイズと見なされるためである．さらに，Content Clustering では “あけましておめでとう” などの，地理的な局所性は持たないが，内容が類似しておりかつ同時刻に大量に投稿されるポストがクラスタとして検出されると考えられるが，これは Spatial Filtering によって地理的な局所性は持たないとしてフィルタリングされる．

[*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．

Algorithm 1 ローカルイベント検出アルゴリズム

Input: set of posts P_j belongs to time window W_j
Output: set of events E_j
  $E_j \leftarrow \emptyset$
  $C_j \leftarrow$ clustering_method(P_j)　// noises are discarded.
  for all cluster c in C_j **do**
    $m_c \leftarrow$ calculate_medoid(c)
    $d_c \leftarrow \frac{1}{|c|} \sum_{p_i \in c} dist_l(l_i, m_c)$
    if $d_c < MaxDispersion$ **then**
      $E_j \leftarrow E_j \cup c$
    end if
  end for
  return E_j

4.1.4　ローカルイベント検出アルゴリズム

イベント検出のアルゴリズムをアルゴリズム 1 に示す．本アルゴリズムは，あるタイムウィンドウ W_j に属するポストの集合を入力として受け取り，検出したイベントの集合を出力する．本アルゴリズムの計算量はクラスタリング部分の計算量による．

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 に属するポスト p を投稿したユーザ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

とりうるロケーションの集合 L について，ユーザ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 $\boldsymbol{\theta_u} = (\theta_{u,1}, \ldots, \theta_{u,|L|})$ で表す．

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

提案手法では，それぞれのユーザ 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 $\boldsymbol{\theta_u}$ を推定する．ユーザのロケーション $\hat{l_u}$ は，推定されたパラメータベクトル $\boldsymbol{\theta_u}$ を用いて以下のように

与えられる．

$$\hat{l_u} = \arg\max_{l \in L} \theta_{u,l} \tag{3}$$

パラメータベクトル $\boldsymbol{\theta_u}$ は MAP 推定，

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

によって得られる．ここで，E_u はユーザ u が言及したイベントの集合であり，以下で与えられる．

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

ただし，$mention(u, e)$ は u が e に属するポスト p を投稿していれば真となる．また，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．

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

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 $P(E_u|\boldsymbol{\theta})$ は次を満たす．

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

ここで，$P(e|\boldsymbol{\theta})$ はパラメータ $\boldsymbol{\theta}$ 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．

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

以下では，イベント e の尤度関数 $P(e|\boldsymbol{\theta})$ およびパラメータの事前分布 $P(\boldsymbol{\theta})$ を求める．

イベント e に対して，e に属するポストのロケーションの集合を考え，bag-of-words モデルと同様の考え方により，ロケーションベクトル $\boldsymbol{v(e)} = (n_{e,1}, n_{e,2}, \ldots, n_{e,|L|})$ 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 $\boldsymbol{v(e)}$ はパラメータ $\boldsymbol{\theta}$ 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．

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

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 $\boldsymbol{\theta}$ の事前分布としてディリクレ分布を採用する．

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

ただし，$\Gamma(\cdot)$ はガンマ関数，$\boldsymbol{\alpha}$ 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およ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[*3]．

$$\begin{aligned}
\boldsymbol{\theta}_u &= \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{\theta}) \prod_{e \in E_u} P(e|\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} Dir(\boldsymbol{\theta}; \boldsymbol{\alpha}) \prod_{e \in E_u} Multi(\boldsymbol{v(e)}; \boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} Dir\left(\boldsymbol{\theta}; \boldsymbol{\alpha} + \sum_{e \in E_u} \boldsymbol{v(e)}\right)
\end{aligned} \tag{11}$$

これを解くと

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

となり，パラメータベクトル $\boldsymbol{\theta_u}$ 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' が検出されたとき，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 $P(\boldsymbol{\theta}|E_u)$ を事前分布 $P(\boldsymbol{\theta})$ とし，以下のように再度 MAP 推定を行う．

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

これを解くと，$\theta_{u,k}$ は次のように更新される．

$$\theta_{u,k} = \frac{\sum_{e \in E_u} n_{e,k} + n_{e',k} + \alpha_k - 1}{\sum_{e \in E_u} N_e + N_{e'} + A + |L|}. \tag{14}$$

さらなるイベント e'' が検出されたとき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．

4.2.4　逐次更新の妥当性と利点

パラメータベクトルの逐次更新はオンライン学習法であるが，その解は現在までのすべてのイベントの集合を用いてバッチ学習をしたときの解と一致することを示す．タイムウィンドウ W_0 から W_t までに抽出され，かつユーザ u が言及したすべてのイベントの集合を $E_u = \{e_0, e_1, \ldots, e_m\}$ とする．このとき，バッチ学習によってパラメータベクトルを求めると，式 (12) と一致する．ただし，α_k は事前分布として採用したディリクレ分布のパラメータである．

一方，これをオンライン学習法を用いて解いた場合について考察する．イベントの集合 E_u を，各イベントの得られた時刻順に並べてイベントの列 $E'_u = (e_0, e_1, \ldots, e_m)$ とする．まず，はじめに抽出されたイベント e_0 のみを用いてパラメータベクトルを求める．事前分布をパラメータ α_k のディリクレ分布すると，

[*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．

$$\begin{aligned}\boldsymbol{\theta}_u &= \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|e_0)\\ &= \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{\theta})P(e_0|\boldsymbol{\theta})\\ &= \arg\max_{\boldsymbol{\theta}} Dir(\boldsymbol{\theta};\boldsymbol{\alpha})Multi(\boldsymbol{v}(\boldsymbol{e_0});\boldsymbol{\theta})\\ &= \arg\max_{\boldsymbol{\theta}} Dir(\boldsymbol{\theta};\boldsymbol{\alpha}+\boldsymbol{v}(\boldsymbol{e_0}))\end{aligned} \tag{15}$$

となり，これを解くと

$$\theta_{u,k} = \frac{n_{e_0,k}+\alpha_k-1}{N_{e_0}+A+|L|} \tag{16}$$

となる．さらに，事後分布 $P(\boldsymbol{\theta}|e_0)$ を事前分布と見なし，次のイベント e_1 を用いてパラメータベクトルを更新する．すると，同様に解

$$\theta_{u,k} = \frac{n_{e_0,k}+n_{e_1,k}+\alpha_k-1}{N_{e_0}+N_{e_1}+A+|L|} \tag{17}$$

が得られる．これを繰り返すと最終的に

$$\theta_{u,k} = \frac{\sum_{i=0}^{m} n_{e_i,k}+\alpha_k-1}{\sum_{i=0}^{m} N_{e_i}+A+|L|} \tag{18}$$

という解が得られる．これはバッチ学習法による解と一致している．

逐次更新にはバッチ学習と比較すると次の利点がある．

- 新しく到着したイベントのみを用いてパラメータベクトルを更新することができる．
- これまでのすべてのイベントを保存しておく必要がない．

後者は新しくイベントが到着した際にバッチ学習によってパラメータベクトルを更新するためには，これまでに検出されているすべてのイベントを用いる必要があるためである．一方，逐次更新ではその必要はない．これら 2 つはイベントが大量に終わりなく検出され続けるソーシャルストリームなどの状況下では大きな利点になると考える．

4.2.5　ユーザ位置推定アルゴリズム

ユーザ位置推定アルゴリズムをアルゴリズム 2 に示す．本アルゴリズムは検出されたイベント e を入力として受け取り，それに基づいて対応するユーザのロケーション分布を更新する．ユーザの位置を推定するには，$n_{e,k}$ のみを保存しておけばよい．これは，式 (19) に示すように，他の項は確率値最大のロケーションを選ぶ際には必要とならないためである．

Algorithm 2 ユーザ位置推定アルゴリズム

```
Input: detected event e, parameter vectors θ[u][l]
Output: updated parameter vectors θ[u][l]
  for all post p in e do
    u ← user(p)   // returns the user who posts p
    for all post q in e do
      v ← user(q)   // returns the user who posts q
      l ← location(v)   // returns the location of v
      if u ≠ v and l ≠ NULL then
        θ[u][l] ← θ[u][l] + 1
      end if
    end for
  end for
  return θ[u][l]
```

$$\begin{aligned}\arg\max_k \theta_{u,k} &= \arg\max_k \frac{\sum_{e\in E_u} n_{e,k}+\alpha_k-1}{\sum_{e\in E_u} N_e+A+|L|}\\ &= \arg\max_k \left(\sum_{e\in E_u} n_{e,k}+\alpha_k\right)\end{aligned} \tag{19}$$

本アルゴリズムの計算量は $O(|e|^2\cdot|L|)$ である．これは，アルゴリズム中の関数 location(·) は $|e|^2$ 回呼び出されるが，それぞれの呼び出しの際にすべての $l\in L$ についての確率値を比較し，それが最大になるロケーションを選ぶためである．計算量は比較的大きいが，以下の理由によりこの計算量は問題にはならない．

- イベントのサイズ（イベントに属するポストの数）はたいていの場合小さい．
- 関数 location(·) は確率値が 0 でないロケーションのみを調べればよく，ほとんどのロケーションの確率値は 0 である．

5.　評価実験

提案手法の有効性を以下の 3 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，イベント検出およ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[*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[*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

[*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
[*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．

表 1　データセット
Table 1　Dataset.

	All	*Earthquake*	*Weather*	*Tornado*	*Emergency*
# of users	508,824	141,978	307,474	15,868	81,592
# of seed users	130,391 (26%)	36,613 (26%)	78,582 (26%)	3,412 (22%)	20,043 (25%)
# of tweets	1,018,164	317,982	519,803	12,073	103,933
keywords	-	地震	大雨, 洪水, 台風，雷	竜巻	パトカー，救急車, 消防車, サイレン

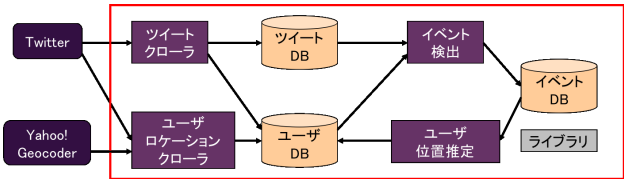

図 5　実験に用いたプロトタイプシステムの概要．Twitter から継続的にデータを収集するコンポーネントと，イベント検出およびユーザ位置推定を行うコンポーネントに分かれる
Fig. 5　The outline of the prototype system for the experiments. The prototype system consists of components that collect data from Twitter and components that perform event detection and location inference.

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目的である．

収集したツイートおよ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いったんイベント 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 $\boldsymbol{\theta_u}$ 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．ただし，本実験ではディリクレ分布のハイパーパラメータは $\alpha_k = 1$ としてある．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

ツイートクローラおよ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 2 週間に 4 つのキーワード群について行った．実験期間中に収集されたデータを**表 1** に示す．各列が 1 つのデータセット，すなわち 1 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 4 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．なお，データセットごとに提案手法の適切なパラメータは異なるため，実験ごとに設定したパラメータは異なっている．具体的な値に関しては各節に示されている．

5.3　検出されたイベントの妥当性

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れ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 4 つを用い，パラメータはそれぞれ $WindowSize = 600\,\mathrm{s}$，$MinPts = 15$，$Eps = 0.2$，$MaxDispersion = 200\,\mathrm{km}$ 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る全ツイートを 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．

- イベントに属する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．
- 複数のユーザ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．

ここで，地理的な局所性を持つイベントとは，イベントが発生した場所にいたユーザしか知りえないものであると定義した．たとえば，ある場所で起きた事件はイベントでありうるが，それがテレビなどで全国中継された後はイベントではないとした．以上の条件により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*

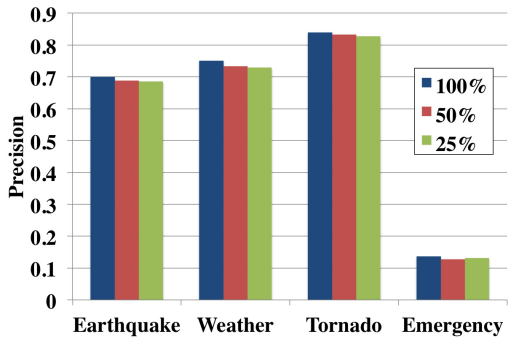

図 6　提案手法により検出されたイベントの適合率．実験期間中に多くのイベントが発生したデータセットにおける適合率が高くなっている．イベント検出に用いる位置情報の割合を大きくするとわずかに適合率が上昇している

Fig. 6　The accuracy of detected events by the proposed method. The accuracy of the datasets where a lot of events occur is high. The accuracy slightly increases with the increase of the amount of location information.

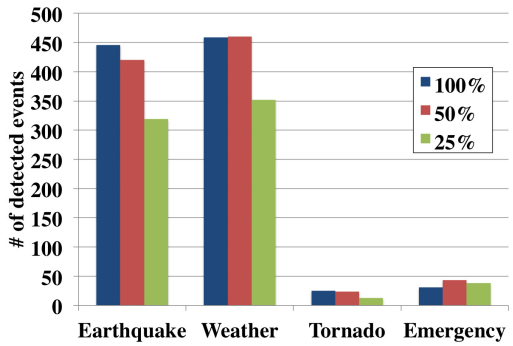

図 7　提案手法により検出されたイベントの数．実験期間中に多くのイベントが発生したデータセットにおいては多くのイベントが検出できている．イベント検出に用いる位置情報の割合を大きくすることで多くのイベントが検出されている

Fig. 7　The number of detected events by the proposed method. The proposed method detects a lot of events from datasets where a lot of events occur. The number of detected events increases with the increase of the amount of location information.

においては 0.1 程度と，非常に低い適合率を示している．これらの結果を以下に考察する．

データセットの比較　1 つは，*Emergency* データセットにおいてはイベントへの言及はほとんど見られなかったにもかかわらず，偶然発生した単語の共起によってイベントでないものを検出してしまったためであると考えられる．たとえば，“サイレン” という単語は救急車などのサイレンではなく，歌の歌詞や曲名などに出現することがあるため，比較的多くのツイートに含まれている．このようなツイートが人口の多い東京などの場所から偶然多く投稿されてしまうと，提案手法はそれをイベントとして誤検出してしまうと考えられる．実際，提案手法によってそのようなツイートの集合がイベントとして誤検出されていたのが観察された．一方で，その他の 3 つのデータセットにおいては，実際に発生した地震や雷，竜巻に対する言及がイベントとして検出されため，高い適合率を示した．

また，*Emergency* データセットにおけるイベントは，同時多発的に発生していたため，提案手法で検出できなかったと考えられる．提案手法は，Content Clustering によって検出されたクラスタのうち，dispersion の小さいもののみをイベントと見なす．しかしこの方法では，たとえば同時刻に 2 カ所以上で交通事故が発生した場合は dispersion が大きくなってしまい，イベントとして検出することができない．地震や雷，竜巻などは同時刻に複数の遠く離れた場所で起こることはめったにないが，*Emergency* データセットにおけるイベント（交通事故などの事故）は同時刻に似たようなイベントが複数の場所で起こっていたため，それらを検出することができず，精度が下がったものと考えられる．このような多峰性を持つ地理的な分布への対応は今後の課題である．

既知である位置情報の割合の比較　次に既知の位置情報のうちそれぞれ 100%，50%，25%を用いた場合の適合率の変化を調査した．その結果，割合を大きくするにつ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，位置情報の増加が検出されたイベントの妥当性向上につながることが期待される．まず，位置情報を増加させることによってイベント検出の適合率が向上した理由を以下のように考察する．Spatial Filtering において，より正確な dispersion を算出できれば，イベントとそうでないものをより正確に区別できると考えられる．そのためには，より多くのユーザの位置情報が既知である必要がある．すなわち，適合率が向上した理由は，位置情報が既知であるユーザの割合が増加し，正確な dispersion が計算できるようになったためであると考えられる．

また，検出されたイベント数が増加したのは，Spatial Filtering において，dispersion を計算できるクラスタの数が増えたためであると考えられる．Content Clustering によって検出されたクラスタ内に位置情報が既知であるユーザが 2 人以上いない場合は dispersion を計算することができない．そのため，Spatial Filtering においてそのようなクラスタはイベントではないとして破棄している．本実験では，全体として位置情報が既知であるユーザの割合が増え，dispersion を算出できるクラスタの数が増加したため，多くのイベントが検出されるようになったと考えられる．

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

表 2　精度および平均推定誤差の具体的な値
Table 2　Specific values of inference accuracies and average error distances.

	Proposed	*Li*	*Cheng*	*Clodoveu*	*Random*
Pre@160km	**0.761**	0.570	0.344	0.573	0.170
Pre@80km	**0.696**	0.378	0.284	0.520	0.092
Mean E.D. (m)	**134,114**	208,699	336,489	227,100	457,666
Median E.D. (m)	**22,862**	129,249	290,465	49,118	416,106

る Cheng らの手法 [5] [*6]，Clodoveu らの手法 [8]，Li らの手法 [16] およ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 5 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 $WindowSize = 600\,\mathrm{s}$，$MinPts = 15$，$Eps = 0.5$，$MaxDispersion = 150\,\mathrm{km}$ である．これらの値は実験的に決定した．

Cheng らの手法については，学習用のデータとテスト用のデータの 2 つを用意する必要がある．そのため，seed user の中からランダムサンプリングした 1,000 ユーザをテストユーザとし，同じくランダムサンプリングした 10,000 ユーザを学習ユーザとした．ただし，テストユーザと学習ユーザの重複は許さない．また，それぞれの学習ユーザに対して直近 100 ツイートを収集し，合計 1,000,000 ツイートを学習ツイートとする．この学習ツイートを用いて，単語のロケーション分布を学習した．さらに，それぞれのテストユーザに対して直近 3,200 ツイートを収集し，これらのツイートを基にユーザの居住地の推定を行った．

Clodoveu らの手法についても同様に，上記の 1,000 ユーザをテストユーザとした．それぞれのテストユーザについて，相互にフォローしているユーザの情報を収集し，最も多くのユーザの居住地となっているロケーションをそのテストユーザの居住地であるとした．

Li らの手法についてもまた同様に，上記の 1,000 ユーザをテストユーザとした．地名の地理的な分散を学習する必要があるため，Cheng らの手法と同様に 10,000 ユーザの直近 100 ツイートを用いた．ツイートを MeCab [*7] を用いて形態素解析し，“固有名詞” かつ “地域” である単語のみを抽出した．さらに，抽出した地名を OpenStreetMap の API [*8] を用いて緯度経度に変換した．また，テストユーザのフォロワーおよびフォロイーが必要であるため，Clodoveu らの手法と同様に収集した．

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．また，*Pre@160km*，*Pre@80km* は推定誤差 160 km および 80 km 以内であれば正解と判断したときの適合率を表す．これは文献 [5] や [16] などで用いられている指標である．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．*Median E.D.* からも分かるように，提案手法は半数のユーザを推定誤差約 23 km 以内で推定できている．また，表 2 から，*Pre@160km* を Li らの手法，Cheng らの手法および Clodoveu らの手法と比較するとそれぞれ約 34%，約 122%，約 33%の向上となった．さらに，推定誤差の平均値および中央値に関しても提案手法が最も良い結果を示している．

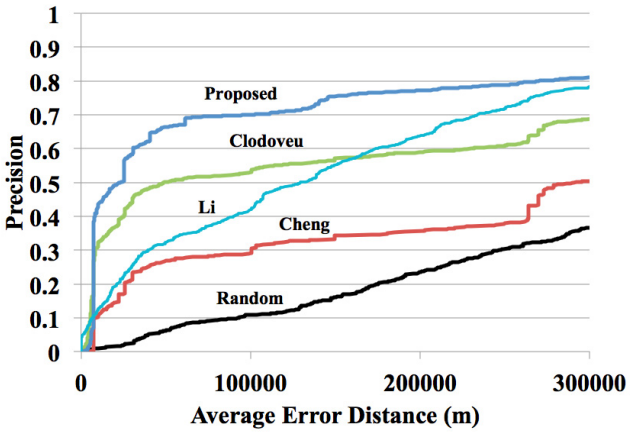

図 8　ユーザ位置推定の精度の比較結果．推定誤差 160 km での提案手法の精度を Li，Cheng，Clodoveu と比較するとそれぞれ約 34%，約 122%，約 33%の精度向上を示している
Fig. 8　The result of the comparison of inference accuracies. The accuracy of Proposed is approximately 34%, 122%, and 33% higher than that of Li, Cheng, and Clodoveu, respectively.

提案手法による推定精度（半数のユーザを約 23 km 以内の誤差で推定）であれば，1 章であげたようなアプリケーションに十分適用できるのではないかと考えられる．また，本手法によって推定したユーザ位置を用いてさらにイベント検出を行うことも可能であると考えられる．地震や雷などの影響範囲の大きなイベントに加えて，火事などの影響範囲の小さなイベントも検出できることが期待される．これは，通常は火事のような投稿が約 23 km の範囲に集中して投稿されることはないと考えられるためである．

[*6] ただし，論文 [5] 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．
[*7] https://code.google.com/p/mecab/
[*8] http://wiki.openstreetmap.org/wiki/JA:Nominatim

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 の精度は文献 [5] での評価実験における精度 0.510 より低下していることが分かる．これは，文献 [5] ではアメリカ全土における位置推定を行っているのに対し，本実験では日本を対象として位置推定を行っているためであると考えられる．文献 [5] 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．

また，現状で最新かつ最も優れている Li らの手法は Clodoveu らの手法より精度が低いという結果になった．これは，日本のユーザは地名を投稿しても実際にはその場所には住んでいないということが多いためではないかと考えられる．そのため，ユーザが自らの居住地付近の地名を投稿することが多い地域に Li らの手法を適用すれば精度が向上するのではないかと考えられる．

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，5 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 $WindowSize = 600\,\mathrm{s}$，$MinPts = 15$，$Eps = 0.5$，$MaxDispersion = 200\,\mathrm{km}$ である．

5.5.1　位置推定されたユーザ数の時間推移

5 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 3 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．したがって，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* の変化による精度と再現率の変化

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再現率の関係を検証した．再現率は各データセットに含まれるユーザのうち，どれだけのユーザのロケーションを推定できたかを表す．結果を**図 10** に示す．横軸は再現率を表し，縦軸は精度を表す．*MaxDispersion* は 50 km から 50 km 刻みで 300 km 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．

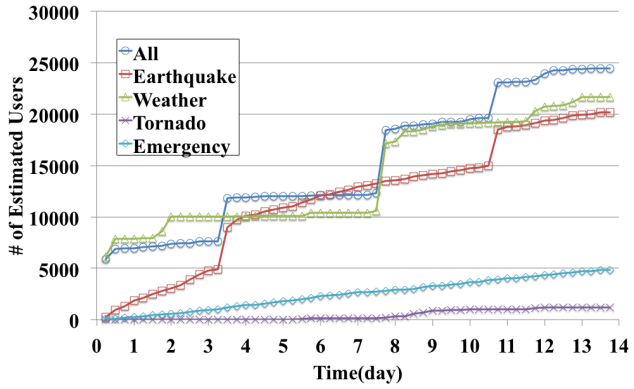

図 9　提案手法によってロケーションを推定されたユーザ数の時間推移．多くのイベントが検出されたデータセットにおいては推定されたユーザ数が大きいのが分かる．また，推定されたユーザ数が階段上に増加しているのは比較的大きなイベントが検出されたからであると考えられる

Fig. 9　The temporal transition of the number of inferred users by the proposed method. The number of users of the datasets with a lot of events appears to be large. It is considered that stepwise increases are due to large events.

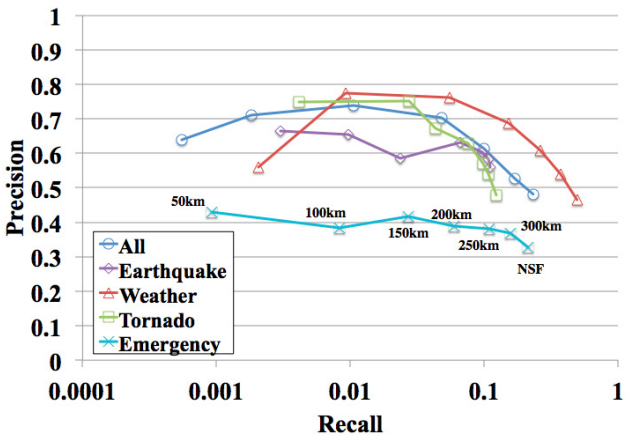

図 10　パラメータ *MaxDispersion* を変化させたときの精度および再現率への影響．精度と再現率はトレードオフの関係にあることが分かる．また，*MaxDispersion* がデータセットに対して小さすぎる場合は精度，再現率ともに小さくなっていることが分かる

Fig. 10　The effects of parameter *MaxDispersion* to the precision and the recall. We can see that there is a tradeoff between the precision and the recall.

図 10 によると，*MaxDispersion* を大きくするに従って，再現率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，それにともな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 50 km 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．

一方，他の 4 つの手法においては，推定の対象であるユーザが 1 つでもローカルワードを含むツイートを投稿していれば，もしくは 1 人でもロケーションが既知である友人がいればロケーションを推定することが可能である．そのため，他の 4 つの手法の再現率はほぼ 100%となる．すなわち，比較手法は任意のユーザの位置情報を推定することが可能であるが，提案手法と比較すると精度は劣る．一方，提案手法は精度は一番良いが，任意のユーザの位置情報を推定することはできない．これは，提案手法と他の手法とのトレードオフを表している．ただし，特定のユーザの位置情報の推定が目的ではなく，位置情報が既知であるユーザの数を増やすことが目的であれば，再現率による差異は問題にはならないと考えられる．たとえば，ローカルイベント検出や災害情報の提供，企業によるマーケティングなどでは，特定のユーザの位置情報が必要になるわけではなく，位置情報が既知であるユーザ数が多いことが重要になる．

5.6　パラメータの変化による影響

本節では，パラメータ *WindowSize* および *Eps* の値の変化にともなって結果がどう変化するかを検証する．データセットは Weather を用いており，変化させていないパラメータはそれぞれ $WindowSize = 600\,\mathrm{s}$，$MinPts = 15$，$Eps = 0.5$，$MaxDispersion = 150\,\mathrm{km}$ である．

5.6.1　WindowSize

WindowSize を 60 s から 3,600 s まで変化させて精度および効率性について調べた．結果を**図 11** に示す．横軸は与えた *WindowSize* の値を示し，縦軸は精度および効率性を示す．

結果から，*WindowSize* を小さくするに従って精度は下がり，効率性は上がることが分かる．精度が下がるのは，タイムウィンドウが小さいことによってイベントが細分化されてしまい，妥当なイベントが検出されないためであると考えられる．効率性が上がるのは，タイムウィンドウが小さいことによってある程度の割合で起こっている小さなイベントも検出されるようになっているためであると考えられる．一方で，*WindowSize* を大きくするに従って効率性は下がることが分かる．精度はある程度までは上がるが，それ以降は一定である．効率性が下がるのは，多くのイベントが結合されてしまうことによって局所性を持たなくなり，Spatial Filtering の段階でイベントではないとして捨てられてしまうためである．

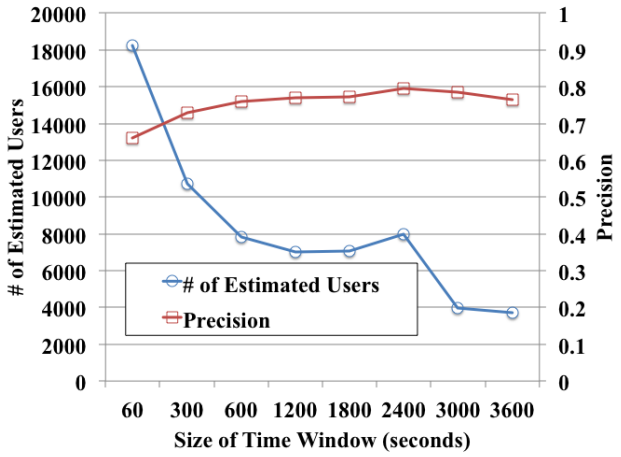

図 11　パラメータ *WindowSize* を変化させたときの精度および効率性への影響．*WindowSize* を大きくするに従って効率性は下がるが，精度はある程度までは上がることが分かる

Fig. 11　The effects of parameter *WindowSize* to the precision and the number of estimated users. With the increase of *WindowSize*, the number of estimated users decreases, while the precision increases.

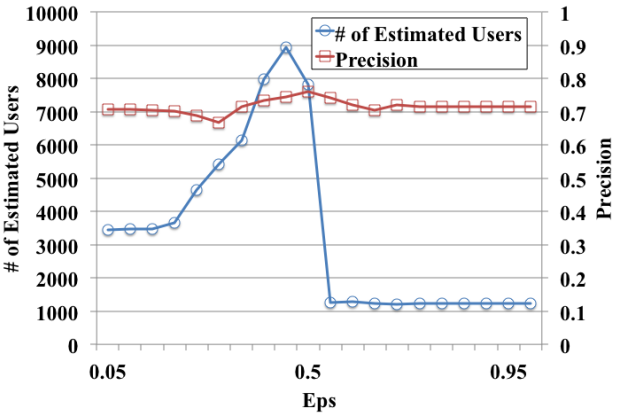

図 12　パラメータ *Eps* を変化させたときの精度および効率性への影響．精度はほとんど変化しないが，効率性は *Eps* が大きすぎても小さすぎても下がることが分かる

Fig. 12　The effects of parameter *Eps* to the precision and the number of estimated users. There is little change in the precision, while the number of estimated users changes sharply.

5.6.2　Eps

Eps を 0.05 から 1 まで変化させて精度および効率性について調べた．結果を**図 12** に示す．横軸は与えられた *Eps* の値を示し，縦軸は精度および効率性を示す．

結果から，精度はほとんど変化しないが，効率性は *Eps* が大きすぎても小さすぎても下がっていることが分かる．*Eps* が小さいときはほとんどのツイートがノイズと見なされて抽出されるクラスタが少なくなってしまい，効率性が

下がってしまう．また，*Eps* が大きいときはほとんどのツイートがまとめられて大きなクラスタが抽出されるため，*WindowSize* が大きいときと同様の理由で Spatial Filtering の段階で捨てられてしまう．そのため，検出されるイベントが少なくなってしまい，効率性が下がると考えられる．

6. 結論

本研究では，ソーシャルストリームから検出されたローカルイベントを用いてユーザ位置推定を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を示し，ユーザ位置推定の精度およ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，以下の 3 つがあげられる．まず，本稿で提案したイベント検出手法は，パラメータを手動で決定する必要がある．そのため，イベントの種類による規模の違い（地震とオープンキャンパスなど）や，地域の違いによるイベント発生頻度の違い（日本とフランスにおける地震など）に対応するにはイベントの種類や場所によってパラメータを自動チューニングする機構が必要であると考えられる．また，離れた箇所で同時に同種のイベントが発生することも考えられる．したがって，この問題に対応するには内容の類似性によってクラスタを抽出した後に地理的な類似性によってもクラスタを抽出する必要があると考えられる．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に深く感謝する．本研究の一部は（財）大川情報通信基金，および科学研究費基盤研究 C（#25330124）による．

参考文献

[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] Chandra, S., Khan, L. and Muhaya, F.B.: Estimating Twitter User Location Using Social Interactions – A Content Based Approach, *SocialCom/PASSAT*, pp.838–843 (2011).
[4] 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).
[5] 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).
[6] Cui, A., Zhang, M., Liu, Y., Ma, S. and Zhang, K.: Discover Breaking Events with Popular Hashtags in Twitter, *CIKM*, pp.1794–1798 (2012).
[7] Ester, M., Kriegel, H.-P., Sander, J. and Xu, X.: A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *KDD*, pp.226–231 (1996).
[8] Clodoveu, Jr., 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).
[9] Kinsella, S., Murdock, V. and O'Hare, N.: "I'm Eating a Sandwich in Glasgow": Modeling Locations with Tweets, *SMUC*, pp.61–68 (2011).
[10] 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).
[11] 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).
[12] Lee, R. and Sumiya, K.: Measuring Geographical Regularities of Crowd Behaviors for Twitter-based Geo-social Event Detection, *GIS-LBSN*, pp.1–10 (2010).
[13] Levandoski, J.J., Sarwat, M., Eldawy, A. and Mokbel, M.F.: LARS: A Location-aware Recommender System, *ICDE*, pp.450–461 (2012).
[14] Li, C., Sun, A. and Datta, A.: Twevent: Segment-based Event Detection from Tweets, *CIKM*, pp.155–164 (2012).
[15] 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).
[16] Li, R., Wang, S., Deng, H., Wang, R. and Chang, K.C.-C.: Towards Social User Profiling: Unified and Discriminative Influence Model for Inferring Home Locations, *KDD*, pp.1023–1031 (2012).
[17] Rattenbury, T., Good, N. and Naaman, M.: Towards Automatic Extraction of Event and Place Semantics from Flickr Tags, *SIGIR*, pp.103–110 (2007).
[18] Ritter, A., Mausam, Etzioni, O. and Clark, S.: Open Domain Event Extraction from Twitter, *KDD*, pp.1104–1112 (2012).
[19] Sadilek, A., Kautz, H.A. and Bigham, J.P.: Finding Your Friends and Following Them to Where You Are, *WSDM*, pp.723–732 (2012).
[20] Sakaki, T., Okazaki, M. and Matsuo, Y.: Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors, *WWW*, pp.851–860 (2010).
[21] Vieweg, S., Hughes, A.L., Starbird, K. and Palen, L.: Microblogging during Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness, *CHI*, pp.1079–1088 (2010).
[22] Walther, M. and Kaisser, M.: Geo-spatial Event Detection in the Twitter Stream, *ECIR*, pp.356–367 (2013).
[23] 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).
[24] Yardi, S. and Boyd, D.: Tweeting from the Town Square: Measuring Geographic Local Networks, *ICWSM* (2010).

山口 祐人　（学生会員）

2010 年筑波大学第三学群情報学類卒業．2012 年同大学大学院システム情報工学研究科博士前期課程修了．修士（工学）．現在，同研究科博士後期課程に在籍．データマイニング，情報検索等に関する研究に従事．日本データベース学会学生会員．

伊川 洋平　（正会員）

2002 年東北大学情報工学科卒業．2004 年同大学大学院情報科学研究科システム情報科学専攻博士課程前期 2 年の課程修了．同年日本アイ・ビー・エム株式会社東京基礎研究所入社．テキストマイニング，Web マイニングの研究に従事．

天笠 俊之　（正会員）

1999 年群馬大学大学院工学研究科修了．博士（工学）．奈良先端科学技術大学院大学情報科学研究科助手，筑波大学大学院システム情報工学研究科講師，同准教授を経て，現在，筑波大学システム情報系准教授．データベース，データマイニング等の研究に従事．電子情報通信学会，IEEE 各シニア会員．日本データベース学会，ACM 各会員．

北川 博之　（フェロー）

1978 年東京大学理学部物理学科卒業．1980 年同大学大学院理学系研究科修士課程修了．日本電気（株）勤務の後，1988 年筑波大学電子・情報工学系講師．同助教授を経て，現在，筑波大学システム情報系教授，ならびに計算科学研究センター教授．理学博士（東京大学）．データベース，情報源統合，データマイニング，情報検索等の研究に従事．著書「データベースシステム」（昭晃堂），「The Unnormalized Relational Data Model」（共著，Springer-Verlag）等．情報処理学会フェロー，電子情報通信学会フェロー，日本データベース学会副会長，ACM，IEEE-CS，日本ソフトウェア科学会各会員．

（担当編集委員　北山 大輔）