

情報マッピングサービスの構築事例と評判ネットワーク実装への課題

Development of regional information and Location Information Mapping System and Implementation of Reputation Network.

大塚 孝信^{1*} 鈴木 涼² 川口 将吾³ 伊藤 孝行^{1,2}
Takanobu Otsuka¹ Ryo Suzuki² Shogo Kawaguchi³ Takayuki Ito^{1,2}

¹ 名古屋工業大学 グリーン・コンピューティング研究所

¹ Center for Green Computing, Nagoya Institute of Technology

² 名古屋工業大学大学院 産業戦略工学専攻

² Master of Computer Science, Nagoya Institute of Technology

³ 名古屋工業大学大学院 情報工学専攻

³ Master of Techno-Business Administration, Nagoya Institute of Technology

Abstract: SNS services using the Internet are more popular than ever. Among them the number of users for FaceBook and Twitter are exploding. In addition, new services used photos and location information like as 'instagram', 'Foursquare' are appeared. However there is no service that users can browse to the activity and post for the information around their current location easily. Therefore, we developed the participatory system with a high regard for location information.

1 はじめに

インターネットを使用した多くのSNSサービスにおいて、位置情報を付加できるケースが増えてきている。しかし、位置情報そのものにはあまり意味がなく、特定の場所へのチェックインや評価などが主なものであり、位置情報そのものを重視したSNSというものは存在しない。[1] 本研究では位置情報を有効に機能させるため、以下のポイントを重視して作成した。

- 地図上でアクティビティが容易にわかる
- 携帯端末を用いて、いつでも投稿/閲覧できるようにする
- 特定の場所へのチェックインではなく、地図上でユーザーが自由にマッピングできる。

これにより、位置情報を身近に感じることができユーザー参加型ネットワークシステムの構築を可能とした。位置情報を主体とすることにより距離的に近いユーザー、投稿する情報が似ているユーザーの情報が見られることができ、実社会での友人関係に発展したりといったことも考えられる。始めて行くような場所、引越越し先

などの知らない土地のでは地域によって違う雰囲気を持つことが多いが、本サービスでは地図上に全ての情報がマッピングされるためどのような町かを判断することができる。また、地元の人が行くような店舗をカテゴリを問わずユーザーが投稿できるようにしたこと、今までとは比較にならない量の情報を手にすることが可能である。しかし、情報が増えた場合地図上に不要な情報が埋め尽くしてしまうといったことが考えられるため、ユーザ間の距離に基づく評判ネットワークの構築も必要である。ユーザの確度をランク付けすることにより地図上の情報を表示/非表示することで地図上の情報を適度に保つことを目標にし、評判ネットワークの提案についても行った。本論文では、提案する評判ネットワークの概要、開発したスマートフォンアプリケーション及びサーバについて述べる。

2 位置情報を考慮した評判ネットワーク

2.1 サービスの課題

本サービスでは位置情報を身近に感じることができユーザー参加型ネットワークシステムの構築を可能とした。位置情報を主体とすることにより距離的に近

*連絡先: 名古屋工業大学 グリーン・コンピューティング研究所
〒466-8555 愛知県名古屋市昭和区御器所町
E-mail: otsuka.takanobu@nitech.ac.jp

いユーザー、投稿する情報が似ているユーザーの情報を見られることができ、実社会での友人関係に発展するといったことも考えられる。始めて行くような場所、引っ越し先などの知らない土地では地域によって違う雰囲気を持つことが多いが、本サービスでは地図上に全ての情報がマッピングされるためどのような町かを判断することができる。また、地元の人が行くような店舗をカテゴリを問わずユーザーが投稿できるようにしたことで、今までとは比較にならない量の情報を手にすることが可能である。そのため、以下の問題点を克服する必要がある。

- 情報量が膨大になるため、地図上に Pin が乱立し、有用な情報が埋もれてしまう可能性がある。
- 悪意あるユーザーによる投稿の除去が必要である。

問題点を解決するためには評判ネットワークによる情報のランク付けが必要となる。

2.2 評判ネットワークの概要

本サービスではユーザをランク付けすることで情報の確度を計ることを目標としている。投稿に対するコメント数、お気に入り追加数だけではなく位置情報を用いてユーザをランク付けすることで情報の確度を求めることとしている。ウェブサイトの評判メカニズムと考えられる PageRank [9] や HITS[10][11] というアルゴリズムは、ページ間のリンク関係を利用し、値の伝搬を行うことで、各ページの重み（評判）を求めている。本手法では PageRank や HITS のアルゴリズムをユーザの重み（評判）として考えると共に、ユーザ間の距離を新たなパラメータとした。これにより、従来手法と比較した場合にユーザ情報の確度を正確に求めることを目標としている。

2.3 HITS を応用したアルゴリズムの提案

本節では、アルゴリズムで最も基本的な HITS について述べる。HITS とは Hypertext Induced Topic Search の略であり、クラインバーグらによって 1998 年に発明された。HITS は PageRank 同様ウェブページに関連した任期得点を作るのにハイパーリンク構造を用いている。しかし HITS と PageRank には重要な違いがある。PageRank は各ページに対して人気特典を 1 つ作成するが、HITS は 2 つ作成する。HITS はウェブページを権威 (authorities) とハブ (hubs) として考える。権威は沢山の入りリンクを持つページであり、ハブは沢山の出リンクを持つページである。権威とハブは次の巡回的な主張が成り立つとき良い (good) と言われて

いる。つまり、良い権威たちは良いハブたちによって指されており、良いハブたちは良い権威たちを示している。[7] また HITS にはいくつかの問題点と、多くの改良法が提案されている。[12][13] これを SNS 要素に置き換えるとユーザ i の情報がユーザ j によりシェアされた場合はユーザ i によっての権威 (authorities) でありユーザ j の情報がユーザ i によってシェアされた場合はユーザ i にとってのハブ (hubs) となる。ユーザ間の関係を Fig. 1 に示す。

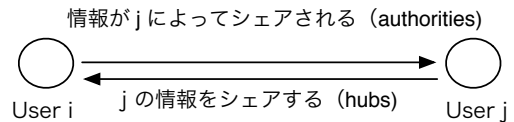


図 1: ユーザ間の関係

この関係を HITS に当てはめると以下のようなになる

$$x_i^{(k)} = \sum_{j:e_j i \in E} y_j^{(k-1)} \quad \text{と} \quad y_i^{(k)} = \sum_{j:e_i j \in E} x_j^{(k)}$$

本アルゴリズムでは HITS の単純なユーザ関係に加え、ユーザ間の現実的な距離 (Distance) を考慮することとしている。これは単純にユーザ間の現実的な距離が小さい場合は現実世界でユーザ同士が顔なじみである可能性が高いと仮定しているため、通常の友人関係での情報のやり取り同様に重要ではない情報をシェアする事が多く想定される。対してユーザ間の現実的な距離が離れている場合でも情報をシェアし合う仲と仮定し、現実的な距離が近い場合に比べ有益な情報が多く存在していると仮定しているためである。ユーザ間の現実的な距離の例を Fig. 2 に示す。

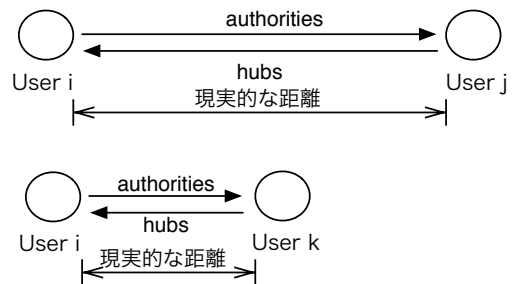


図 2: ユーザ間の現実的な距離

この場合では現実的な距離が近いユーザ i,k と比較しユーザ i,j では現実的な距離が遠いため情報の重みを距離によって考慮する必要がある。そこで、ユーザ間の現実的な距離を d として用いて HITS アルゴリズムに挿

入し, *Distance-HITS*として構築した, *Distance-HITS*は以下の数式によって表される.

$$x_i^{(k)} = \sum_{j:e_j i \in E} dy_j^{(k-1)} \quad \text{と} \quad y_i^{(k)} = \sum_{j:e_i j \in E} dx_j^{(k)}$$

これにより, 従来リンク/被リンクのみの単純な順位付けであったものをユーザ間の現実的な距離を考慮することによりリンクの重みを付加することができ, 従来手法と比較した場合にユーザ確度のランク付けをより正確に行うことが可能であると考え.

2.4 PageRank を応用したアルゴリズムの提案

GoogleがPageRankは「多くの良質なページからリンクされているページは, やはり良質なページである」という再帰的な関係をもとに, 全てのページの重要度を判定している. PageRankとは単純な総和公式, その源は学術誌の間での論文参照構造の分析にさかのぼる公式である. [8] ページ P_i のPageRankは, $r(P_i)$ と書くが, P_i を指している全てのページのPageRankの総和となる. ここで, B_{P_i} は, P_i を指すページ(バックリンク)の集合であり, $|P_j|$ はページ P_j からの出リンクの個数である. この際, ページ P_i の入リンクとなるページのPageRankである値 $r(P_j)$ が未知であるが, 反復法を用いて解決している. すなわち, 最初に全てのページが同じPageRankの値(ウェブインデックスにあるページの個数を n として, $1/n$)を持つと仮定する. そこでインデックスの各ページ P_i について $r(P_i)$ を計算する. それらを繰り返し計算することにより算出することができる. 計算式を以下に示す.

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|}$$

この手続きはすべてのページ P_i に対して, $r_0(P_i) = 1/n$ として開始され, PageRankの得点が最終的には安定した値に収束するものと期待され繰り返される. 図3のような6つのインデックスのページを計算した場合次のような有向グラフが形成される.

ここまでのPageRankの仕組みであるが, *Distance-HITS*と同じくウェブページのランク付けをユーザの評価とした上でユーザ間の現実的な距離情報を付加する. これにより以下計算式となる.

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \left\{ \frac{r_k(P_j)}{|P_j|} + \alpha d(P_i, P_j) \right\}$$

単純に d を足すだけではなく α を挿入することにより, パラメータの設定を容易としている. パラメータ

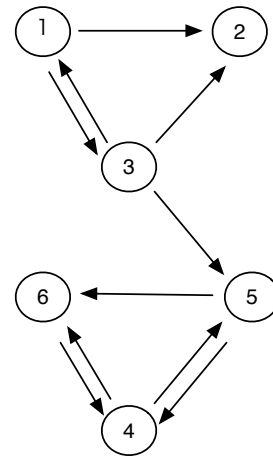


図 3: ウェブページの有向グラフ

については評価実験を含めて実施する際に最適な値を模索していくこととしている.

3 アプリケーションの開発

3.1 モバイルアプリケーションの開発

アプリケーションはiOS及びAndroid向けに開発を行っている. 開発当初はiOS及びAndroidデバイス双方に対応させることが容易なTitanium環境にて構築していた. しかし, Titanium環境ではJavaを介しての動作となるため動作が遅く, 各機種への最適化も困難な状況であったためiOSについてはObjective-C, AndroidについてはJavaを用いるNative環境での開発へとスイッチした. [5] [6] 本項目ではiOSアプリケーションに焦点を当てて詳細を記述する. 開発するにあたり以下の点を考慮した.

- GPSによる現在地取得情報
- サーバーへの通信機能
- カメラコントロール機能
- ソーシャルメディア連携機能

GPS機能についてはCLLocationManagerを用いてiOSデバイス内のGPSユニット及からの情報と周辺の基地局, WiFiアクセスポイントからの情報を得ている. CLLocationManagerでは現在位置の緯度, 経度, 位置精度, 位置情報取得間隔を設定することが可能である. また, ユーザーが直感的に位置情報を把握できるよう緯度, 経度から住所情報へ変換するリバーズジオコーディングについても実装している. 地図データに関して

は現状、GoogleMap API を用いている。GoogleMap は規約によりデバイス単体にキャッシュを持つことができない。そのため、地図情報の取得が通信環境に依存しやすく最適な速度を得られないことも多いことから wikipedia と同様にユーザーの情報によって作成される OpenStreetMap への移行を行うこととしている。OpenStreetMap は道路地図などの地理情報データを誰でも利用できるよう、フリーの地理情報データを作成することを目的としたプロジェクトのため、地図情報をユーザーや運営側で自由に変更することが可能である。

投稿データについては地図上にオーバーレイすることで表示させるが、投稿情報が増えた場合はレイヤ構造することで各情報別の表示やオーバーレイ表示などが容易にできる構造とする必要があるが将来的には情報にタグ付けすることを目標としている。サーバーへの通信機能については iOS SDK を用いて実装している、Native 環境の利点として通信間隔の設定ができ、電池持続時間に貢献することができる。投稿時の補足として画像を添付したい要求があると考え、内蔵カメラのコントロール機能についても搭載している。カメラコントロールは UIImagePickerControllerContoroller を用いている、UIImageController では撮影した画像の拡大縮小の編集機能を持っており撮影した画像を最適なサイズにリサイズすることで通信時とサーバーへの負担を減らすことができる。加えて、カメラが非搭載であるデバイスにも使用できるように isSourceTypeAvailable:souceType メソッドによりカメラ搭載の有無を確認している。また、カメラ機能には多くのメモリが必要とされ、メモリ管理が重要であることが判明した。試作初期ではシングルスレッドによる動作待ちや、メモリ不足によるエラーが多く発生していた。そのため、GrandCentralDispatch によるマルチスレッド化を行うことで良好な動作速度を実現した。GrandCentralDispach を用いることでマルチコア CPU も用いることにより iPhone4S のような最新のデバイスでは申し分ない速度を実現することができた。ソーシャルメディア連携機能については計測させられているという気持ちをユーザーが抱かないように実装した。Twitter, FaceBook への自動投稿機能、ユーザー間の投稿に対するコメント機能を実装した。各サービスへのユーザー認証方法は Twitter には OAuth を、FaceBook には公式 API を用いている。[4] 以下に選定した理由を述べる。Twitter のユーザー認証には Basic 認証, XAuth 認証及び OAuth 認証の 3 種の方法がある。Basic 認証は、暗号が平文により送信されるため盗聴や改竄が容易であるという欠点を持つ。XAuth 暗証はアプリケーションにユーザー名、パスワードを入力するのでサービス提供側にユーザー名とパスワードを悪用される恐れがある。OAuth 認証では Twitter サービスが認証を行い、アプリケーションのアクセス許可を取得するため安全である。よって本システムでは

OAuth 認証を利用して連携を行っている。FaceBook との連携は FaceBook 公式 API が提供するため、そちらを利用している。アルゴリズムについては Twitter の OAuth と同様の仕組みを用いており、サービスが許可したアプリケーションにアクセス権を与える方法をとっている。これらにより周囲の皆で作る環境マップとして共有意識を高め、ユーザーが積極的に利用できるような仕組みを実現している。

アプリケーションのアーキテクチャを Fig. 4 に示すと共に、投稿情報の表示を Fig. 5 に示す。情報の有無は地図上に Pin で示され、Pin をタップすることで詳細情報の閲覧が可能となっている。

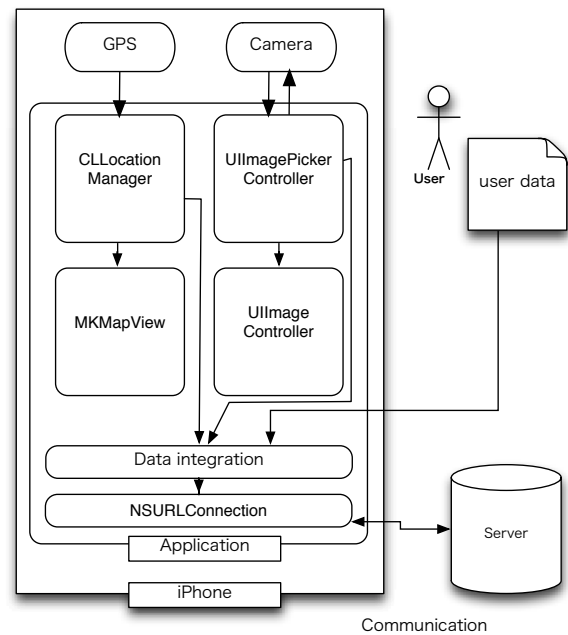


図 4: アプリケーションのアーキテクチャ

3.2 投稿の流れ

モバイルアプリケーションによる投稿のフローを実際の投稿画面により説明する。

1. アプリケーションは現在地情報を取得し、現在地を画面中心として地図を表示する
2. 画面下部のカメラアイコンをタップすることで撮影画面へ遷移し、写真を撮影する。この際、本サービスにログインしていない場合はログインを求められる。
3. 撮影した画面とともにタイトルとコメントを自由に入力することができる。



図 5: 投稿情報の表示

4. 投稿ボタンを押すことでサーバーへ情報をアップロードする。
5. 現在地に投稿した情報があることを示す pin が描画される。

3.3 閲覧時の機能

モバイルアプリケーションでの情報の閲覧方法を説明する閲覧時に利用できる機能については3つあり

- 最新の投稿を閲覧する。
- 現在地周辺の投稿を閲覧する
- 地図上の好きな場所の投稿を閲覧する。
- 自分によりコメントした投稿, 他のユーザーによりコメントされた投稿を閲覧する。
- お気に入りに追加した投稿を閲覧する。

ことが可能である。以下に各機能についての詳細を示す。

1. 最新の投稿を表示する際は右下のボタンをタップすることで最新の投稿を時間軸に沿って閲覧することができる。投稿をタップすることで詳細画面

に移ることができ、コメントや位置情報が閲覧できるようにしている。

2. 現在地周辺の投稿を閲覧する際は右下のコンパスボタンをタップすることで現在地にスナップすることができる。最新の投稿とは違い、地図上の任意の pin をタップすることで詳細情報を見ることができる。また、地図上に表示される pin の詳細情報を一覧として閲覧することも可能としている。
3. 地図上の好きな場所の投稿を閲覧する際は map 上の好きな位置へタップすることで自分の好きな場所の投稿を閲覧できるようになっている。地図をピンチアウトすることで今いる地域の情報はもとより、世界中に投稿された情報を自由に閲覧することが可能である。
4. 画面左下のボタンをタップすることで自分がコメントした投稿, 他のユーザーにより投稿がある投稿をリスト表示することができる。未読のコメントがある場合はボタンに未読件数が表示されるようになっており、ユーザー同士のコメントを楽しめる設計としている。
5. ユーザーは自分の気に入った投稿に対し、お気に入りボタンをタップすることでお気に入りに登録することができる。お気に入りの投稿はリスト表示で閲覧することができる。

3.4 サーバーアプリケーションの開発

本サービスはユーザーサイドのアプリケーションにより投稿された情報をサーバーにより処理し格納している。サーバーサイドについては当初は My-SQL を用いて開発していたが、処理速度の向上とユーザー数が増加した際のスケールを確保するためにユーザーのログイン情報を格納するキャッシュサーバには KyotoTycoon、データ格納には MongoDB を利用したシステムとしている。MongoDB はバイナリファイルの格納に適しており、投稿はジオロケーションをインデックスとしたひとつの集合体として扱われている。従来の JSON 構造と似ているが、BSON 記述によりバイナリファイルを格納可能でありオブジェクト指向の実装が可能である。キャッシュサーバの KyotoTycoon は多数のデータベースにアクセスすることが可能な軽量かつ高速なデータベース管理環境であり、処理を分散させることで多数のリクエストを高速に処理することが可能である。実装の初期段階より My-SQL サーバでは負荷が高すぎ、要求するパフォーマンスが得られなかったが KyotoTycoon と mongoDB をチューニングすることにより要求する

パフォーマンス以上の性能が実現できた。サービスの成熟と共にユーザー数の増加が見込まれるが、本システムを利用している限り処理サーバを並列に増加させることでスケールを確保できる。格納されたデータは位置情報と投稿情報が一つのパッケージとなって格納されており、日付での並べ替えにも対応しているため時系列でのデータ管理を可能としている。これにより投稿情報をサーバー上から容易にソートすることが可能となっている。

4 おわりに

本稿では、位置情報を重視した情報共有システムの開発について提案した。位置情報を自由に投稿、閲覧できることにより地域情報の共有が簡単に行えるよう設計した。また、自由に投稿できるシステムのデメリットについても理解しており、ユーザ情報の確度に基づき表示情報を選択することにより利便性を向上させる予定である。また、ユーザ間の現実的距離も考慮することで情報確度の正確さを表すこととしている。現実的距離を用いたユーザ確度の評価については今後も継続して研究を行い、最適なパラメータを設定することとしている。今後は AppStore での一般公開を目標に開発を進めるとともに、位置情報を用いた研究目的に活用できるよう API 公開も含めて検討していく。

参考文献

- [1] Scott Counts , Karen E.Fisher, “ Mobile social networking as information ground:A case study, Microsoft reseach, ” 2010 p98-115
- [2] T. R. Oke, “ The energetic basis of the urban heat island., Quarterly Journal of the Royal Meteorological Society, ” 1982 Vol.108 P1-24
- [3] C. P. Lo, D. A. and Quattrochi J. C. Luvall, ” Application of high-resolution thermal infrared remote sensing and GIS to assess the urban heat island effect., ” International Journal of Remote Sensing, 1997 Vol.18
- [4] David Recordon, Laurie Rae, Chris Messina, “ OpenID: The Definitive Guide: Identity for the Social Web, ” 2010 O’Reilly Media
- [5] Bill Dudney , Christopher Adamson, “ iPhone SDK Development, ” 2009 Bookshelf
- [6] Rick Rogers, John Lombardo, Zigurd Mednieks, G. Blake Meike, “ Android Application Development: Programming with the Google SDK, ” 2009 O’Reilly
- [7] Amy N.Langville, Carl D.Meyer 著, 岩野 和生, 黒川 利明, 黒川 洋 訳, “ Google PageRank の数理 - 最強検索エンジンのランキング手法を求めて, ” 2009 共立出版
- [8] Taher H. Haveliwala, “ Efficient Computation of PageRank, ” 1999 Stanford Technical Report
- [9] S.Brin and L.Page, “ The anatomy of a large-scale hyper textual web search engine, ” WWW7/Computer Networks, vol.30, no.1-7), pp.107-117, 1998
- [10] L. Li, Y. Shang, and W. Zhang, “Improvement of hits-based algorithms on web documents, ” Proceedings of WWW2002, pp.527-535, 2002
- [11] K.Bharat and M.R. Henzinger, “ Improved algorithms for topic distillation in a hyperlinked enviroment., ” Proceedings of the 21st Annual International ACM SIGIR Confarence on Research and Development in Information Retrival, pp.104-111, 1998
- [12] J.Kleinberg, “ Authoritative sources in a hyper-linked environment, ” Journal of the ACM, vol.46, no.5, 1999
- [13] 手塚 友, 浅野 奉仁, 西関 隆夫, “ 現在の web における hits について, ” 電位情報通信学会技術研究報告. COMP, コンピューテーション, vol.105, no.679, pp39-46, 2006.