博士論文

拡散方程式を用いた位置情報SNSからの時空間パターンの抽出
– 推薦システムへの応用を目指して –

西岡　賢一郎

目 次

第1章　序章

1.1　位置情報SNSとチェックイン

近年、スマートフォンやウェアラブルデバイスの普及により、多くのユーザがGPSを備えたデバイスを持ち歩くようになり、ユーザが屋外で、様々なアプリケーションを利用できるようになった。そのようなアプリケーションの中で、位置情報SNSと呼ばれるサービスがある。位置情報SNSは、スマートフォンなどのGPSを用いて、自分の行った場所を記録したり、記録した場所を友人などと共有したりできるサービスである。

位置情報SNSでは、ポイントシステムなどを導入しゲーミフィケーションをもたせたFoursquare[1]、Facebookの一機能であるFacebook Places[2]、渋滞や交通取り締まりの情報を共有するWaze[3]などが有名である。これらの位置情報SNSを用いることで、ユーザは自分の行った位置の情報を記録したり、他の人の記録した位置の情報を簡単に知ることができるようになった。位置情報SNS上で登録されている(無い場合は追加で登録することもできる)施設に行った時に、位置情報SNSを使ってその施設に行ったことを記録することを、一般にチェックインと呼ぶ。チェックインには、チェックインした施設の緯度・経度などの位置情報が含まれており、このチェックインデータを用いた研究も盛んになってきている[29]。

位置情報SNSが登場する以前の位置情報を用いた研究では、専用のGPSトラッカーデバイスをユーザに持たせて、位置情報をトラッキングしてデータを集め、GPSの軌跡データを使用していた[22][41]。しかし、GPSトラッカーを被験者に持たせるやり方だと、被験者に専用のデバイスやアプリケーションをデータ収集の期間使ってもらう必要があり、

[1] https://foursquare.com/
[2] https://www.facebook.com/about/location
[3] http:/www.waze.com/

大量のユーザのデータを得ることが難しかった。

位置情報 SNS では、位置情報 SNS が提供している API などを利用して、ユーザの位置情報を得ることができるため、多種多様なユーザの位置情報を得ることができる。これにより、位置情報 SNS のデータを用いて、POI (Point Of Interest) [32] や、交通情報 [7] や、異なる時間尺度からパターンを抽出する方法 [16] や、個人の行動を抽出する方法 [11] などが研究されている。また、抽出された情報を、交通安全 [13] のアプリケーションなどに、活かすような技術も出てきている。

位置情報 SNS の 1 つであり、チェックインに特化したサービスである Foursquare は、チェックインを Twitter[4]に共有する機能があるため、Twitter 経由で Foursquare のチェックインデータを取得することができ、多くのユーザのチェックインの取得が容易なため、チェックインの研究として用いられている [29]。

Twitter 経由で取得した Foursquare の東京周辺での 1ヶ月間のチェックインを用いて、表 1.1 にチェックインの統計情報を、図 1.1、図 1.2、図 1.3 にそれぞれユーザごとのチェックインの回数、連続するチェックイン間の時間、連続するチェックイン間の距離の分布を示した。表 1.1 より、チェックイン回数の平均は 1ヶ月で 29.71 回になっており、ユーザ 1 人の 1 日あたりのチェックイン回数は、平均 1 回と言える。これより、Foursquare から取得できるチェックインは、かなりスパースなデータであることが分かる。チェックインの中央値を見てみると、中央値は 6 回となっており、平均値 29.71 回と比べかなり小さくなっており、多くのユーザは平均値よりも少ない回数しかチェックインをしていないようである。チェックイン間の時間・距離の平均値と中央値もそれぞれ 0.89 日、0.1 日、36.44km、3.55km と大きくずれていることがわかる。図 1.1、図 1.2、図 1.3 からも、ユーザによってチェックイン傾向の偏りがあるロングテールの構造となっていることが分かる。

本論文では、この Foursquare のチェックインを使って、主に

- 時空間パターンの抽出
- 位置予測

の 2 つを可能にする手法を提案する。

[4]https://twitter.com/

	チェックイン回数	チェックイン間の時間 (単位: 日)	チェックイン間の距離 (単位: km)
mean	29.71	0.89	36.44
std	78.68	2.11	282.96
min	1	0	0
25%	1	0.02	0.39
50%	6	0.10	3.55
75%	27	0.81	16.52
max	3190	29.60	16309.82

表 1.1: Foursauqre ユーザの 1ヶ月のチェックインの回数、間隔、距離の統計情報

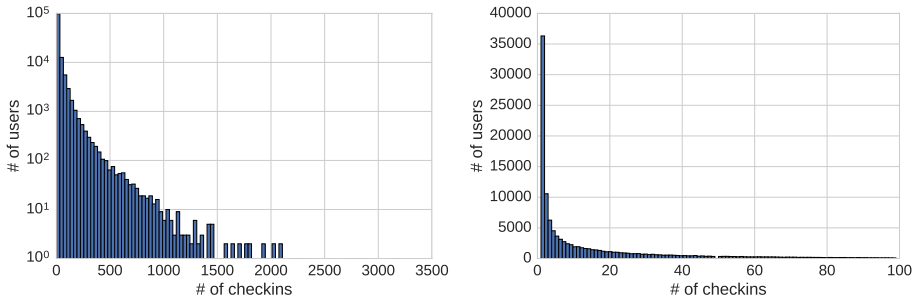


図 1.1: Foursquare ユーザの一ヶ月のチェックイン回数の分布

1.2 時空間パターンの抽出

1.2.1 全ユーザの時空間パターンの抽出

GPS トラッキングデータおよび位置情報 SNS データなど、ユーザの位置情報を含んだデータには、様々なユーザの趣味嗜好や日常の行動パターンが含まれており、これらのデータセットからユーザの行動パターンを抽出することは、ユーザの将来行動の予測、位置情報 SNS のサービスの質の向上、また位置情報データ自体の管理方法の改善などに有用だと考えられている [18][5][4][27]。

位置情報 SNS から抽出できるユーザの行動パターンには、ユーザがよくカフェに行くというような、カフェなどのカテゴリを用いたパターンや、ユーザが特定の座標によく行くというような緯度経度を用いたパターンなどがある。前者は、特定地域でチェックインするときに候補となる場所リストのフィルタリングに使え、後者は、ユーザが生活する範

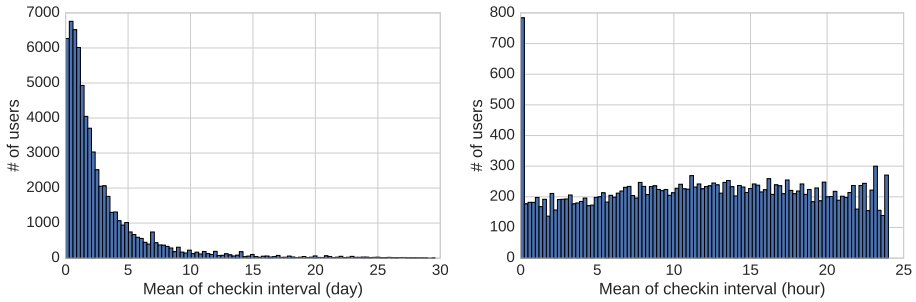


図 1.2: Foursquare ユーザの一ヶ月のチェックイン間隔の分布

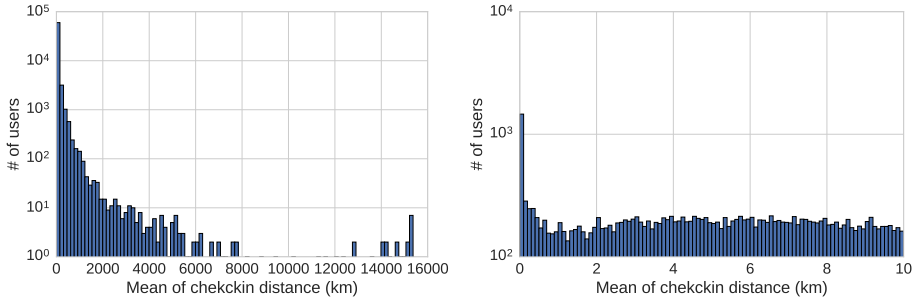


図 1.3: Foursquare ユーザの一ヶ月のチェックイン間の距離の分布

囲の推定に使えると考えられる。

本論文では、時刻・位置 (本論文では緯度と経度) のみに着目し、複数のユーザがどの時刻にどの位置にいる傾向があるかをパターンとして抽出する。このパターンは、時間と空間に関するパターンで時空間パターンの一種である。

本論文における時空間パターンとは、時刻と位置におけるユーザのパターンとし、主に以下の 3 つのパターンを示すとする。

- 定常的なパターン: 時刻に影響されないユーザのパターン (例: ユーザが都心部で生活している)
- 周期的なパターン: 周期的にユーザが集まったりいなくなったりするパターン (例: 通勤時間帯に駅周辺の人が増える)
- 突発的なパターン: 周期性も定常性もなく、特定日に突然あらわれるパターン (例:

イベントなどでユーザが急激に集まる)

一般に、GPS トラッキングデータや位置情報 SNS データには、このような時空間パターンが重なり合っており、各々の時空間パターンは元のデータから分離する必要がある。

さらに、位置情報 SNS では、特定時刻の特定場所のチェックインを保存しているが、チェックインした時刻以外に、ユーザがどこにいたかが分からないため、データが GPS トラッキングデータに比べ、かなりスパースであるという欠点がある。このスパース性により、位置情報 SNS のデータからの時空間パターンの抽出は、さらに難しくなっている。従って、位置情報 SNS における時空間パターンの抽出のために、まずチェックインとチェックインの間にユーザがどこにいたか推定する。

チェックインとチェックインの間にユーザがどこにいたかを推定するために、二次元の拡散方程式[5] $(2\pi t)^{-1}\exp\left(-|\boldsymbol{x}-\boldsymbol{y}|^2/2t\right)$ を用いて、ユーザのチェックイン間の位置を時空間分布として推定する。時空間分布は、特定の時刻にユーザがどの位置にいるかを確率分布で表す。

拡散方程式を用いるに当たって、チェックインが 1 次マルコフ性を持つと仮定した。拡散方程式は、ガウシアン放射基底関数と似ているが、ガウシアン放射基底関数が距離のみに依存し、時間とともに変化しないのに対し、拡散方程式は時間とともに広がる分布となっている。そのため、時間によって広がっていく物質の動きを表すために、量子力学 [12] や、地震学 [34] などでよく使われている。本論文におけるユーザの時空間分布についても、チェックイン後に時間が経てば経つほどユーザの移動可能な範囲が広がるため、時間と共に広がっていく拡散方程式がモデルとして適切だと考えられる。

拡散方程式は、そのままではユーザの時空間分布が時間とともに広がり続けてしまい、その後のチェックイン時に、ユーザがチェックインした位置にいたという条件を満たせなくなってしまう。そこで、本論文ではチェックイン時刻に必ず、ユーザがその時刻にチェックインした位置に必ずいるような制約を満たすようにモデルを構築する。

また、拡散方程式には、物理的な定数であるスケールパラメータが含まれている。スケールパラメータは、ユーザの移動速度と関係し、データによって異なってくると考えら

[5]本来二次元の拡散方程式は $\frac{\partial\phi}{\partial t}=K(\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2})$ のような微分方程式のことであるが、ここでは無限に広く他の制約がないときの解を用いるので、簡単のために解自体を拡散方程式と呼んでいる。

れるため、位置情報 SNS のチェックインデータから統計的に推定する方法を示す。

次に、推定した時空間分布からユーザの時空間パターンを抽出するために、時空間分布を離散化し、PCA (Principal Component Analysis) と ICA (Independent Component Analysis) を適用し、時空間分布から時間成分と空間成分を分解する。これにより、重なり合った時空間パターンから、定常的なパターン、周期的なパターン、突発的なパターンを抽出する方法を示す。

全ユーザのチェックインデータから時空間パターンを抽出する際に PCA を用いると、全ユーザに共通する大域的なパターンが抽出される。大域でないパターンも抽出するために ICA を用いた時空間パターンの抽出も行う。

まとめると、全ユーザのチェックインからの時空間パターンの抽出のために、本論文では以下の方法を提案する。

1. チェックイン間のユーザの位置を拡散方程式を用いて推定 (スパース性の解決)
   (a) チェックインが 1 次にマルコフ性を持つと仮定し、チェックイン時刻にチェックイン位置にいるように拡散方程式を用いてユーザの時空間分布を推定
   (b) チェックインデータから拡散方程式のパラメータであるスケールパラメータを推定
2. 推定した全ユーザの時空間分布から ICA を用いて、ユーザの時空間パターンを抽出 (時空間分布からの時空間パターンの抽出)

また、1b におけるスケールパラメータは、チェックインの外れ値の影響を大きく受けるため、チェックインの外れ値の影響を抑えた頑健な推定とするため事前分布を用いた推定方法についても提案をする。

1.2.2 類似ユーザの時空間パターンの抽出

全ユーザのチェックインデータから時空間パターンを抽出すると、全体的な傾向を見ることはできるが、ユーザが普段生活している範囲によって異なる時空間パターンを抽出す

ることが難しい。そこで本論文では、全ユーザの時空間パターン抽出よりも、一人のユーザに関係した時空間パターンを抽出するため、類似した行動範囲のユーザを集めて時空間分布を作成し、より一人のユーザに関係した時空間パターンの抽出を行う。

まず、類似しているユーザの定義を、同じ時刻に近い位置にいるユーザとする。これを、各ユーザに対して定義された時空間分布の類似度としてヘリンジャー距離を用いることで具体化した。ヘリンジャー距離はユークリッド距離であるため、分類でよく使われるWard 法を用いることができる。

ヘリンジャー距離で定義されたユーザ間の類似度を使って、Ward 法を用いてクラスタリングを行い、クラスタごとに時空間パターンの抽出を行う。

ここでは、各々のクラスタの時空間パターンを抽出するために、ICA ではなくて PCA を用いる。

類似したユーザの時空間パターンでは、1.2.1 節の手法に加えて以下を提案する。

3. 推定されたユーザの時空間分布を用いて、ユーザ間の距離 (類似度) を計算

4. 距離が近いユーザを集めてクラスタを形成し、類似したユーザの特徴を反映したユーザの時空間分布を推定。推定した時空間分布に PCA を適用して、時空間パターンを抽出

1.3 特定周期の時空間パターンを用いたユーザの位置予測

位置情報 SNS のデータから、ユーザの位置を予測する研究も盛んである [39][9]。ユーザの位置予測も時空間パターンと同様に位置情報 SNS のサービスの質の向上などに使えると考えられる。ユーザの位置予測では、チェックイン時のコメントなどの付加情報を使った研究 [39] やチェックインの位置と時刻のみを利用した研究 [9] などがある。

本論文で時空間パターンの抽出にて定義した各々のユーザの時空間分布は、ユーザが特定の時刻にどこにいるかを確率分布として表しているため、ユーザの未来の位置の予測にも適用できると考えられる。そこで、ユーザの時空間分布を使った、ユーザの未来の位置の予測モデル DPM (Diffusion-type Periodic Model) を提案する。

Foursquare のチェックインデータでは、ユーザ個々のチェックイン数が少ないので、例えば、たまたま旅行中に行ったチェックインなどが、予測モデルの学習において外れ値となる可能性がある。そこで、予測モデルを構築するにあたって、対象となるユーザのチェックインデータから時空間分布を推定するのではなく、類似したユーザのチェックインデータを集めて時空間分布を推定する、対象のユーザのチェックインデータの外れ値の影響を抑えた手法 DPMU (Diffusion-type Periodic Model with similar Users) を提案する。

更に、時空間分布に PCA を適用し、寄与率の低い成分を除くことで次元削減をしてノイズを減らす RDPMU (Reduced Diffusion-type Periodic Model with similar Users) を提案する。

以上をまとめると、本論文では、位置予測に対し以下のような手法を提案する。

5. 対象のユーザのチェックインデータから推定した時空間分布を用いた予測モデル DPM
6. 類似したユーザのチェックインデータから推定した時空間分布を用いた予測モデル DPMU
7. 時空間分布の PCA から寄与率の低い成分を除いて次元削減をした予測モデル RDPMU

評価指標としては、log-likelihood と mean rank (Rank) と mean reciprocal rank (MRR) の三種類を使用した。実験の結果、log-likelihood 及び Rank では DPM より DPMU を用いることで精度が向上し、MRR では逆に DPM の方が精度が良かった。RDPMU では精度の大きな向上はなかった。

1.4 論文構成

本論文では、まず第 2 章にて先行研究について説明をする。第 3 章にて、拡散方程式を用いて時空間分布を推定し、その時空間分布から PCA と ICA を用いて、時空間パターンを抽出する手法について説明する。第 4 章では、第 3 章で使用した拡散方程式のスケールパラメータの事前分布を用いた頑健な推定方法を導入する。第 5 章では、ユーザ間の距離

を定義し、その距離を使用したクラスタリングによりユーザをクラスターに分け、分けられたクラスターからユーザの時空間分布を推定し、推定した時空間分布から PCA で時空間パターンを抽出する。第 5 章で説明した手法で推定した時空間分布、時空間パターン、ユーザ間の距離を用いて、ユーザの位置予測を行う。第 6 章では、時空間パターンの抽出に用いた時空間分布を用いたユーザの未来の位置予測モデルを提案する。第 7 章では、本論文のまとめと今後の展望について議論する。

1.5 用語集

本論文で用いる重要な用語を以下にまとめた。

位置

緯度と経度の組み合わせでできる座標。本論文では狭い範囲のチェックインデータを用いるため簡単のために緯度と経度を二次元平面の座標として扱う。

位置情報 SNS

スマートフォンなどで利用できるユーザが自分の行った場所などをログできるサービス。様々な種類のサービスが存在するが、Twitter 経由でデータ取得をしやすい Foursquare を本論文のデータとして利用する。

チェックイン

ユーザが位置情報 SNS を用いて、特定時刻にどこの施設にいたかをログとして残すこと。チェックインデータにはユーザの緯度と経度、施設情報などが記録されている。本論文では緯度と経度のみを利用している。

時空間分布

時刻と位置により決まるユーザの存在確率を表す確率分布。

チェックイン間の位置推定

チェックイン間のユーザの位置を推定すること。チェックインとチェックインの間は、データ存在がしないためユーザがどこにいたかがわからない。本論文では拡散方程式を用いて、チェックイン間にユーザがどこにいたかを推定する。

拡散方程式

拡散現象を表す微分方程式。本論文では二次元の無限の領域の制約のない拡散方程式の解である $(2\pi t)^{-1} \exp\left(-|\boldsymbol{x}-\boldsymbol{y}|^2/2t\right)$ を利用している。簡単のため、この式を拡散方程式と呼ぶ。ユーザの時空間分布を推定するために、ユーザの移動モデルとして利用する。

スケールパラメータ λ

時空間分布の推定で使用する拡散方程式のパラメータ。ユーザの移動距離の二乗の時間変化の逆数の次元を持つ。λ が大きいほど平均移動速度が速い。各ユーザのチェックイン数が少ないため、全ユーザのチェックインデータから適切な値を推定する。

時空間分布の推定

チェックインデータとスケールパラメータ λ により行う、ユーザごとの時空間分布の推定。

時空間パターン

時空間分布のパターン。本論文では、定常的な時空間パターンや、周期的な時空間パターン、突発的に現れる時空間パターンを考える。

位置予測

ユーザの未来の位置の予測。チェックイン間の位置を推定する位置推定と区別する。拡散方程式を用いて推定したユーザの時空間分布のままでは、末尾のチェックインの後に時空間分布が広がり続けるため、本論文では周期性を利用した未来の位置予測の方法を提案する。

第2章　先行研究

2.1　時空間パターンの抽出

一般のパターン抽出の手法としては、Trajectory Mining や Discrete Sequence Mining [1] などが有名である。Trajectory Mining は、GPS トラッキングデータなどから移動の軌跡のパターンを抽出する手法である。Discrete Sequence Mining は、遺伝子情報やサーバのトランザクションのログなど非連続な列から、パターンを抽出する手法として広く使われている。Discrete Sequence Mining は、Sequential Pattern Mining とも呼ばれ、頻度の高いシンボルの列や、長いシンボルの列をデータから探す手法として使われている。これらの手法は、細かい間隔で取得されたデータやシンボルがついたデータには有効であり、幅広い分野で使われている。

しかし、位置情報 SNS のチェックインは、ユーザがログを残したいタイミングでしかユーザの位置情報のデータがないため、スパース性が非常に高い。また、チェックインした施設のカテゴリは、施設のカテゴリ情報をユーザが設定できるため、カテゴリが付いていなかったり、同じ様な施設でもカテゴリの付け方が異なることがある。このため、Trajectory Mining や Discrete Sequence Mining によって、チェックインからパターンを抽出することが難しくなっている。チェックインデータを時間または空間もしくは両方の特定の範囲でまとめることで、これらの手法を時空間パターンの抽出に適用できる可能性もあるが、ユーザ一人ひとりのチェックインの頻度が低いため、複数のユーザで共通する軌跡やパターンなどを作成するために、データをまとめる範囲を日単位など大きくする必要があり、粗いパターンしか抽出できない。

時空間パターンは、時間と空間に関係したパターンである。様々な分野にて時空間パターンの抽出方法が研究がされている。例えば、地球科学のデータから温度の変化など環

境の変化を見つけるためにパターンを見つける研究 [35][2] や、移動の頻度の高いユーザの軌跡を抽出する研究 [27][4][14][18] などがある。

人の動きに関係した時空間パターンの抽出として、文献 [27][4] では、毎朝同じ時間に起きて仕事に向かう人の行動や、時間通りに動くバスの動きなど、日常的に起きる周期的なパターンの抽出に焦点を当て、軌跡の近さを用いて時空間パターンを抽出する方法を提案している。

時空間パターンの抽出は計算コストが高くなりがちであるが、文献 [5] では、時空間パターン抽出の計算コストを下げるために、まず地域ごとに分割して時空間パターンを抽出し、それぞれの地域の時空間パターンを組み合わせることで、地域をまたいだ時空間パターン抽出の方法を提案している。

文献 [14] では、場所間の移動ではなく、地域の移動を ROI (Region of Interest) を使って定義して、パターン抽出を行っている。POI に比べて ROI は範囲が広いため、パターン抽出がしやすくなっている。場所の間の移動に関しての問題を解決する他のアプローチとして、文献 [18] では、場所から場所への移動軌跡を格子状に分け、どの区画を通ったかの情報を加えて、パターン抽出を行っている。

これらの手法は、一般に人工データや、実際のユーザに GPS をもたせたトラッキングデータなどスパース性が低いデータに提案手法を適用しており、スパース性の高い位置情報 SNS のチェックインデータの時空間パターン抽出に適用することが難しい。チェックインとチェックインとの間にユーザがチェックイン間を直線に移動するわけではないため、チェックイン間の移動を考慮することが時空間パターン抽出では重要になってくる。チェックイン間の移動を考えることで、どこでユーザ同士が同じ位置にいたかを推定できるようになり、複数のユーザが共有する時空間パターンを抽出しやすくなると考えられる。

位置情報 SNS のチェックインを用いたパターン抽出の研究では、文献 [29][8] にて、チェックインの周期性や時間帯によるチェックインのピークの違いなど、ユーザの傾向が抽出できることが示された。文献 [29][8] では、時系列パターンとしてチェックインの頻度を見ており、空間パターンとしてチェックインの場所の分布を集計して、時系列パターンと空間パターンを分離して抽出している。

文献 [29][8] では、チェックインデータから時空間パターンの抽出を行っているが、複数の時空間パターンが混ざった状態となっており、1 日毎の周期的パターンしか取得できていない。本論文で提案する手法では、より細かい時空間パターンを抽出できる。

本論文では、類似したユーザからの時空間パターンの抽出も行う。類似したユーザを集めるためには、ユーザ間の類似度を定義する必要があるが、位置情報を用いたユーザ間の類似度も盛んに研究されている。

文献 [24] では、ユーザの移動軌跡を使ったユーザ間の類似度の計算方法を提案している。しかし、ユーザが 1 日 1 回しかチェックインしないようなチェックインデータでは、GPS などの軌跡を対象とした類似度計算の手法を適用することは難しい。

文献 [22] では、ユーザの GPS トラッキングデータから、階層的クラスタリングにより、位置をクラスタにまとめて、ユーザ間の類似度を計算している。この方法では、Foursquare のチェックインデータでは、ユーザ個々のチェックインが少ないことから類似度の計算が難しい。またクラスタの大きさによって、類似度が変わってしまい、クラスタの大きさごとにパターンを生成し比較検討する必要がある。

文献 [37][23] では、チェックインデータのカテゴリを利用して、ユーザ間の類似度を計算している。本論文で使用する Foursquare のチェックインデータは、位置情報のカテゴリ情報はユーザが指定するため誤りが多く、カテゴリ情報を使って類似度を求める方法を適用することは難しい。

位置情報を用いてユーザ間の類似度を計算するために、空間を格子状に分割して、格子の区画から区画への移動履歴からのユーザ間の類似度を求める方法は一般的であるが、格子の区画を大きくしないと類似度がほとんど 0 となってしまうという問題がある。

本論文では、拡散方程式を用いたユーザの時空間分布を用いてユーザ間の類似度を与える式を用いることで計算を可能としている。

2.2 位置予測

位置予測とは、末尾のチェックインデータの後の指定した時刻にユーザがどこにいるかを予測することである。

文献 [23] では、ユーザがその場所にいる目的のカテゴリを推定する手法として、類似した行動のユーザのカテゴリから、その場所にいる目的のカテゴリを推定している。目的のカテゴリを推定し、位置推定手法と組み合わせることで位置予測の精度を向上できる可能性はあるが、目的カテゴリをあらかじめ用意しておく必要があるため、本論文の目的では使えない。

文献 [39] では、Twitter のツイートについたタグを利用して、誰がどこでいつ何をしているか (Who, Where, When What) に関する確率モデルを構築し、ユーザ情報、ツイート内容、時刻に対して位置を予測している。この確率モデルでは、ツイートした内容を使ってユーザの位置を予測するため、ツイートした時でないと、ユーザの位置を予測することができない。

文献 [38] では、携帯の通信網のセンサーデータから、木構造と Sequential Pattern Mining を用いて頻度の高いユーザの行動を抽出し、ユーザが次にどこに行くかを予測する方法を提案している。しかしこの手法は、チェックインの間にユーザが移動していることを考慮に入れてることができないため、スパースな位置情報 SNS のデータでは、ユーザ間の類似度のほとんどが 0 となってしまう。

文献 [21] では、LDA を拡張した GLDA (Geo Latent Dirichlet Allocation) を提案している。GLDA では、時間情報を考慮していない。この問題を解決するため、文献 [15] では、GLDA を拡張し、時間を考慮した STT (Spatio-Temporal Topic) モデルを提案している。STT では、ユーザのチェックインの周期性を考慮にいれていないが、位置情報 SNS におけるユーザのチェックインには、周期性があることがあることが知られており [29][8]、周期性を用いることは位置予測において重要である。

文献 [9] では、チェックインデータを、HOME 状態でのチェックインと WORK 状態でのチェックインの 2 種類に分割し、周期性を使用した位置予測モデルである PMM と PSMM を提案している。PMM および PSMM では、HOME 状態でのチェックインと WORK 状態でのチェックインをそれぞれ二次元正規分布で表す (図 2.1)。次に、HOME 状態と WORK 状態のどちらの状態にいるかを周期的に変化させる確率モデルを導入し、それを先の二次元正規分布を組み合わせることで、ユーザが指定された時刻にどこにいるかの時空間分布

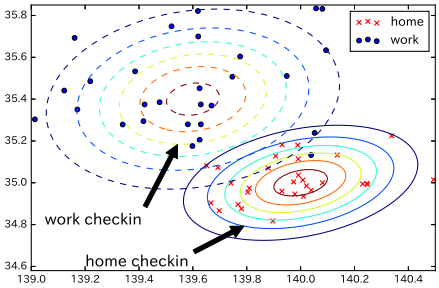


図 2.1: PMM のフィッティングの例。x 軸が経度、y 軸が緯度を表す [9]。

を与えている。PMM と PSMM は状態が HOME と WORK の 2 つに限定されているため、本論文で提案する DPMU などとくらべて精度が低い。

文献 [40] では、カーネル密度推定量として $\hat{f}(d) = \frac{1}{|D|h} \sum_{d' \in D} K(\frac{d-d'}{h})$ を提案している。ここで、$K(\cdot)$ は、$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ と定義されたカーネル関数で、D はサンプル同士の距離の集合、h は帯域幅と呼ばれるパラメータである。n を過去のチェックインの数とすると、$\hat{f}$ を用いて位置 l_j にいる確率を

$p(l_j|\text{past check-ins}) = \frac{1}{n} \sum_{i=1}^{n} \hat{f}(\text{distance from } l_j \text{ to past check-in indexed by } i)$ と定義する。これは、MGM (Multi-center Gaussian Model) [6] の一種で、時間を固定した場合の本論文で提案する予測モデルと一致する。しかし、このモデルは時間の変化を考慮に入れておらず、h (平滑化のためのパラメータ、本論文の分散に一致する) はどの時刻でも固定されている。本論文で提案するモデルは、時刻により分布が変わり、より正確に時空間分布の表現を可能としている。

第3章　全ユーザの時空間パターンの抽出

3.1　概要

位置情報SNSでは、ユーザがログを残すチェックインと呼ばれる行為を行って初めて、ユーザの位置がデータとして記録される。チェックインは、ユーザが任意のタイミングで行うため、GPSトラッカーのデータなどに比べて、スパース性がかなり高いデータとなっている。本章では、位置情報SNSのデータから、ユーザの時空間分布を推定し、推定した時空間分布から、ユーザの時空間パターンが抽出できることを示す。

本章で抽出する時空間パターンは以下のとおりである。

- 定常的なパターン: 時刻に影響されないユーザのパターン (例: ユーザが都心部で生活している)
- 周期的なパターン: 周期的にユーザが集まったりいなくなったりするパターン (例: 通勤時間帯に駅周辺の人が増える)
- 突発的なパターン: 周期性も定常性もなく、特定日に突然あらわれるパターン (例: イベントなどでユーザが急激に集まる)

ユーザの時空間分布の推定には、ユーザのチェックイン後の移動を自由粒子のように動くと仮定して、拡散方程式 [12][34] を用いる。

拡散方程式には、ユーザの平均的な移動の速度を決めるパラメータであるスケールパラメータ λ が存在する。本章では、λ の最尤推定について説明する。λ のよりよい推定方法は、第4章で導入する。

次に、推定した時空間分布から時空間パターンを抽出するため、PCAとICAを適用できるように、共分散行列を求める方法を提案する。

ユーザがどこに集まるかどこからいなくなるか、また定期的な行動パターンとなっているかを、地図上へのマッピングとパターンの周波数解析を行うことで、時空間パターンが抽出できたかどうかを調べる。

3.2 チェックインのスパース性を考慮したユーザの時空間分布推定

まずはじめに、ユーザの時空間分布を推定する。位置情報 SNS から得たチェックインデータは、特定時刻に特定位置にユーザがいたという情報であり、時間と空間に関してスパースなデータである。このチェックインデータに対して拡散方程式を適用して、ユーザのチェックインした時刻以外の時空間分布を推定する。

まず、時刻 t に位置 $\boldsymbol{x}$ にいることを $\boldsymbol{x}t$ として表す。$\boldsymbol{x}t$ が 1 つのチェックインに対応する。チェックイン列は $(\boldsymbol{x}_1t_1, \boldsymbol{x}_2t_2, \cdots, \boldsymbol{x}_nt_n)$ と表す。チェックインは地球上の任意の位置に対し起こりうるため、2 つの位置の距離を考えるときに、大圏距離を用いたり標高を考慮したりする必要があるが、本論文では、地球全体に比べて狭い範囲で推定を行うため、単純化して $\boldsymbol{x}$ を二次元平面上で近似する。

次に、あるユーザの i 番目のチェックイン $\boldsymbol{x}_it_i$ と $i+1$ 番目のチェックイン $\boldsymbol{x}_{i+1}t_{i+1}$ の間を補間するためのユーザの移動のモデルを作成する。ユーザはチェックインの後、時間が経つと他の位置に移動している可能性が高くなると考えられるので、ユーザ移動のモデルは時間が経つにつれて、空間に関するユーザの存在確率が広がっていくモデルが適切と考えられる。補間の手法としては、線形補間やガウシアン放射基底関数などが存在するが、これらの補間手法は、時間とともに存在確率が広がっていくという時間による存在確率の変化を考慮できない。そこで、本論文ではユーザが無限に広い空間を自由粒子として動くと仮定したときの拡散方程式の解を、ユーザの移動のモデルとして利用する。時刻 t に位置 $\boldsymbol{x}$ にチェックインしたユーザが、自由粒子のように広がっていくとすると、時刻 u に位置 $\boldsymbol{y}$ に存在する確率密度関数 $P_{\text{diff}}(\boldsymbol{y}|\boldsymbol{x}t, u)$ は式 (3.1) で表される。

$$P_{\text{diff}}\left(\boldsymbol{y}|\boldsymbol{x}t, u\right) = \frac{\lambda}{2\pi\left(u-t\right)} \exp\left(-\frac{\lambda\left|\boldsymbol{x}-\boldsymbol{y}\right|^2}{2\left(u-t\right)}\right) \tag{3.1}$$

$|\boldsymbol{x}-\boldsymbol{y}|^2$ は $\boldsymbol{x}$ と $\boldsymbol{y}$ のユークリッド距離の二乗である。スケールパラメータ λ は、時間によってどの程度分布が広がるかを表すパラメータである。λ はユーザの移動距離の二乗の時間変化の逆数の次元を持つ。ユーザの平均移動速度が遅いと λ が大きくなり、ユーザの平均移動速度が速いと λ が小さくなる。λ の最尤推定の方法については、3.3.3 節にて説明する。

次に、直前のチェックインが次のユーザの位置の時空間分布を決めるという、1 次マルコフの性質を持つと仮定する。これにより、$P\left(\boldsymbol{x}_{i+1}|\boldsymbol{x}t, \boldsymbol{x}_i t_i, t_{i+1}\right)$ は、$t_i \leq t < t_{i+1}$ とした時、式 (3.2) を満たす。

$$P\left(\boldsymbol{x}_{i+1}|\boldsymbol{x}t, \boldsymbol{x}_i t_i, t_{i+1}\right) = P_{\text{diff}}\left(\boldsymbol{x}_{i+1}|\boldsymbol{x}t, t_{i+1}\right) \tag{3.2}$$

式 (3.2) を変形をすると $P\left(\boldsymbol{x}|\boldsymbol{x}_i t_i, \boldsymbol{x}_{i+1} t_{i+1}, t\right)$ は式 (3.3) となる。

$$P\left(\boldsymbol{x}|\boldsymbol{x}_i t_i, \boldsymbol{x}_{i+1} t_{i+1}, t\right) = \mathcal{N}_x\left(\boldsymbol{\mu}_{\text{dist}}, \sigma^2_{\text{dist}}\right) \tag{3.3}$$

$\mathcal{N}_x\left(\boldsymbol{\mu}, \sigma^2\right)$ は二次元のガウス分布で式 (3.4) 定義される。

$$\mathcal{N}_x\left(\boldsymbol{\mu}, \sigma^2\right) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{\mu}|^2}{2\sigma^2}\right) \tag{3.4}$$

$\boldsymbol{\mu}_{\text{dist}}$ と σ^2_{dist} はそれぞれ式 (3.5) と式 (3.6) となる。

$$\boldsymbol{\mu}_{\text{dist}} = \frac{\left(t-t_i\right)\boldsymbol{x}_{i+1} + \left(t_{i+1}-t\right)\boldsymbol{x}_i}{t_{i+1}-t_i} \tag{3.5}$$

$$\sigma^2_{\text{dist}} = \frac{\left(t-t_i\right)\left(t_{i+1}-t\right)}{\lambda\left(t_{i+1}-t_i\right)} \tag{3.6}$$

データセットが N 人のユーザのチェックインデータに対して、全てのユーザに対する式 (3.3) の時空間分布の平均値を、全ユーザの時刻 t での時空間分布とする。つまり、ユーザ ω のチェックインの列を $\left(\boldsymbol{x}^\omega_1 t^\omega_1, \boldsymbol{x}^\omega_2 t^\omega_2, \cdots, \boldsymbol{x}^\omega_n t^\omega_n\right)$ とし、t が $t^\omega_i \leq t \leq t^\omega_{i+1}$ とした時に、全てのユーザに対する $P(\boldsymbol{x}|t)$ を式 (3.7) で定める。

$$P(\boldsymbol{x}|t) = \frac{1}{N}\sum_{\omega} P\left(\boldsymbol{x}|\boldsymbol{x}_i^{\omega}t_i^{\omega}, \boldsymbol{x}_{i+1}^{\omega}t_{i+1}^{\omega}, t\right) \tag{3.7}$$

3.3 時空間分布からのパターン抽出

次に全ユーザの時刻 t の時空間分布の式 (3.7) の $P(\boldsymbol{x}|t)$ から、PCA と ICA を用いたユーザの時空間パターンの抽出方法を述べる。

3.3.1 PCA を用いた時空間パターン抽出

PCA は、一般に分散を最大化しながら、行列を直交する成分に分解する手法であるが、ここでは時空間分布を、時系列パターンと空間パターンに分解する。PCA は、$P(\boldsymbol{x}|t)$ の固有関数の近似と考えることができる。無限次元の特異値分解 (SVD) では $P(\boldsymbol{x}|t)$ は式 (3.8) のように分解する。

$$P(\boldsymbol{x}|t) \approx \sum_{m=1}^{M} \nu_m U_m(\boldsymbol{x}) V_m(t) \tag{3.8}$$

M は特異値の個数、ν_m は m 番目の特異値であり、$U_m(\boldsymbol{x})$ と $V_m(t)$ は、それぞれ m 番目の空間パターンと時系列パターンである。

空間パターン $U_m(\boldsymbol{x})$ は、ユーザが空間上どこに集まるか (もしくは、いなくなるか) を表し、時系列パターン $V_m(t)$ は、その空間パターンの時間変化を表す。また、$U_m(\boldsymbol{x})$ は $\boldsymbol{x}$ にのみ依存し、$V_m(t)$ は t にのみに依存する。さらに、$m = n$ のとき $\int V_m(t) V_n(t)\,\mathrm{d}t = 1$ となり、それ以外の場合は、$\int V_m(t) V_n(t)\,\mathrm{d}t = 0$ となる。$\int U_m(t) U_n(t)\,\mathrm{d}t$ も同様に、$m = n$ のとき 1 となり、それ以外では 0 となる。これは、$V_m(t)$ と $U_m(\boldsymbol{x})$ が、正規化されており、直交条件を満たしているからである。

任意時間 t と t' の空間分布の共分散行列 $\Sigma(t,t')$ は、式 (3.9) で定義される。

$$\begin{aligned}\Sigma\left(t,t'\right) =& \int P\left(\boldsymbol{x}|t\right) P\left(\boldsymbol{x}|t'\right) \mathrm{d}\boldsymbol{x} \\ =& \frac{1}{N^2} \sum_{\omega,\omega'} \frac{1}{2\pi\sigma^2_{\mathrm{cov}}[\omega,\omega']} \exp\left(-\frac{\left|\boldsymbol{\mu}_{\mathrm{cov}}[\omega,\omega']\right|^2}{2\sigma^2_{\mathrm{cov}}[\omega,\omega']}\right)\end{aligned} \tag{3.9}$$

ここで $\boldsymbol{\mu}_{\mathrm{cov}}$ と σ^2_{cov} は式 (3.10) 与えられる。

$$\begin{aligned}\boldsymbol{\mu}_{\mathrm{cov}}[\omega,\omega'] =& \frac{\left(t-t_i^{\omega}\right)\boldsymbol{x}_{i+1}^{\omega} + \left(t_{i+1}^{\omega}-t\right)\boldsymbol{x}_i^{\omega}}{t_{i+1}^{\omega}-t_i^{\omega}} \\ & - \frac{\left(t'-t_{i'}^{\omega'}\right)\boldsymbol{x}_{i'+1}^{\omega'} + \left(t_{i'+1}^{\omega'}-t'\right)\boldsymbol{x}_{i'}^{\omega'}}{t_{i'+1}^{\omega'}-t_{i'}^{\omega'}}\end{aligned} \tag{3.10}$$

$$\sigma^2_{\mathrm{cov}}[\omega,\omega'] = \frac{(t-t_i^{\omega})(t_{i+1}^{\omega}-t)}{\lambda(t_{i+1}^{\omega}-t_i^{\omega})} + \frac{(t'-t_{i'}^{\omega'})(t_{i'+1}^{\omega'}-t')}{\lambda(t_{i'+1}^{\omega'}-t_{i'}^{\omega'})} \tag{3.11}$$

$\Sigma(t,t')$ の m 番目の固有関数は、時系列パターン $V_m(t)$ となる。時系列パターン $V_m(t)$ は、式 (3.12) を満たす関数である。

$$\nu_m^2 V_m(t) = \int \Sigma\left(t,t'\right) V_m\left(t'\right) \mathrm{d}t' \tag{3.12}$$

この時系列パターン $V_m(t)$ は解析的に解くことが出来ないため、t を離散化して、有限次元での行列分解を行う。時刻を離散化して $(t_1,\cdots,t_L)$(L は時刻の数) とし、共分散行列を $L\times L$ 行列 $\tilde{\Sigma}$ として扱う。$\tilde{\Sigma}$ の (p,q) 番目の要素は $\Sigma(t_p,t_q)$ とする。$V_m=(v_{mp})$ と ν_m^2 はそれぞれ、近似された $\tilde{\Sigma}$ の m 番目の主固有ベクトルと m 番目の固有値となる。ここで、時空間分布から PCA で取得した主成分を M 個扱うとし、M 個の固有ベクトルからなる行列を式 (3.13) で表す。

$$\boldsymbol{V} = (v_{mp}) \tag{3.13}$$

m 番目の固有ベクトルの寄与率 ρ_m は、式 (3.14) で定義する。

$$\rho_m = \nu_m^2 / \sum_{m=1}^{M} \nu_m^2 \tag{3.14}$$

次に、$U_m(\boldsymbol{x})$ は正規化されており直交条件を満たすので、$U_m(\boldsymbol{x})$ は式 (3.15) を満たす。

$$\int P(\boldsymbol{x}|t)\, V_m(t)\, \mathrm{d}t = \nu_m U_m(\boldsymbol{x}) \tag{3.15}$$

よって、$U_m(\boldsymbol{x})$ は、式 (3.16) で求められる。

$$U_m(\boldsymbol{x}) = \sum_p P(\boldsymbol{x}|t_p)\, v_{mp} \tag{3.16}$$

PCA によって抽出した時系列パターン $\boldsymbol{V}$ を、$\boldsymbol{V}_{\mathrm{PCA}}$ とする。ここで、$\boldsymbol{V}_{\mathrm{PCA}}$ と $U_m(\boldsymbol{x})$ は、$\tilde{\Sigma}$ と $P(\boldsymbol{x}|t_p)$ $(p = 1, \cdots, L)$ からのみ推定されることに注意しておく。

3.3.2 ICA を用いた時空間パターン抽出

ICA では、主固有関数を回転することにより、時空間パターンを抽出していく。ICA では、可能な限り成分が独立になるように、$\boldsymbol{V}_{\mathrm{PCA}}$ の列ベクトルを回転する。独立性が保てない場合は、複数の列ベクトルに同じパターンが含まれることとなり、同じ時空間パターンが複数の成分で観察されてしまうので、それを避けるためである。

$\boldsymbol{R} = (r_{ij})$ を $M \times M$ の直交行列の回転行列とすると、ICA における最適な $\hat{\boldsymbol{R}} = (\hat{r}_{ij})$ は、以下のように与えられる。

$$\hat{\boldsymbol{R}} = \mathrm{argmax}_{\boldsymbol{R}}\, \Phi(\boldsymbol{R}\boldsymbol{V}_{\mathrm{PCA}}) \text{ subject to } \boldsymbol{R}\boldsymbol{R}^{\mathrm{T}} = \boldsymbol{I}_M \tag{3.17}$$

Φ は、$\boldsymbol{R}\boldsymbol{V}_{\mathrm{PCA}}$ の列ベクトルの独立性の度合いを測る目的関数である。$\boldsymbol{I}_M$ は、$M \times M$ の単位行列である。様々な目的関数の ICA が存在するが、本論文では一般によく使われる、尖度を目的関数とする fastICA アルゴリズム [17] を適用する。fastICA を用いて、$\boldsymbol{R}$ は初期値として $\boldsymbol{I}_M$ を設定し、固有値が大きい成分から順番に、尖度が高くなるように回転していく。ICA によって抽出される時系列パターンを $\boldsymbol{V}_{\mathrm{ICA}} = \left(v_{mp}^{\mathrm{ICA}}\right)$ とすると、$\boldsymbol{V}_{\mathrm{ICA}}$ は、以下のように与えられる。

$$\boldsymbol{V}_{\mathrm{ICA}} = \hat{\boldsymbol{R}}\boldsymbol{V}_{\mathrm{PCA}}. \tag{3.18}$$

ここで、$\boldsymbol{V}_{\mathrm{ICA}}$ はまだ正規化されており、直交であることに注意する。次に、式 (3.16) と同様に、$U_m^{\mathrm{ICA}}(\boldsymbol{x})$ を式 (3.19) で近似する。

$$U_m^{\mathrm{ICA}}(\boldsymbol{x}) = \sum_p P(\boldsymbol{x}|t_p)\, v_{mp}^{\mathrm{ICA}} \tag{3.19}$$

U_m^{ICA} は、正規であり直交であるという条件を満たさなくなるが、U_m^{ICA} のユークリッドノルムの二乗は、m 番目のパターンの寄与率として扱うことができる。U_m^{ICA} のノルムの二乗を μ_m^2 とすると、μ_m^2 は以下のように与えられる。

$$\mu_m^2 = \sum_{n=1}^{M} \hat{r}_{mn}^2 \nu_n^2 \tag{3.20}$$

m 番目のパターンの寄与率 ρ_m^{ICA} は、$\sum_{m=1}^{L} \nu_m^2$ が回転により不変であることより、以下のように計算できる。

$$\rho_m^{\mathrm{ICA}} = \mu_m^2 / \sum_{m=1}^{L} \nu_m^2 \tag{3.21}$$

上記が、ICA により時系列パターン (式 (3.18)) と、空間パターン (式 (3.19)) と、寄与率 ρ_m^{ICA}(式 (3.21)) を抽出する方法となる。

3.3.3 スケールパラメータ λ の最尤推定

この章の実験では、3.2 節で導入したスケールパラメータ λ の推定を最尤推定を用いて行う。事前分布を用いた λ の推定方法とは第 4 章で比較する。

ユーザ ω のチェックイン列 $(\boldsymbol{x}_1^\omega t_1^\omega, \boldsymbol{x}_2^\omega t_2^\omega, \cdots, \boldsymbol{x}_n^\omega t_n^\omega)$ において、$\boldsymbol{x}_i^\omega t_i^\omega$ は $\boldsymbol{x}_{i-1}^\omega t_{i-1}^\omega$ と $\boldsymbol{x}_{i+1}^\omega t_{i+1}^\omega$ の間のチェックインとなる。$\boldsymbol{x}_{i-1}^\omega t_{i-1}^\omega$ と $\boldsymbol{x}_{i+1}^\omega t_{i+1}^\omega$ から推定される t_i^ω の時の $\boldsymbol{x}_i^\omega$ の確率 $P\left(\boldsymbol{x}_i^\omega | \boldsymbol{x}_{i-1}^\omega t_{i-1}^\omega, \boldsymbol{x}_{i+1}^\omega t_{i+1}^\omega, t_i^\omega\right)$ を用いると対数尤度関数は以下のようになる。

$$
\begin{aligned}
L(\lambda) &= \sum_{i,\omega} \log P\left(\boldsymbol{x}_i^\omega | \boldsymbol{x}_{i-1}^\omega t_{i-1}^\omega, \boldsymbol{x}_{i+1}^\omega t_{i+1}^\omega, t_i^\omega\right) \\
&= \sum_{i,\omega} \left(\log \lambda + \log C_i^\omega - \lambda q_i^\omega\right)
\end{aligned} \tag{3.22}
$$

ここで、q_i^ω は以下で与えられる。

$$
q_i^\omega = \frac{\left(\boldsymbol{x}_i^\omega - \frac{\left(t_i^\omega - t_{i-1}^\omega\right)\boldsymbol{x}_{i+1}^\omega + \left(t_{i+1}^\omega - t_i^\omega\right)\boldsymbol{x}_{i-1}^\omega}{t_{i+1}^\omega - t_{i-1}^\omega}\right)^2}{2\frac{\left(t_i^\omega - t_{i-1}^\omega\right)\left(t_{i+1}^\omega - t_i^\omega\right)}{t_{i+1}^\omega - t_{i-1}^\omega}} \tag{3.23}
$$

C_i^ω は λ と独立である。

式 (3.22) の 1 次導関数が 0 になる値を解くと、最適な λ は式 (3.24) で与えられる。

$$
\lambda = \frac{\sum_\omega n_\omega}{\sum_{i,\omega} q_i^\omega} \tag{3.24}
$$

n_ω は ω に対する、三つ組の数である。

3.4 データ

この章の実験で使用する Foursquare のチェックインデータは以下のようにして集めた。Foursquare は独自の API[1] を提供しており、これを利用することによりチェックインデータを取得できる。しかし、この API はアクセス回数の制限があり、また取得できるデータの範囲が、API を使っているユーザの友人までに限られており、今回の実験を行うには適切ではなかった。そこで本論文では、文献 [29] と同様にチェックインデータを Twitter から取得した。Foursquare は、ユーザがチェックインするときに、Twitter を用いてチェックインデータを共有することができる機能を備えている。共有されたチェックインのツイートには、チェックインデータの詳細を見ることができる Foursquare への URL が付加され

[1] `https://developer.foursquare.com/`

ている。ユーザの Twitter の Feed が public に設定してあると、このチェックインの詳細情報は誰でもアクセスすることができる。詳細情報には、チェックインした場所の名称や緯度経度、また場所に付加されたタグ情報が含まれている。チェックインデータの収集では、Twitter のストリーミング API を使って、'4sq' というキーワードで検索し、2012 年 2 月 28 日から 2012 年 3 月 31 日までのおよそ 1ヶ月の 1,000 万件のツイートデータを集めた。Twitter で共有されたチェックインデータの中で、Twitter を日本語設定にしているユーザによるチェックインデータで、東京都周辺のチェックインデータのみを取得した。さらに、データ収集の対象の期間中に少なくとも 50 回のチェックインしているユーザのみを選択し、7,710 人の 843,194 件のチェックインデータを使用した。このチェックインデータを 3.6 節の実験に用いた。

3.5 実装

チェックインデータからのユーザの時空間分布の推定は、C++のプログラムにて実装した。

式 (3.3) は $t = t_i$ または $t = t_{i+1}$ にて、$\mathcal{N}_x\left(\boldsymbol{\mu}_{\mathrm{dist}}, \sigma^2_{\mathrm{dist}}\right)$ が $\sigma^2_{\mathrm{dist}} = 0$ となり発散するので、発散を避けるために、$\mathcal{N}_x(m, v)$ の v に、正で微小な値の下限を設ける必要がある。本章の実験では、λv の下限を 8.64 秒 (10^{-4} 日) とした。これにより、式 (3.3) は発散しなくなった。以降この下限を λ_0 と表す。

式 (3.24) において、連続したチェックインで物理的に不可能な速度で移動していることに対応する外れ値の λ の推定への影響を和らげるために、式 (3.24) の計算では平均より 10 倍より大きい q_i^{ω} を除いた。チェックインの時間間隔や移動した距離を計算するデータの前処理と ICA によるパターン抽出は、それぞれ Ruby と MATLAB によって実装した。fastICA の MATLAB 用のパッケージとして、fast ica version2.5[2]を使用した。抽出した時空間パターンは、Javascript を用いてブラウザーベースのインターフェースで表示するプログラムを実装した。図 3.1 は実装したインターフェースのスクリーンショットである。時間パターンと時間の周波数領域 (左上と上の中央)、時間パターンのピークの時点

[2] http://research.ics.aalto.fi/ica/fastica/

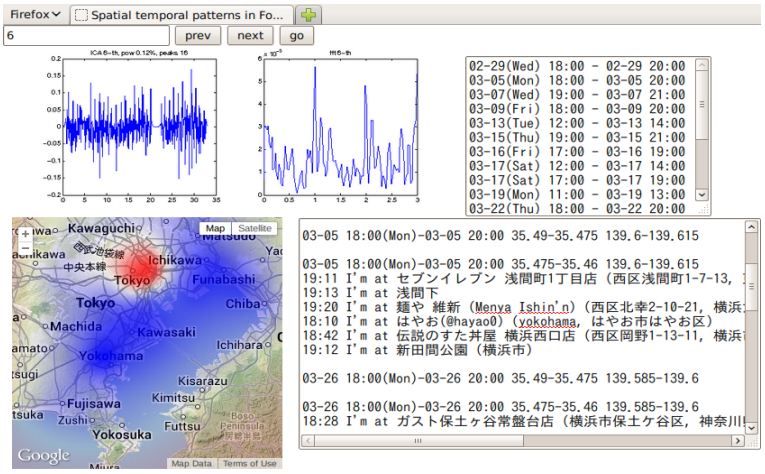


図 3.1: 時空間パターンを表示するシステムのインターフェイス

(右上)、マップ上の空間パターン (左下) 及び、時空間パターンのピーク周辺でのツイート (右下) を表示している。マップ上の空間パターンは Google Maps API を用いて地図上に重ねて表示した。

3.6 Foursquare からのパターン抽出

全ユーザのチェックインデータから時空間パターンを抽出するため、全ユーザのチェックインデータから推定した時空間分布の共分散行列 (式 (3.9)) が必要となる。ここでは、空間分布と式 (3.11) による共分散行列を推定するために、3.4 節にて取得した、7,710 人のユーザのチェックインデータを利用した。この共分散行列に ICA を適用し、8 つの独立した成分を抽出した。8 つの成分の総寄与率は 99.87%である。抽出した 8 つの空間パターン (式 (3.19)) を図 3.2 に示す。赤が平均より混んでいる状態を表し、青が平均より空いている状態を表している。抽出した空間パターンのうち幾つかはマップの広域に広がるパターン (SP1, SP6, SP8) であり、また他のいくつかは狭いエリアに集中したパターンとなった (SP3, SP4, SP7)。狭い範囲に集中したパターンは、全て東京の中心付近に集まっている。図 3.3 は、SP1 から SP8 の空間パターンと周波数領域を示しており、ユーザの行動パターンに周期性があるかどうかを表している。

以下では、抽出した各パターンについて個別に検討する。SP1(図 3.2-(a), 図 3.3-(a)) は、都市部全体に広がる空間パターンであり、安定した時間パターンである。よって、SP1 は東京の中のユーザの定常的なパターンといえる。図 3.2-(b), 3.2-(d), 3.2-(e), 3.2-(f) は、SP2, SP4, SP5, SP6 が一日もしくは半日の周期的なパターンであることを表している。これらのパターンは、東京の中心に強く関係した空間パターンを表しており、毎日もしくは毎半日の中心部の人口の分散と集中を表していると考えられる。

SP3 と SP7 では、長い周期が観測された (図 3.3-(c), 3.3-(g))。この長い周期をもう少し詳しく調査してみるために、SP3 と SP7 の時間パターンのピークを図 3.4 に示す。SP3 は、2 月 28 日, 3 月 20 日, 4 月 1 日の 3 つのピークを持っている。2 月 28 日と 4 月 1 日は、それぞれ集めたデータの始まりと終わりの次の日に一致する。3 月 20 日に関しても 2 月 28 日と 4 月 1 日と同じように、システムトラブルなどの影響によりデータが取得できていない部分が観測されていた。これらより、SP3 はチェックインのデータが完全に取得できなかった周期を抽出している、突発的なパターンと考えられる。SP7 の全てのピークは週末 (土曜、日曜) となっている。さらに、SP の空間パターン 図 3.2-(g) は中心部が青くなっている (通常より人が減っていることを意味する)。これは、ユーザが中心部 (多くのビジネス地域) を週末に避けていることを意味する。最後に、SP8 の寄与率はかなり低くなっている (0.02%) ことより、SP8 はユーザの行動に大きく関連しないと考えられるので、ノイズと考えることができる。

抽出したパターンについてまとめる。SP1 はユーザの定常で広範囲に広がる定常的なパターンを表している。また SP2, SP4, SP5, SP6 は中心部の短期の周期的なパターンを表している。SP3 はチェックインデータが取れなかった突発的なパターンを表している。SP7 は週末の行動と一致する周期的なパターンを表している。これらはもっともらしく、直感的にもつじつまが合うと考えられる。よって、提案手法はスパースな Foursquare のチェックインデータから、ユーザの時空間分布を推定し、推定した分布からユーザの時空間パターンを抽出できているといえる。

3.7 推定におけるサンプリングの影響

ユーザのサンプルサイズが限られている場合に、拡散方程式による推定の精度を測るために、サンプルサイズを変えて (10, 20, 50, 100, 200, 500, 1,000 人のユーザ数)、全てのユーザ (7,710 人) のチェックインデータを使った場合とサンプルしたユーザのチェックインデータを使った場合との比較を行う。どのサンプルサイズでも 10 回独立にランダムサンプリングを行い時空間分布の推定を行った。ここでは全てのサンプルで作った時空間分布を真であるとする。サンプリング中で推定された分布と全てのサンプルを用いて推定した時空間分布の誤差を考える。ここで、推定されたユーザの分布は連続な分布であるが、この連続空間を離散化し、東京付近の 100×100 の確率行列として扱う。1 時間毎にサンプルした時点は 792 個からなるので、推定された時空間分布は $100 \times 100 \times 792$ のテンソルとなる。テンソル同士の距離をフロベニウスノルム (全ての要素のユークリッド距離) で計算し、さらにこれを全てのユーザで作ったテンソルのフロベニウスノルム (全ての要素の二乗和の平方根) で割り、相対誤差とし以下のように定義する。

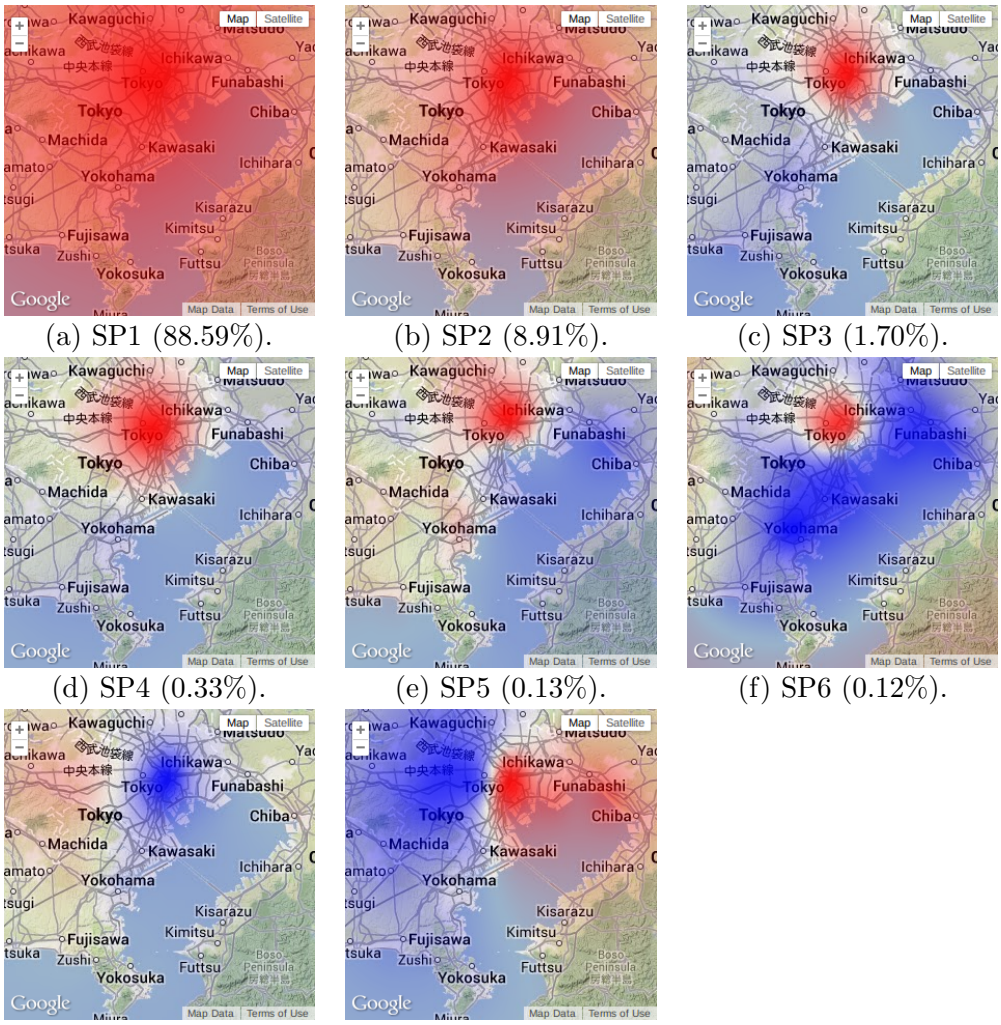
$$\frac{\|\hat{\boldsymbol{A}} - \boldsymbol{A}\|_F}{\|\boldsymbol{A}\|_F} \tag{3.25}$$

$\hat{\boldsymbol{A}}$ と $\boldsymbol{A}$ は、それぞれサンプリングして推定した分布と全てのサンプルで推定した分布を表す。時空間分布に加え、式 (3.9) により推定された共分散を、全てのサンプルを用いた真の共分散と比較する。この共分散の誤差もまたフロベニウスノルムの相対誤差で求める。提案手法で使用した ICA は, 推定した共分散が同じであれば同じ結果を出すように目的関数が設定されている。

図 3.5 は推定した分布と共分散の誤差の減衰曲線を表している。図 3.5 より、10 回の試行の平均誤差はサンプルサイズが大きくなるに連れて減少していると言える。サンプルサイズが 1,000 人まで大きくなると、1.0 付近の誤差となっていることがわかる。ただし、これは全てのユーザを利用した $\|\boldsymbol{A}\|_F$ に対する誤差の比率であるので、この 1.0 は小さくない。提案手法は、サンプルサイズを大きくすることによって誤差は小さくなるが、サンプルサイズに対して誤差が単調に減少し、精度が高くなっているといえる。

サンプルサイズを大きくすると、全てのユーザのチェックインデータから推定した時空間分布から抽出した時空間パターンと誤差の少ない時空間パターンを抽出できる一方で、1 人のユーザに特徴的な時空間パターンの抽出をすることができなくなり、ユーザの位置の推定の精度が下がると考えられる。これはサンプリングがランダムに行われ、様々なユーザの特徴が混ざってしまうことが原因と考えられるので、ランダムサンプリングによってユーザを集めるかわりに、挙動が類似したユーザのチェックインデータを集めることで、ユーザに特徴的な時空間パターンを含んだ時空間分布を推定できる可能性がある。第 5 章では、類似したユーザを集めることで、ユーザに特徴的な時空間パターンを抽出できることを示す。

また、推定の精度には、λ が影響する。そこで、より頑健である事前分布を用いた λ の推定方法を第 4 章にて提案する。

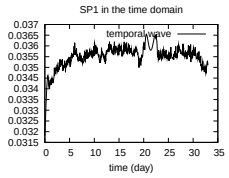


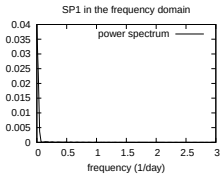
(a) SP1 (88.59%). (b) SP2 (8.91%). (c) SP3 (1.70%).

(d) SP4 (0.33%). (e) SP5 (0.13%). (f) SP6 (0.12%).

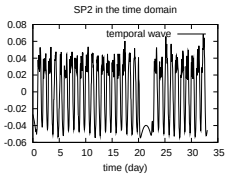
(g) SP7 (0.07%). (h) SP8 (0.02%).

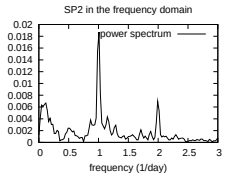
図 3.2: 抽出した空間パターン SP1-SP8 とその寄与率。マップは東京周辺に限定している。



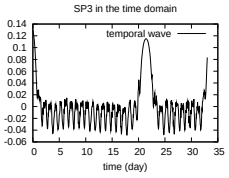


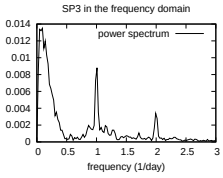
(a) for SP1.



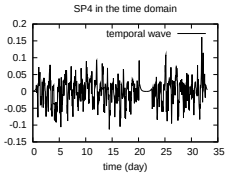


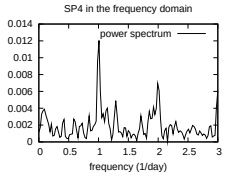
(b) for SP2.



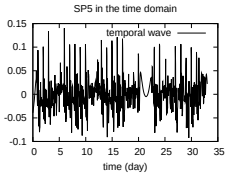


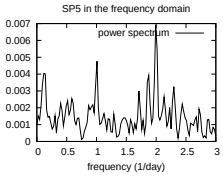
(c) for SP3.



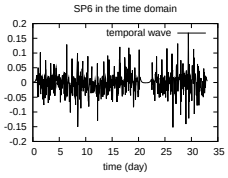


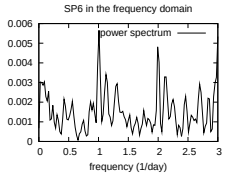
(d) for SP4.



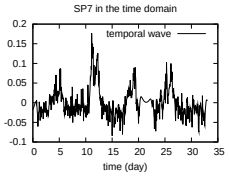


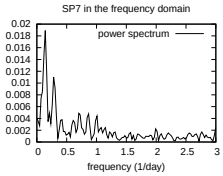
(e) for SP5.



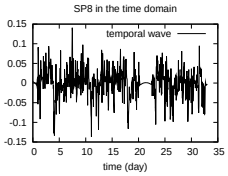


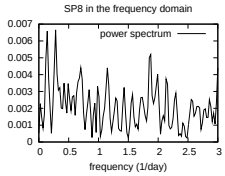
(f) for SP6.





(g) for SP7.





(h) for SP8.

図 3.3: SP1-SP8 の時間と周波数領域における時系列パターン: 左が時間を表し、右が周波数領域を表す。

```
02-28(Tue) 08:00 - 02-28 23:00
03-19(Mon) 20:00 - 03-21 18:00
04-01(Sun) 06:00 - 04-01 10:00
```

(a) for SP3.

```
03-10(Sat) 09:00 - 03-10 11:00
03-10(Sat) 11:00 - 03-10 20:00
03-10(Sat) 21:00 - 03-10 23:00
03-11(Sun) 06:00 - 03-11 08:00
03-11(Sun) 09:00 - 03-11 18:00
03-18(Sun) 12:00 - 03-18 14:00
03-18(Sun) 17:00 - 03-18 19:00
03-24(Sat) 15:00 - 03-24 17:00
03-25(Sun) 11:00 - 03-25 13:00
```

(b) for SP7.

図 3.4: SP3(左) と SP(7) のピークの時点のリスト: 各行がピークの始まり (月, 日, 週, 時間) と終わりを表している。

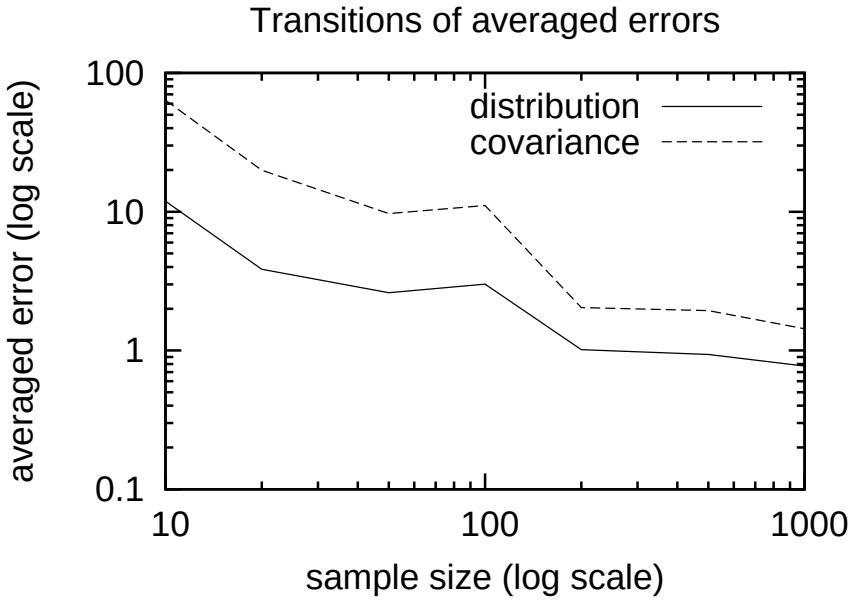


図 3.5: x 軸: サンプルサイズ (対数軸)。y 軸: 推定した時空間分布の誤差 (対数軸)。実線は推定した分布の平均誤差。破線は推定した共分散の平均誤差。10 回のランダムサンプリングの平均値。

第4章　スケールパラメータ λ の推定

4.1　概要

第 3 章で説明したように、本論文ではユーザを自由粒子として扱い、拡散方程式に基づきユーザの時空間分布を推定している。この拡散方程式には、ユーザの移動距離の二乗の時間変化の逆数の次元を持つスケールパラメータ λ がある。スケールパラメータ λ が小さすぎると、ユーザが瞬間的に遠くに移動するモデルとなり、逆にスケールパラメータ λ が大きすぎると、ユーザが全然動かないというモデルになってしまう。従って、使用するサンプル内のユーザの時空間分布の推定に適した拡散方程式にするために、スケールパラメータ λ の推定はあまり不正確でない必要がある。そこで、本章では事前分布を用いた λ の推定方法を導入し、第 3 章で用いた最尤推定と比較する。

4.2　極座標を用いたユーザの時空間分布

λ の推定の式変形を簡単にするため、連続したチェックインのペアを極座標で表現することで確率をユークリッド距離と時間間隔の式で表す。

まず、チェックインの列を $(\boldsymbol{x}_1 t_1, \boldsymbol{x}_2 t_2, \cdots, \boldsymbol{x}_n t_n)$ と表す。式 (3.1) で定義した P_{diff} は、$\boldsymbol{x}t$ のチェックインしたユーザが、時刻 u に位置 $\boldsymbol{y}$ に存在する確率であることより、$\boldsymbol{x}_{i+1} t_{i+1}$ と $\boldsymbol{x}_i t_i$ の間の P_{diff} は、$\bar{\boldsymbol{x}}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i$ と $\bar{t}_i = t_{i+1} - t_i$ にのみ依存しているため、式 (4.1) に書きなおすことができる。

$$P_{\text{diff}}\left(\bar{\boldsymbol{x}} | \bar{t}, \lambda\right) = \frac{\lambda}{2\pi\bar{t}} \exp\left(-\frac{\lambda\left|\bar{\boldsymbol{x}}\right|^2}{2\bar{t}}\right). \tag{4.1}$$

次に、ユーザの $\bar{t}$ 秒後の存在確率を表す式 (4.1) を極座標を用いて表す。まず、$\bar{\boldsymbol{x}}$ を極

座標 $\left(\sqrt{D}\cos\theta, \sqrt{D}\sin\theta\right)$ で表す。D はユークリッドにおける $\bar{\boldsymbol{x}}$ のノルムの二乗 (すなわち $\boldsymbol{x}_{i+1}$ と $\boldsymbol{x}_i$ の間の二乗距離) であり、θ は、緯度と経度を軸とした場合の移動後の偏角 $(0 \leq \theta < 2\pi)$ である。D と θ の分布は式 (4.2) で与えられる。

$$\begin{aligned} & P\left(D, \theta | \bar{t}, \lambda\right) \\ = \quad & P_{\text{diff}}\left(\sqrt{D}\cos\theta, \sqrt{D}\sin\theta | \bar{t}, \lambda\right) |\boldsymbol{J}| \\ = \quad & \frac{\lambda}{2\pi\bar{t}} \exp\left(-\frac{\lambda D}{2\bar{t}}\right) |\boldsymbol{J}| \end{aligned} \tag{4.2}$$

$|\boldsymbol{J}|$ は座標変換の Jacobian 行列の行列式であり、式 (4.3) で与えられる。

$$|\boldsymbol{J}| = \left| \begin{pmatrix} \frac{\partial\sqrt{D}\cos\theta}{\partial D} & \frac{\partial\sqrt{D}\cos\theta}{\partial\theta} \\ \frac{\partial\sqrt{D}\sin\theta}{\partial D} & \frac{\partial\sqrt{D}\sin\theta}{\partial\theta} \end{pmatrix} \right| = \frac{1}{2} \tag{4.3}$$

$P\left(D, \theta | \bar{t}, \lambda\right)$ は θ に依存しない。ここで、θ に関して、$P\left(D, \theta | \bar{t}, \lambda\right)$ を積分すると、$P\left(D | \bar{t}, \lambda\right)$ は式 (4.4) で与えられる。

$$P\left(D | \bar{t}, \lambda\right) = \int P\left(D, \theta | \bar{t}, \lambda\right) \mathrm{d}\theta = \frac{\lambda}{2\bar{t}} \exp\left(-\frac{\lambda D}{2\bar{t}}\right) \tag{4.4}$$

これにより、2 つの連続したチェックインのペア $(\boldsymbol{x}_i t_i, \boldsymbol{x}_{i+1} t_{i+1})$ を、ユークリッド距離と時間間隔の $(D, \bar{t})$ で表すことができた。チェックイン列のチェックインのペアのインデックスを k として、チェックインのペアを $(D_k, \bar{t}_k)$ で表す。

4.3 最尤推定

3.3.3 節で説明した最尤推定による λ を求める式を $(D_k, \bar{t}_k)$ を使って表す。$(D_k, \bar{t}_k)$ に対して、$\sum_k \log P\left(D_k, \bar{t}_k | \lambda\right)$ を最大化する λ の最尤推定は、$P\left(D, \bar{t} | \lambda\right) \left(= P\left(D | \bar{t}, \lambda\right) P\left(\bar{t}\right)\right)$ を最大化することで得られる。もし $P\left(\bar{t}\right)$ が λ に依存しない場合は、最尤推定 (MLE) による λ の推定値 $\hat{\lambda}_{\text{MLE}}$ は、K を $(D_k, \bar{t}_k)$ の数として、式 (4.5) で求められる。

$$\hat{\lambda}_{\mathrm{MLE}} = \mathrm{argmax}_{\lambda} \sum_{k} \log P\left(D_k, \bar{t}_k | \lambda\right)$$
$$= \mathrm{argmax}_{\lambda} \sum_{k} \log P\left(D_k | \bar{t}_k, \lambda\right) = \frac{K}{\sum_k \left(D_k / 2\bar{t}_k\right)} \tag{4.5}$$

式 (4.5) は、$D_k/2\bar{t}_k$ は $\bar{t}_k$ が 0 に近づくと発散する。

4.4 事前分布を用いた推定

$\bar{t}_k$ が 0 に近づいた場合に、$D_k/2\bar{t}_k$ が発散する影響を抑えるために、事前分布を用いた推定方法を導入する。この方法ではまずは、事前分布 $P\left(\bar{t}\right)$ を仮定し、周辺分布 $P\left(D|\lambda\right) = \int_0^\infty P\left(D|\bar{t}, \lambda\right) P\left(\bar{t}\right) \mathrm{d}\bar{t}$ を計算し、λ の推定を、$P\left(D|\lambda\right)$ に基いて行う。

適切な事前分布を選ぶために実際のチェックインデータを調べたところ、時間間隔 $\bar{t}$ は以下の様な性質があることがわかった。

1. $\bar{t}$ は常に正である ($\bar{t} > 0$)。言い換えると時間間隔は常に正であり、同時刻に複数のチェックインがあることはない。

2. 図 4.1 の $\bar{t}$ の分布を見ると、0 の近くに $\bar{t}$ の分布のピークが存在する。ほとんどのチェックインは前のチェックインのすぐ後に行われることを意味する。

3. $\bar{t}$ の分布は長いロングテールとなっている。言い換えると、いくつかのチェックインは前のチェックインから、かなり時間をあけてチェックインされる。

これらより、$P(\bar{t})$ の事前分布として、逆ガンマ分布を適用することにした。逆ガンマ分布は上に上げた 3 つの性質をすべて満たしており、周辺分布 $P\left(D|\lambda\right)$ を解析的に計算できるからである。$\tau = 1/\bar{t}$ とすると、τ の分布は式 (4.6) のガンマ分布となる。

$$P\left(\tau|\alpha, \beta\right) = \frac{\beta^\alpha}{\Gamma\left(\alpha\right)} \tau^{\alpha-1} \exp\left(-\beta\tau\right) \tag{4.6}$$

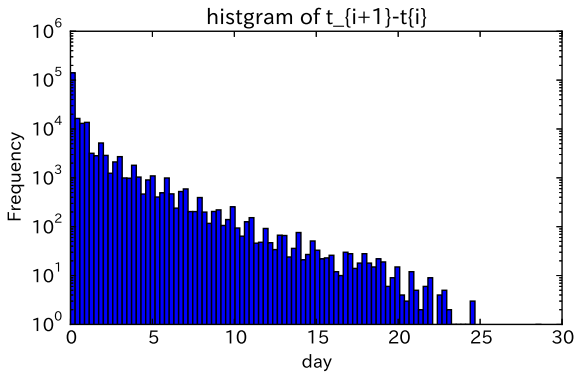


図 4.1: チェックインデータ中の 2 つの連続したチェックインの間の時間 $\bar{t}$ の頻度 (対数軸)。

$\Gamma(\alpha)$ はガンマ関数で、式 (4.7) で与えられる。

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} \exp(-x)\, \mathrm{d}x \tag{4.7}$$

(α, β) を最尤推定する。$\tau_k = 1/\bar{t}_k$ とする。

$$(\alpha, \beta) = \mathrm{argmax}_{(\alpha,\beta)} \sum_k \log P(\tau_k|\alpha, \beta) \tag{4.8}$$

式 (4.8) は、解析的には解くことができないが、MATLAB の関数 “gamfit” を使って、数値計算で最適な $\hat{\alpha}$ と $\hat{\beta}$ を計算できる。

$\hat{\alpha}$ と $\hat{\beta}$ を用いると、周辺分布 $P(D|\lambda)$ は以下のようになる。$\hat{\alpha}$ と $\hat{\beta}$ は正の値となる。

$$\begin{aligned}
&P(D|\lambda) = \int_0^\infty P(D|\tau, \lambda)\, P\left(\tau|\hat{\alpha}, \hat{\beta}\right) \mathrm{d}\tau \\
=&\int_0^\infty \frac{\lambda\tau}{2} \exp\left(-\frac{\lambda\tau D}{2}\right) \frac{\hat{\beta}^{\hat{\alpha}}}{\Gamma(\hat{\alpha})} \tau^{\hat{\alpha}-1} \exp\left(-\hat{\beta}\tau\right) \mathrm{d}\tau \\
=&\frac{\lambda\hat{\beta}^{\hat{\alpha}}}{2\Gamma(\hat{\alpha})} \int_0^\infty \tau^{\hat{\alpha}} \exp\left(-\left(\frac{\lambda D}{2} + \hat{\beta}\right)\tau\right) \mathrm{d}\tau \\
=&\frac{\lambda\hat{\beta}^{\hat{\alpha}}\Gamma(\hat{\alpha}+1)}{2\Gamma(\hat{\alpha})} \left(\frac{\lambda D}{2} + \hat{\beta}\right)^{-(\hat{\alpha}+1)} \\
=&\frac{\lambda\hat{\alpha}}{2\hat{\beta}} \left(\frac{\lambda D}{2\hat{\beta}} + 1\right)^{-(\hat{\alpha}+1)}
\end{aligned} \tag{4.9}$$

式 (4.9) の確率密度関数は、Lomax 分布 [25] として知られている。よって、逆ガンマ分布を事前分布に使った λ の推定値 $\hat{\lambda}_{\text{prior}}$ は、式 (4.10) となる。

$$\hat{\lambda}_{\text{prior}} = \text{argmax}_{\lambda} \sum_{k} \log P\left(D_k|\lambda\right) \tag{4.10}$$

$\sum_k \log P\left(D_k|\lambda\right)$ を最大とする λ を求めるために、λ に関して、式 (4.10) を微分すると式 (4.11) となる。

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\lambda}\sum_{k}\log P\left(D_k|\lambda\right) &= \sum_{k}\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\log\lambda - (\hat{\alpha}+1)\log\left(\frac{\lambda D_k}{2\hat{\beta}}+1\right)+\log\frac{\hat{\alpha}}{2\hat{\beta}}\right)\\ &= \sum_{k}\left(\frac{1}{\lambda}-(\hat{\alpha}+1)\frac{D_k}{\lambda D_k+2\hat{\beta}}\right)\end{aligned} \tag{4.11}$$

式 (4.11) が 0 となるのは、$\hat{\lambda}_{\text{prior}}$ が式 (4.12) を満たすときとなる。

$$\frac{1}{K}\sum_{k}\frac{D_k}{D_k+2\hat{\beta}/\hat{\lambda}_{\text{prior}}} = \frac{1}{\hat{\alpha}+1} \tag{4.12}$$

式 (4.12) の右側は、$\hat{\alpha}$ が正であることより、式 (4.13) を満たす。

$$0 < \frac{1}{\hat{\alpha}+1} < 1 \tag{4.13}$$

また、$\hat{\beta}$ と D_k が正であるより、式 (4.12) の左側は、$\hat{\lambda}_{\text{prior}}$ に関して単調増加であるので、式 (4.12) を満たす唯一解 $\hat{\lambda}_{\text{prior}}$ が存在する。これは MATLAB の関数 “fzero” にて求めることが可能である。

4.5 データ

第 3 章で、使用していたチェックインデータはシステムの不都合によりデータが取得できない期間があり、一定期間欠損しているデータを含んでいた。この章では、データが取得できなかった期間の影響をできるだけ抑えるために、3.4 節と同様の手法を用いて、新

たに Twitter streaming 経由で東京周辺の Foursquare のチェックインデータを、2013 年 12 月 26 日から 2014 年 1 月 24 日までのおよそ 1ヶ月の期間で集めた。その結果、期間中に最低でも 50 回以上チェックインしているユーザは 5,862 人で、そのチェックインの総数は 612,828 回であった。本章でも、3.5 節と同様に、$\bar{t}_k = 0$ となる場合を避けるため、$\bar{t}_k$ に下限として、極小の正の値 λ_0 を設定した。この下限を用いて、$\bar{t}_k$ が下限である λ_0 を下回るとき、$\bar{t}_k = \lambda_0$ と置き換えた。

4.6 実験結果

本章では λ の推定値 $\hat{\lambda}_{\mathrm{MLE}}$ (4.3 節) と 4.5 節のチェックインデータの 606, 966 回のチェックインのペアから、$\hat{\lambda}_{\mathrm{MLE}}$ と $\hat{\lambda}_{\mathrm{prior}}$ はそれぞれ 0.0184 と 460 と推定された。$\hat{\lambda}_{\mathrm{MLE}}$ が大きな平均速度となったのは、$D_k/2\bar{t}_k$ は $\bar{t}_k$ が 0 に近づいた場合に発散し、推定の安定性が失われてしまったことが原因だと考えられる。推定の安定性を調べるために、サンプリングしたチェックインデータからの $\hat{\lambda}_{\mathrm{MLE}}$ と $\hat{\lambda}_{\mathrm{prior}}$ の推定を、サンプリングして 100 回行った。サンプリングでは 10000 個サンプルした。その結果、$\hat{\lambda}_{\mathrm{MLE}}$ は平均 0.0448 (標準偏差 0.0465) となり、$\hat{\lambda}_{\mathrm{prior}}$ は平均 454 (標準偏差 25.7) となった。$\hat{\lambda}_{\mathrm{MLE}}$ の平均値は全てのサンプルで計算した $\hat{\lambda}_{\mathrm{MLE}} = 0.0184$ とかなり異なっている。$\hat{\lambda}_{\mathrm{MLE}}$ の標準偏差 0.0465 は、平均値 0.0448 と同程度で大きい。これに対して、$\hat{\lambda}_{\mathrm{prior}}$ の平均値は全てのサンプルで計算した最適な $\hat{\lambda}_{\mathrm{prior}} = 454$ と近い値となっている。さらに $\hat{\lambda}_{\mathrm{prior}} = 454$ のとき、標準偏差は 25.7 で、小さい。このように事前分布を用いた λ の推定は安定している。

次に様々なスケールパラメータに対する $P(\boldsymbol{x}|t)$ の対数尤度を 5 分割交差検定にて調べた (図 4.2)。5 分割交差検定では、ユーザーをランダムに 5 つに分割し、5 つのうち 4 つのユーザのチェックインデータをトレーニングデータ、残りの 1 つのユーザのチェックインデータをテストデータとして対数尤度を計算し、テストデータを 5 回入れ替えた対数尤度の平均値を計算する。図 4.2 より、$\hat{\lambda}_{\mathrm{prior}} = 454$ は対数尤度の最大値に近いため適切な値であるといえる。一方で、$\hat{\lambda}_{\mathrm{MLE}} = 0.0184$ は相対的に小さい対数尤度となって適切でないといえる。交差検定により尤度が高い λ を求めることは可能だが、最適な λ を探すために、様々な λ に交差検定をかける必要があり、計算コストが非常に高いので、事前分布を

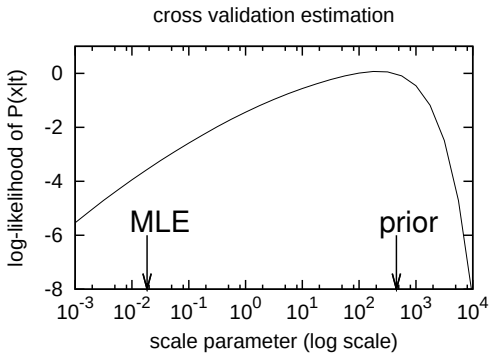


図 4.2: スケールパラメータ λ と 5 分割交差検定結果。λ を 10^{-3} から 10^4 の範囲で動かした時の $P(\boldsymbol{x}|t)$ の対数尤度の 5 分割交差検定の結果を表している。矢印は $\hat{\lambda}_{\mathrm{MLE}} = 0.0184$ と $\hat{\lambda}_{\mathrm{prior}} = 454$ を表している。

用いたモデルは実用的だと考えられる。

第5章　類似ユーザの時空間パターンの抽出

5.1　概要

第3章では、全ユーザのチェックインデータから推定した時空間分布から時空間パターンを抽出した。この全ユーザでの時空間パターンの解析は、全ユーザの一般的な動きを抽出するのには向いているが、時空間分布が大きく異なるユーザをまとめて時空間パターンを抽出するので、1人のユーザに特徴的な時空間パターンは抽出できない。しかし、1人のユーザのチェックインデータだけでは、チェックインデータのスパース性のため有効な時空間パターンを抽出できない。そこで本章では、時空間分布が類似しているユーザのチェックインデータを集めて、そこから推定した時空間分布から時空間パターンを抽出することにより、より1人のユーザの特徴を反映した時空間パターンを抽出する。時空間分布の類似性はヘリンジャー距離を用いて定義する。ヘリンジャー距離によりWard法でユーザをクラスタリングをし、時空間分布の類似しているユーザを集める。クラスタ毎に、そのクラスタに属すユーザのチェックインデータから時空間パターンを抽出する。クラスタ毎に抽出された時空間パターンがユーザの特性を反映している時空間パターンとなっているかどうかを、第3章と同様に、実際の地図上へのマッピングと周波数の解析により示す。

5.2　ユーザ間の距離

本章では階層的クラスタリングを適用するために、ユーザの時空間分布間の距離を定義する。文献 [26] で時空間クラスタリングの手法が提案されているが、本論文では、スパースな位置情報SNSのデータを扱うため適用できないため、第3章で定義したユーザの時

空間分布を用いて距離を定義する。時空間分布は確率分布であるので、確率分布間のユークリッド距離として定義できるヘリンジャー距離 [31] を、ユーザ間の距離として用いる。

まず、t が $[t_i^\omega, t_{i+1}^\omega]$ に含まれる i とし、$P\left(\boldsymbol{x}|t,\omega\right)$ が式 (5.1) で定義されるとする。

$$P\left(\boldsymbol{x}|t,\omega\right) = P\left(\boldsymbol{x}|\boldsymbol{x}_i^\omega t_i^\omega, \boldsymbol{x}_{i+1}^\omega t_{i+1}^\omega, t\right) \tag{5.1}$$

この $P\left(\boldsymbol{x}|t,\omega\right)$ を用いて、時刻 t におけるユーザ ω とユーザ ω' のヘリンジャー距離 $H(\omega,\omega',t)$ は、式 (5.2) で定義される。

$$\begin{aligned} H\left(\omega,\omega',t\right) &= \sqrt{\frac{1}{2}\int\left(\sqrt{P\left(\boldsymbol{x}|\omega,t\right)} - \sqrt{P\left(\boldsymbol{x}|\omega',t\right)}\right)^2 \mathrm{d}\boldsymbol{x}} \\ &= \sqrt{1 - K\left(\omega,\omega',t\right)}. \end{aligned} \tag{5.2}$$

$K\left(\omega,\omega',t\right)$ はバタチャリア係数 [19] で、式 (5.3) で定義される。

$$K\left(\omega,\omega',t\right) = \int\sqrt{P\left(\boldsymbol{x}|\omega,t\right)P\left(\boldsymbol{x}|\omega',t\right)}\mathrm{d}\boldsymbol{x}\ \left(0 \leq K\left(\omega,\omega',t\right) \leq 1\right). \tag{5.3}$$

ヘリンジャー距離はユークリッド距離であり、値の範囲は $[0,1]$ となる。また、2 つの分布が同一である場合にのみ、$H\left(\omega,\omega',t\right) = 0$ を満たす。式 (3.3) の中の確率 $P\left(\boldsymbol{x}|\boldsymbol{x}_i^\omega t_i^\omega, \boldsymbol{x}_{i+1}^\omega t_{i+1}^\omega, t\right)$ に対し、$K\left(\omega,\omega',t\right)$ は閉形式として計算でき、式 (5.4) で与えられる。

$$K\left(\omega,\omega',t\right) = \frac{2\sigma^\omega\sigma^{\omega'}}{(\sigma^\omega)^2 + (\sigma^{\omega'})^2}\exp\left(-\frac{\left(\boldsymbol{\mu}^\omega - \boldsymbol{\mu}^{\omega'}\right)^2}{4\left((\sigma^\omega)^2 + (\sigma^{\omega'})^2\right)}\right) \tag{5.4}$$

時刻 t における $\boldsymbol{\mu}^\omega$ と σ^ω は、式 (3.5) と式 (3.6) で定義されている。また、式 (5.4) の分母が、$t = t_i^\omega = t_{i'}^{\omega'}$ となって 0 に近づいても、$K\left(\omega,\omega',t\right)$ は無限大に発散することはない。$[T_1, T_2]$ の間の距離を、ヘリンジャー距離の二乗 $H^2\left(\omega,\omega',t\right)$ から、式 (5.5) で定義する。

$$H^2\left(\omega,\omega'\right)=\frac{1}{T_2-T_1}\int_{T_1\leq t\leq T_2}\left(1-K\left(\omega,\omega',t\right)\right)\mathrm{d}t. \tag{5.5}$$

式 (5.5) の $H^2\left(\omega,\omega'\right)$ は閉形式ではないが、数値積分によって $H^2\left(\omega,\omega',t\right)$ の近似値から求めることができる。

5.3 階層的クラスタリングと時空間パターン抽出

ここでは、類似した時空間分布となるユーザをいくつかのクラスタに分けるために、クラスタリングを用いる。$H^2\left(\omega,\omega'\right)$ はユークリッド距離の二乗なので、$H\left(\omega,\omega'\right)$ に、階層的クラスタリングの一種である Ward 法 [36] [10] を適用できる。

Ward 法は凝縮型の階層的クラスタリングの一種で、各クラスタの分散が最小になるようにクラスタを構築していく。Ward 法では、クラスタの数を指定する必要があるが、これは適切なクラスタが求まるように調整する。分割されたクラスタのユーザのチェックインデータから、第 3 章の方法を用いて時空間パターンの抽出を行う。PCA は対象となる時空間分布全体の特性を抽出できるので、クラスタの特性を出すために ICA ではなく PCA を用いてパターン抽出を行う。まとめると、提案手法のアルゴリズムは以下のようになる。

1. 全てのユーザの組み合わせに対して $H^2\left(\omega,\omega'\right)$ を計算
2. ユーザ間の距離 $H\left(\omega,\omega'\right)$ に対して、Ward 法を用いて階層的クラスタを作成 (何個のクラスタに分けるかは結果を見て調整する)
3. それぞれのクラスタ内の全てのユーザから第 3 章の方法を用いて、時空間パターンを抽出。ただし、ここでは ICA ではなく PCA を用いる。

5.4 データ

この章では、2013 年 12 月 26 日から 2014 年 1 月 24 日までのおよそ 1ヶ月の東京周辺のチェックインデータを使用する。このデータは第 3 章で用いたデータと同じである。集め

表 5.1: 10 個に分けたクラスタの中のユーザの数、チェックイン数及びチェックインの平均回数

クラスタ	1	2	3	4	5
ユーザ数	1103	1407	1327	2296	4705
チェックイン数	41590	44267	42889	77860	208524
一人あたりの平均チェックイン数	37.7	31.5	32.3	33.9	44.3

クラスタ	6	7	8	9	10
ユーザ数	365	715	1603	999	1840
チェックイン数	8186	14077	39694	33745	47346
一人あたりの平均チェックイン数	22.4	19.7	24.8	33.8	25.7

たユーザの中で、期間中に 10 回以上のチェックインをしているユーザ 16,360 人の 558,178 回のチェックインデータを実験で使用する。$\sigma^2_{\mathrm{dist}} \to 0$ となることにより、$t = t_i$ と $t = t_{i+1}$ での式 (3.3) の無限大への発散を防ぐため、ここでも $\lambda\sigma^2_{\mathrm{dist}}$ の下限を λ_0 とした。

λ は、第 4 章の事前分布を用いる推定方法 (式 (4.10)) を用いて、実験で用いたデータから 2678.0 を得た。テストデータを用いてパラメータを調整した事になるが、λ によるモデルの適応は粗いので過剰適合にはなっていないと考えられる。

5.5 実験結果

5.2 節で説明したヘリンジャー距離を使った、5.3 節の階層的クラスタリングにより、ユーザをクラスタに分割した。クラスタの数としては、クラスタ毎の地域による特性の現れていた 10 を用いた。表 5.1 に、それぞれのクラスタに属するユーザ数、チェックイン数及び 1 ユーザあたりのチェックイン数の平均を示す。

図 5.1 にクラスタリングなしの時空間パターンを示す。SP1, SP2, SP3 は、3.3 節で説明した、上位 3 つの寄与率となる空間パターンである。TP1, TP2, TP3 は時系列パターンで、それぞれ SP1, SP2, SP3 に対応する。上が時間領域で、下が周波数領域でのパターンとなる。ユーザは主に、東京の中心部 (SP1, TP1=安定) におり、埼玉、池袋、お台場に年度末 (SP2, TP2=年末の急騰) に行っており、東京駅やお台場に定期的に行っている

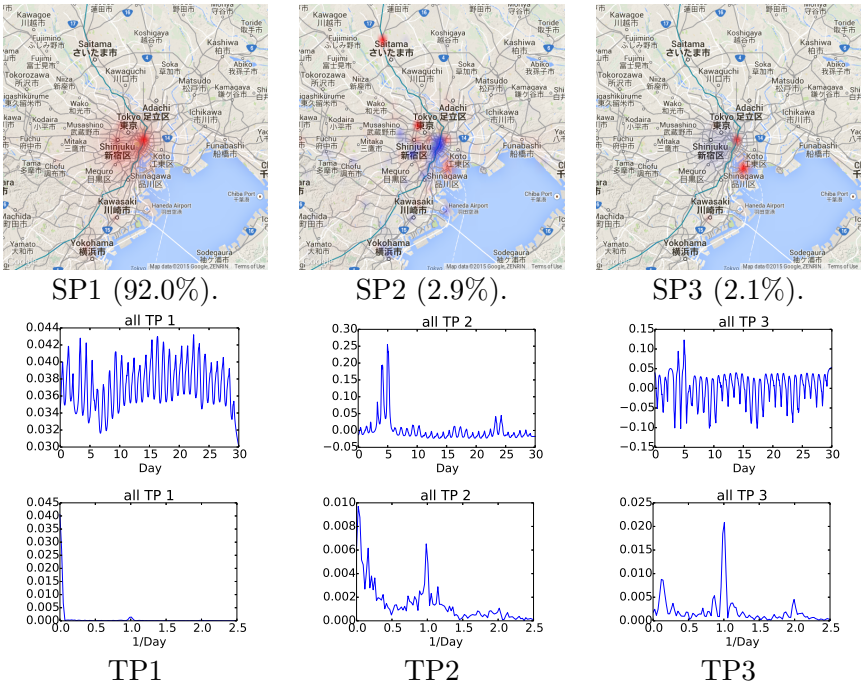


図 5.1: クラスタリングなしの空間パターン (SP1, SP2, SP3) と時系列パターン (TP1, TP2, TP3)

(SP3, TP3=1 日周期)。

図 5.2〜図 5.12 に、クラスタに分けた場合の時空間パターンの抽出の結果を示した。図 5.2 に全てのクラスタの分を示してある。SP2、TP2、SP3、TP3 はクラスタ毎に図 5.3〜図 5.12 に示してある。

これらのデータに基づいた各クラスタの分類を以下に示す。

- クラスタ 1(図 5.3): 東京と埼玉に定常的なパターンがあるユーザのクラスタ (SP1, TP1=安定)。お台場に年末に行き (SP2, TP2=年末の急騰)、東京とお台場に定期的に行っている周期的なパターンがある (SP3, TP3=1 日周期)。

  いわゆるオタクと呼ばれるユーザが、自分の場所を少なくとも 10 回以上ツイートしていると考えられる。お台場で年末に行われる “コミケ” と呼ばれる行事に、多くのオタクが参加している。全てのクラスタにお台場に関連したパターンが出てきたのは、これが原因と考えられる。

- クラスタ 2(図 5.4): クラスタ 2 のユーザは、池袋周辺に定常的なパターンがある。池袋は、東京の中の大きな商業や娯楽の街である (SP1, TP1=安定)。東京のその他の場所に定期的に行く周期的なパターンがあり (SP2, TP2=1 日周期)、池袋、東京駅、そしてお台場に年末に行く突発的なパターンがある (SP3, TP3=年末の急騰)。

- クラスタ 3(図 5.5): クラスタ 3 のユーザは、川崎周辺に定常的なパターンがある。川崎は東京中心部近くで、通勤・通学でよく利用される街である (SP1, TP1=安定)。また、年末に目黒と年始に目黒に行っている突発的なパターンがあり (SP2, TP2=年末と年始の増加)、羽田空港に定期的に行く周期的なパターンがある (SP3, TP3=1 日周期)。ユーザのツイートを見ると、クラスタ 3 のユーザは川崎周辺に住んでいるようであった。

- クラスタ 4(図 5.6): クラスタ 4 のユーザは、東京駅周辺に定常的なパターンがある。東京駅は東京の東部に位置する (SP1, TP2=安定)。東京駅に年度末に行く突発的なパターンがあり (SP2, TP2=年末の急騰)、東京駅とお台場に定期的に行く周期的なパターンがある (SP3, TP3=1 日周期)。

- クラスタ 5(図 5.7): クラスタ 5 のユーザは、調布付近に定常的なパターンがある。調布は通勤・通学でよく利用される街である (SP1, TP1=安定)。調布からお台場に年末に行く突発的なパターンがあり (SP2, TP2=年末の急騰、ただしはっきりとしていない)、お台場に定期的に行く周期的なパターンがある (SP3, TP3=1 日周期)。

- クラスタ 6(図 5.8): クラスタ 6 のユーザは、羽田空港周辺に定常的なパターンがある (SP1, TP1=安定)。東京、川崎、横浜に年末に行く突発的なパターンがあり (SP2, TP2=年末と年始の増加)、東京駅とお台場に部分的に定期的に行く周期的なパターンがある (SP3, TP3=1 日周期)。

- クラスタ 7(図 5.9): クラスタ 7 のユーザは、東京の東のエリアに定常的なパターンがある (SP1, TP1=安定)。魚市場で有名な築地に年末に行く突発的なパターンがあり (SP2, TP2=年末の急騰)、東京駅、お台場にも年末に行く突発的なパターンがある (SP3, TP3=年末の急騰)。ユーザのツイートによると、自分の居住している街に

帰るために、12 月 31 日に駅にいるようであった。クラスタ 7 のユーザは、東京以外の地域からコミケに参加するために来ていると考えられる。そのため、ユーザ一人あたりのチェックインも、他のクラスタに比べて低い値となっている。

- クラスタ 8(図 5.10): クラスタ 8 のユーザは、東京駅周辺に定常的なパターンがある (SP1, TP1=安定)、お台場に定期的に行く周期的なパターンがあり (SP2, TP2=1 日周期)、東京駅に年末に行く突発的なパターンがある (SP3, TP3=年末の急騰)。

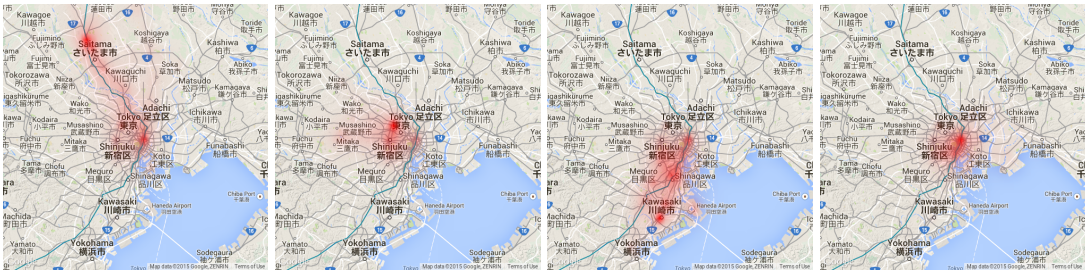
- クラスタ 9(図 5.11): クラスタ 9 のユーザは、横浜付近に定常的なパターンがある。横浜は、東京の横にある神奈川県で一番大きな街である (SP1, TP1=安定)。東京駅に定期的に行く周期的なパターンがあり (SP2, TP2=1 日周期)、横浜と東京駅に年末に行く突発的なパターンがある (SP3, TP3=年末の急騰)。

- クラスタ 10(図 5.12): クラスタ 10 のユーザは東京の中心部に定常的なパターンがある (SP1, TP1=安定)。東京のその他の地域に定期的に行く周期的なパターンがあり (SP2, TP2=1 日周期)、東京駅に年末に行く突発的なパターンがある (SP3, TP3=年末の急騰、ただしはっきりとしていない)。

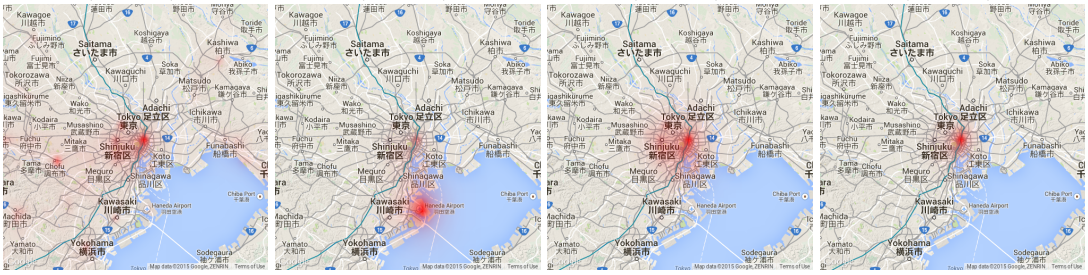
場所に関して述べると、クラスタ 1 は埼玉、クラスタ 3 は川崎と羽田、クラスタ 5 は調布、クラスタ 6 は羽田、クラスタ 9 は横浜、その他クラスタはおおよそ東京のパターンのクラスタとなっていた。クラスタ 2 は東京の西部で、クラスタ 4 は東京の東部のパターンのクラスタであった。クラスタ 7 は築地を特徴としており、クラスタ 8 は東京駅を特徴としたパターンのクラスタであった。クラスタ 10 は、銀座を含む東京中心部の毎日の動きを表していた。時系列パターンについて述べると、それぞれのクラスタの定常的なパターン、周期的なパターン、突発的なパターンを分けて抽出できた。また、お台場のパターンの影響をクラスタリングにより防ぐことができ、他の地域のパターンを抽出することができた。

このようにユーザをクラスタに分割することで、よりユーザの特徴を反映した時空間パターンを抽出することができた。クラスタリングにより類似したユーザを集めることにより、定常的なパターン、周期的なパターン、突発的なパターンを抽出できたので、ユーザ

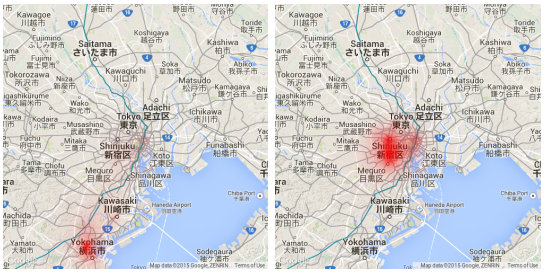
の位置予測においても、位置を予測したいユーザに類似したユーザのチェックインデータを使用することで、位置予測の精度を向上させる可能性がある。第 6 章では、この方法について述べる。



クラスタ 1, SP1 (77.4%).　クラスタ 2, SP1 (87.5%).　クラスタ 3, SP1 (84.5%).　クラスタ 4, SP1 (88.9%).

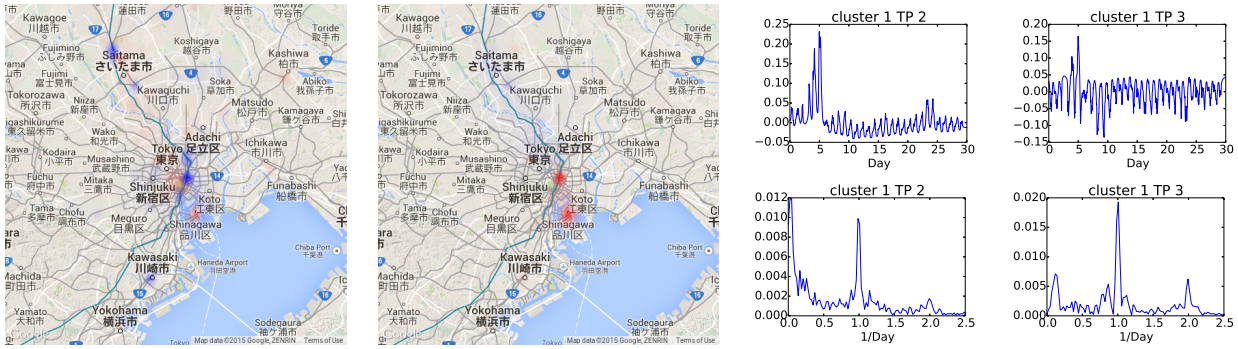


クラスタ 5, SP1 (74.3%).　クラスタ 6, SP1 (75.6%).　クラスタ 7, SP1 (70.9%).　クラスタ 8, SP1 (93.5%).



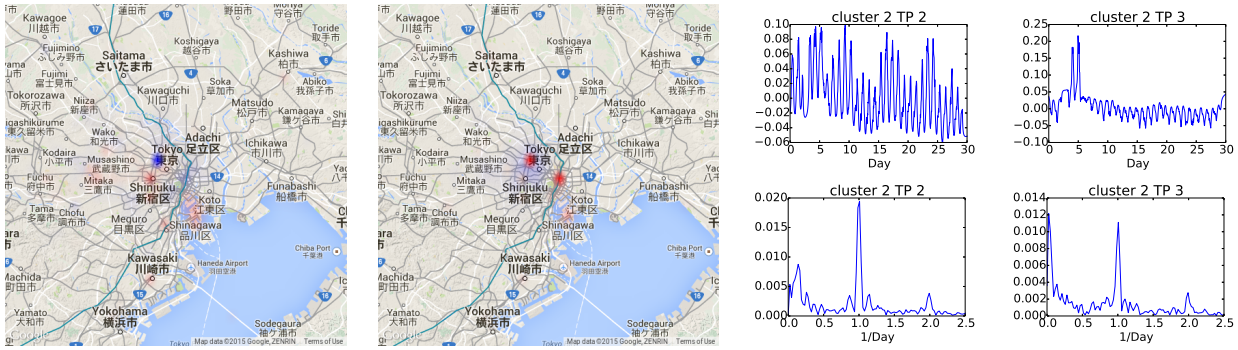
クラスタ 9, SP1 (84.9%).　クラスタ 10, SP1 (94.4%).

図 5.2: 10 個の空間パターン (SP1) とその寄与率。東京周辺のみを表示。赤色の濃さで通常より人が増加していることを示し、青色の濃さで通常より人が減少していることを示す。



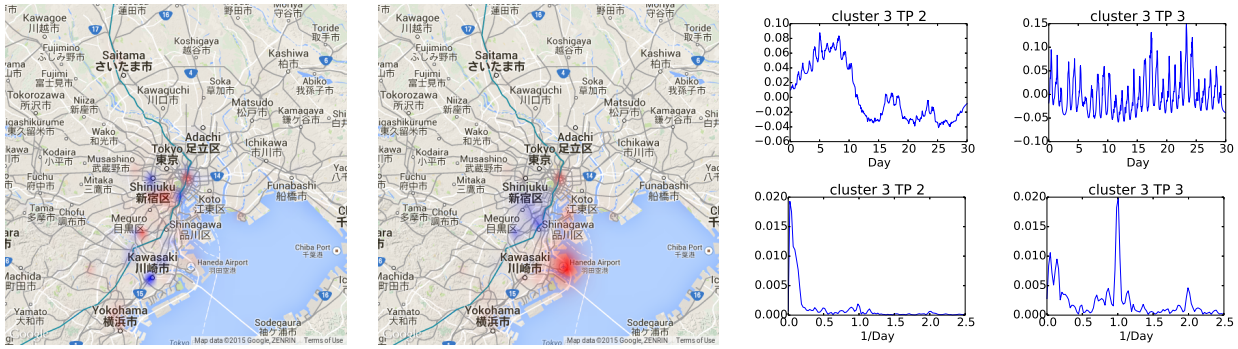
SP2 (2.6%). SP3 (1.9%). TP2 TP3

図 5.3: クラスター 1 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。



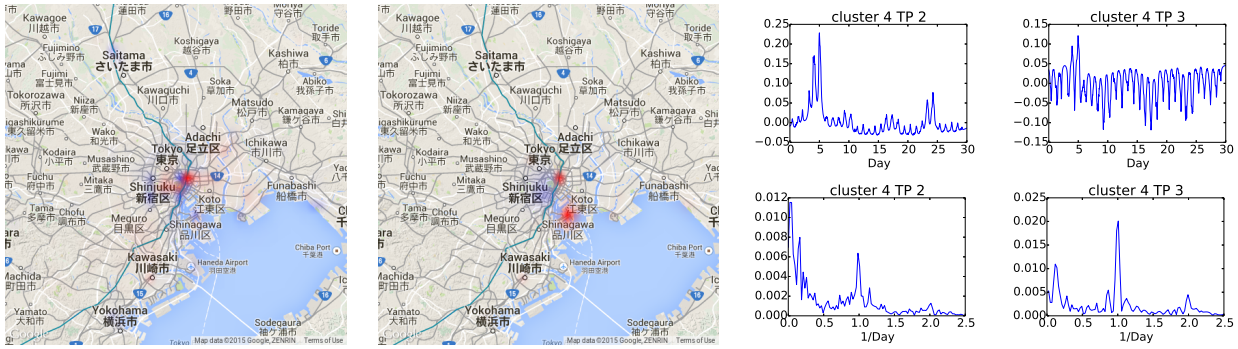
SP2 (1.8%). SP3 (1.4%). TP2 TP3

図 5.4: クラスター 2 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。



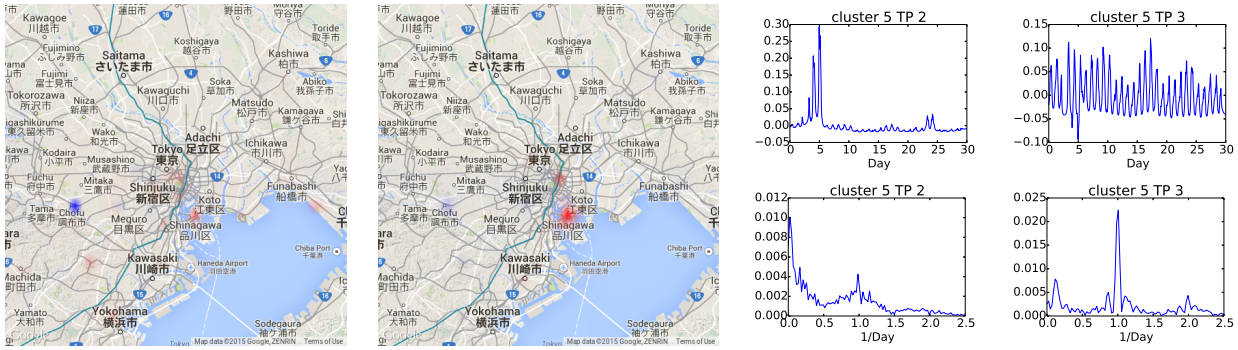
SP2 (3.3%). SP3 (1.3%). TP2 TP3

図 5.5: クラスター 3 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。



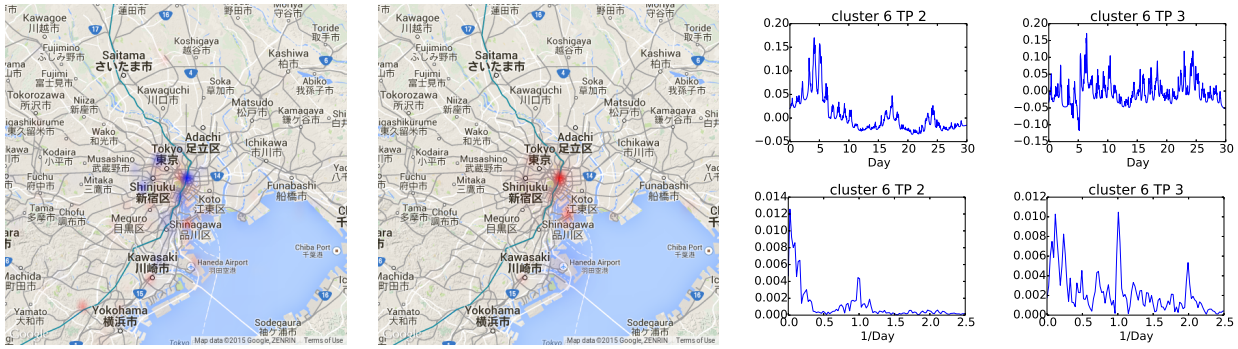
SP2 (2.4%). SP3 (1.8%). TP2 TP3

図 5.6: クラスター 4 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。



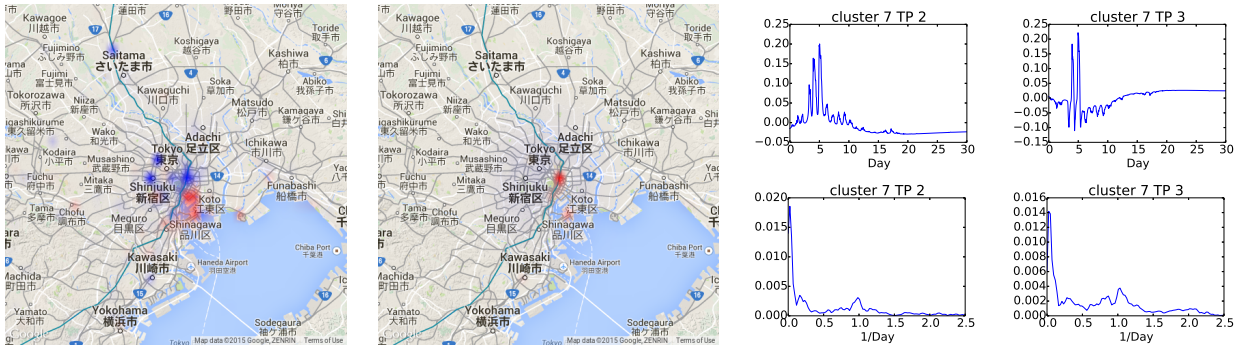
SP2 (4.3%). SP3 (2.9%). TP2 TP3

図 5.7: クラスター 5 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。



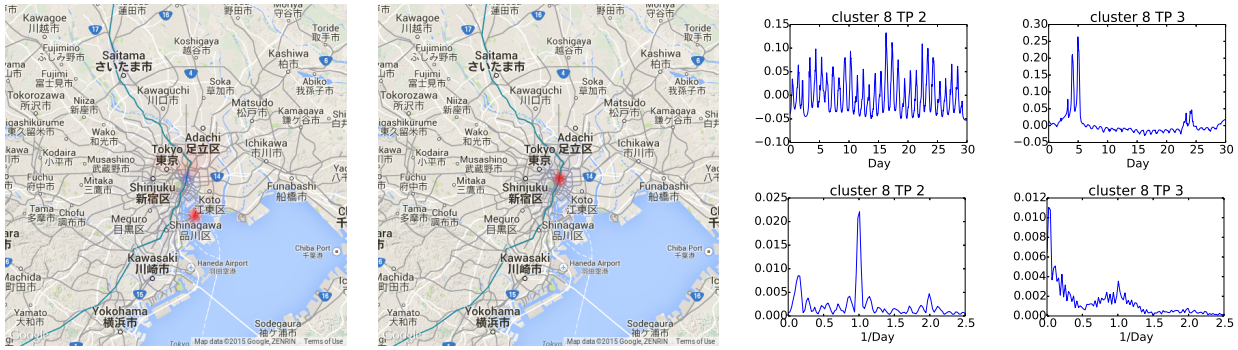
SP2 (3.8%). SP3 (1.9%). TP2 TP3

図 5.8: クラスター 6 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。



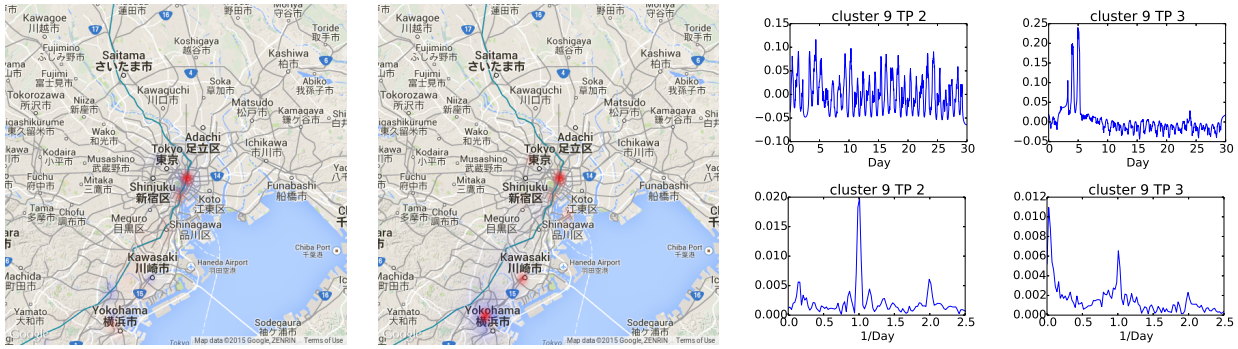
SP2 (10.2%). SP3 (6.5%). TP2 TP3

図 5.9: クラスター 7 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。



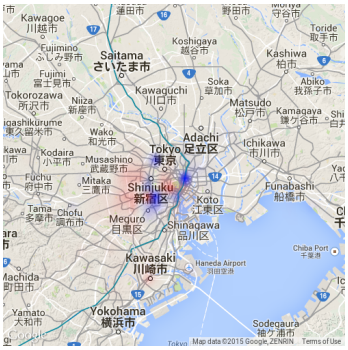
SP2 (2.2%). SP3 (1.2%). TP2 TP3

図 5.10: クラスター 8 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。



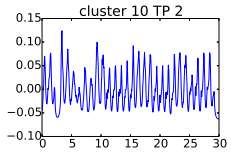
SP2 (1.7%). SP3 (1.3%). TP2 TP3

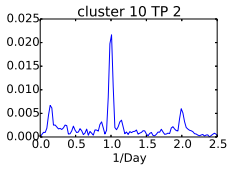
図 5.11: クラスター 9 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。



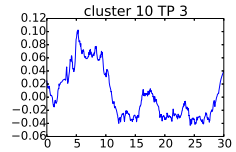
SP2 (1.2%).

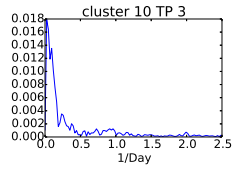
SP3 (0.8%).





TP2





TP3

図 5.12: クラスター 10 の時空間パターン (SP2, SP3, TP2, TP3) と寄与率。他は図 5.2 と同様。

第6章　特定周期の時空間パターンを用いたユーザの位置予測

6.1　概要

第3章で提案した手法で推定したユーザの存在確率の時空間分布は、チェックインデータの末尾の時刻以降のユーザの存在確率が時間とともに広がってくため、末尾のチェックインの後の位置を予測するのに適していない。

そこで本章では、第5章で抽出したユーザの時空間パターンに1日周期が多く観察されたことを利用し、1人のユーザのチェックインデータから予測するDPM (Diffusion-type Periodic Model) と、類似した複数のユーザのチェックインデータ予測するDPMU (Diffusion-type Periodic Model with similar Users)、第5章で行ったPCAを用いて、次元削減により時空間分布のノイズを取り除いたRDPMU (Reduced Diffusion-type Periodic Model with similar Users) を提案する。

予測モデルの性能を比較するために、確率の大きさを利用するlog-likelihoodと、順位を利用するRankとMRRの3つを評価指標として用いた結果を示す。

6.2　位置予測モデルDPM (Diffusion-type Periodic Model)

この節では、予測する対象ユーザωに対する拡散方程式と周期性を用いた予測モデルDPM (Diffusion-type Periodic Model) について述べる。

まず、$P(\boldsymbol{x}|t)$のtを、1時間ごとに離散化しておく。これにより、ユーザの距離 (式(5.3)) が計算できるようになる。

第3章と第5章でユーザのチェックインデータに1日周期のパターンが強く抽出できた

ので、ユーザの存在確率に 1 日の周期性を仮定した予測を作る。式でいえば、$P(\boldsymbol{x}|t+24)$ が $P(\boldsymbol{x}|t)$ と似ているということを仮定することになる。ここで、式中の 24 は、1 日が 24 時間であることに対応する。

実際のユーザが行動する時間の範囲は、24 時間以上の時間の長さがあるので、24 時間ごとに時空間分布を平均することにする。例えば、1/1 から 1/31 までのデータから 12:00 の分布を求める場合、1/1 12:00 の分布、1/2 12:00 の分布、... 、1/31 12:00 の分布を平均することにより 12:00 の分布を求める。DPM は式 (6.1) で定義する。

$$P_{DPM}(\boldsymbol{x}|\omega,t) = \frac{1}{M}\sum_{d=1}^{M} P(\boldsymbol{x}|\omega,t-24d) \tag{6.1}$$

M は学習として使うデータの期間の日数で、t は 1 から 24 の範囲の整数である。ここで、$1-24M,\cdots,0$ がチェックインデータの存在する期間の範囲とする。

6.3 l 人の類似したユーザを用いた予測モデル DPMU (Diffusion-type Periodic Model with similar Users)

6.2 節で提案したユーザの位置予測モデルは、予測対象となるユーザがまれに普段生活している場所と異なる位置でチェックインした場合 (例えば旅行に行ったなど) に、予測する分布がその「まれ」なチェックインデータの影響を受ける可能性がある。予測対象となるユーザと類似したユーザのチェックインデータを用いて予測を行うことで、このような変則的なチェックインの影響を抑えることができる可能性がある。そこで、対象ユーザと類似した $(l-1)$ 人のチェックインデータを用いた予測モデル DPMU$_l$ (Diffusion-type Periodic Model with similar Users) を導入する。DPMU$_l$ では予測対象のユーザ ω が、時刻 t に位置 $\boldsymbol{x}$ にいる確率 $P^l_{DPMU}(\boldsymbol{x}|\omega,t)$ としては、ω を含めて l 人の類似したユーザの DPM により予測される空間分布を平均したものを使用する。DPMU$_l$ を式 (6.2) で定義する。

$$P^{l}_{DPMU}\left(\boldsymbol{x}|\omega,t\right)=\frac{1}{l}\sum_{\psi\in\Omega_{\omega,l}}P_{DPM}(\boldsymbol{x}|\psi,t) \tag{6.2}$$

ここで、$\Omega_{\omega,l}\in\{\omega,\psi^{\omega}_{1},\psi^{\omega}_{2},...,\psi^{\omega}_{l-1}\}$ は、予測対象のユーザ ω とユーザ ω に類似したユーザの上位 $l-1$ 人のユーザの集合である。ユーザ間の類似度は、5.2 節で定義したヘリンジャー距離を利用している。$l=1$ の場合は、DPM と一致する。つまり DPM は DPMU_1 と一致する。

6.4 データ

この章では、学習とテストに十分な量のデータを取得するため、3.4 節と同様に Twitter streaming 経由にて、東京周辺の Foursquare のチェックインデータを、2014 年 4 月 7 日から 2014 年 7 月 11 日までのおよそ三ヶ月間を集めた。最初の二ヶ月をトレーニングデータ、残りをテストデータとして利用する。集めたユーザの中で、期間中に 60 回以上のチェックインをしているユーザ 4,096 人のチェックインを実験で使用する。

プログラムは C++にて記述した。計算では 60 コアを用いて並列計算を行った。$\sigma^{2}_{\mathrm{dist}}\to 0$ となることにより、$t=t_i$ と $t=t_{i+1}$ で、式 (3.3) が無限大への発散を防ぐため、ここでも $\lambda\sigma^{2}_{\mathrm{dist}}$ の下限を λ_0 に制限した。

6.5 比較対象

比較の対象として、単純ガウス分布を用いた予測モデルと、先行研究で用いられていた PMM [9] の 2 つを使用する。

ガウス分布を用いた予測モデル

ガウス分布を用いた予測モデルは、チェックインの位置が 1 つの二次元ガウス分布に基づいて分布すると仮定する。このモデルでは、チェックインの時刻を考慮しない。

$$P_{Gaussian}\left(\boldsymbol{x}\right) = \mathcal{N}_x\left(\boldsymbol{\mu}_{\text{dist}}, \Sigma_{\text{dist}}\right) \tag{6.3}$$

ここで、$\mathcal{N}_x\left(\boldsymbol{\mu}, \Sigma\right)$ は二次元のガウス関数である。チェックインデータをトレーニングデータとして用いて、元の二次元ガウス分布の推定を行う。

$\boldsymbol{\mu}_{\text{dist}}$ は、式 (6.4) を用いて推定する。

$$\boldsymbol{\mu}_{\text{dist}} = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i}{n} \tag{6.4}$$

ここで、n はトレーニングデータのチェックインの数である。Σ_{dist} は、式 (6.5) を用いて推定する。

$$\Sigma_{\text{dist}} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{pmatrix} \tag{6.5}$$

これは、二次元空間における、チェックインの緯度経度の共分散行列である。

PMM

PMM は、文献 [9] にて提案された手法である。PMM は、HOME 状態と WORK 状態という 2 つの状態を持っており、ユーザがどちらの状態にいるかを確率的に決めていく。どちらの状態も、二次元のガウス分布としてチェックインデータの中の位置が分布するとしており、時刻によって、HOME 状態と WORK 状態のどちらの状態にいるかの確率が変化するというモデルとなっている。PMM は、このように時刻により HOME 状態と WORK 状態の確率分布が変化するという時間成分と、チェックインデータの位置が HOME 状態と WORK 状態それぞれのガウス分布に従って広がるという空間成分の 2 つを組み合わせたモデルである。時間成分は以下のように定義される。$c_u(t) = H$ は、時刻 t にユーザが HOME 状態にあることを表し、$c_u(t) = W$ は、時刻 t にユーザが WORK 状態にあるこ

とを表す。$c_u(t) = H$ と $c_u(t) = W$ の確率は、式 (6.6) と式 (6.7) で定義される。

$$P[c_u(t) = H] = \frac{N_H(t)}{N_H(t) + N_W(t)}, \tag{6.6}$$

$$P[c_u(t) = W] = \frac{N_W(t)}{N_H(t) + N_W(t)}, \tag{6.7}$$

ここで、$N_H(t)$ と $N_W(t)$ はそれぞれ式 (6.8) と式 (6.9) で定義される。

$$N_H(t) = \frac{P_{cH}}{\sqrt{2\pi\sigma_H^2}} \exp\left[-(\frac{\pi}{12})^2 \frac{(t-\tau_H)^2}{2\sigma_H^2}\right] \tag{6.8}$$

$$N_W(t) = \frac{P_{cW}}{\sqrt{2\pi\sigma_W^2}} \exp\left[-(\frac{\pi}{12})^2 \frac{(t-\tau_W)^2}{2\sigma_W^2}\right] \tag{6.9}$$

ここで、τ_H は、1 日における HOME 状態のチェックイン時刻の平均値で、σ_H^2 は、1 日における HOME 状態のチェックイン時刻の分散[1]となる。また、P_{cH} は HOME 状態によって生成されるチェックインの確率である。同様に、τ_W と σ_W^2 と P_{cW} は、WORK 状態のチェックイン時刻の平均値と分散及びチェックインの生成確率を表す。PMM は、この時間成分に空間成分加えた式 (6.10) で定義される。

$$\begin{aligned} &P_{PMM}\left(\boldsymbol{x}|\boldsymbol{x}_1t_1, \boldsymbol{x}_2t_2, ..., \boldsymbol{x}_nt_n, t\right) \\ =&P[c_u(t) = H]\mathcal{N}_x\left(\boldsymbol{\mu}_H, \Sigma_H\right) + \\ &P[c_u(t) = W]\mathcal{N}_x\left(\boldsymbol{\mu}_W, \Sigma_W\right) \end{aligned} \tag{6.10}$$

ここで、$\mathcal{N}_x\left(\boldsymbol{\mu}_H, \Sigma_H\right)$ と $\mathcal{N}_x\left(\boldsymbol{\mu}_W, \Sigma_W\right)$ は二次元のガウス関数である。$\boldsymbol{\mu}_H$ と Σ_H は、それぞれ HOME 状態のチェックインの位置の平均値と共分散行列を表す。$\boldsymbol{\mu}_W$ と Σ_W も同様に、それぞれ WORK 状態のチェックインの位置の平均値と共分散行列を表す。$P[c_u(t) = H]$ と $P[c_u(t) = W]$ は、式 (6.6) と式 (6.7) で定義されたユーザの HOME 状態と WORK 状態の確率である。

τ_H などのパラメータは実際のチェックインデータの尤度が最大となるように EM アル

[1] 24 時間周期があるので厳密な分散ではない

ゴリズムで求められる [9]。

6.6 評価手法

この節では、予測モデルをどのように評価するかを述べる。位置予測の使い方として、以下の二種類が考えられる。

1. 予測された位置の確率が特定の値以上の場合、つまり高い確率でユーザがいると予想された時のみ利用。予測位置が多くなりすぎることを排除したい場合に使用する。

2. 予測された位置の候補を大きい方から順に使用。確率が低くても何かしら予測結果を出さないといけない場合に使用する。

よって、予測モデルの能力を測るためには、実際に予測した位置の確率を利用した指標と、予測した位置が他の予測位置と比較して何番目にあるかという順位を利用した指標の二種類を用いる必要がある。本論文では、前者の指標として log-likelihood、後者の指標として Rank (mean Rank) と MRR (Mean Reciprocal Rank) を用いた。

以下で、log-likelihood、Rank、MRR を定義する。

$(\boldsymbol{x}_1^\omega t_1, \boldsymbol{x}_2^\omega t_2, \cdots, \boldsymbol{x}_{n^\omega}^\omega t_{n^\omega}^\omega)$ は、予測するユーザ ω のテストデータのチェックイン列を表す。$\mathrm{RANK}(P, \boldsymbol{x}_k^\omega, t_k^\omega)$ を、ユーザ ω の実際のチェックインの位置を含む領域の中心の確率の順位とする。

log-likelihood

log-likelihood は、実際のチェックインデータ中の位置の予測したモデルにおける確率の対数の平均である。log-likelihood は、likelihood と違い、アンダーフローしにくいことが性質としてあげられる。ユーザ ω に対する log-likelihood は式 (6.11) で定義する。

$$\mathcal{L}(\omega) = \frac{1}{n^\omega} \sum_{k=1}^{n^\omega} \log P(\boldsymbol{x}_k^\omega, t_k^\omega). \tag{6.11}$$

Rank

Rank は、限られた範囲内を格子状に分け、格子状の領域で、実際のチェックインデータ中の位置を含む領域の中心の確率が上位何番目かを求め、その平均値を利用する。Rank は、ユーザが確率の絶対値そのものを使わない予測位置のリストなどの評価には便利であるが、外れ値の影響を受けやすい評価手法である。ユーザ ω に対する Rank は式 (6.12) で定義する。

$$\mathcal{R}(\omega) = \frac{1}{n^\omega} \sum_{k=1}^{n^\omega} \mathrm{RANK}(P, \boldsymbol{x}_k^\omega, t_k^\omega). \tag{6.12}$$

Mean Reciprocal Rank

MRR は、順位の逆数の平均値を用いることで外れ値の影響を受けにくい評価指標となっている。ユーザ ω に対する MRR (Mean Reciprocal Rank) は式 (6.13) で定義する。

$$\mathcal{M}(\omega) = \frac{1}{n^\omega} \sum_{k=1}^{n^\omega} \frac{1}{\mathrm{RANK}(P, \boldsymbol{x}_k^\omega, t_k^\omega)}. \tag{6.13}$$

log-likelihood と MRR では、値が大きい方が良い予測となり、逆に Rank では値が小さいほうが良い予測となる。

本論文では、予測したいユーザの位置がある空間を格子の領域に分割して評価を行う。

6.7 実験結果

本節では、サンプリングした 100 人のユーザのチェックインデータを用いて予測手法を log-likelihood、Rank、MRR で評価する。100 人のユーザに限定したのは、計算時間を抑えるためである。DPMU_l の l 人のユーザは 4096 人のユーザの中から選ぶ。DPMU_l の l として、$1, 2, 3, 4, 5, 6, 11, 21, 31, 41, 51, 61, 71, 81, 91, 101, 111, 121, 131, 141, 151, 201, 251, 301, 401, 501, 1001, 2001, 3001, 4096$ を使用して実験を行った。 緯度を N 34.5°-36.0° 、経度を E 139.0°-140.5° の範囲に限定し、この範囲を 100×100 の領域 (1 つの領域は約 1.5km $\times$ 1.5km) に分割して実験を行った。

log-likelihood

まず始めに、DPMU_l の log-likelihood とガウス分布を用いた予測モデルと PMM の log-likelihood を比較する。ガウス分布を用いた予測モデルと PMM の log-likelihood はそれぞれ -6.66、-5.75 となった。DPMU_l の log-likelihood の平均値と、ガウス分布を用いた予測モデルと PMM の log-likelihood を図 6.1 示す。DPMU_l の log-likelihood の平均値は、サンプリングした 100 人のユーザから計算した。$2 \leq l \leq 2001$ においては、DPMU_l はガウス分布および PMM よりも良い結果となった。予測する対象ユーザただ一人のデータを利用して予測を行った DPMU_1 は、ガウス分布を用いた予測モデルより log-likelihood で良い結果を出したが、PMM より悪い結果となった。これは、似たユーザを集めることで、log-likelihood を向上させることができることを意味する。DPMU_l は、$l = 81$ のときに最も良い結果となり、log-likelihood は -5.33 となる。

Rank

図 6.2 に、DPMU_l と PMM とガウス分布を用いた予測モデルの Rank を示す。DPMU は、$l = 401$ で最も良い結果となり、ガウス分布や PMM より良く、Rank は 155.2 となる。$100 \times 100 = 10000$ 個の領域があるので、Rank は 1 から 10000 の間となる。PMM とガウス分布を用いた予測モデルの Rank は、それぞれ 186.5 と 204.2 となった。$l \leq 2001$ において DPMU_l の Rank は、PMM よりも良い結果となった。

MRR

図 6.3 に、DPMU_l と PMM とガウス分布を用いた予測モデルの MRR を示す。$l = 1$ で最も良い 0.256 となり、ガウス分布と PMM より良くなった。PMM とガウス分布を用いた予測モデルの MRR は、それぞれ 0.177 と 0.06 となった。他の評価手法と違い MRR では、類似したユーザを集めることで結果が悪くなっている。ここで MRR が他の指標と逆の結果となっているのは、MRR が特に外れ値を低く評価し、予測が当たっている場合を高く評価していることが要因と考えられる。類似したユーザのチェックインデータを用い

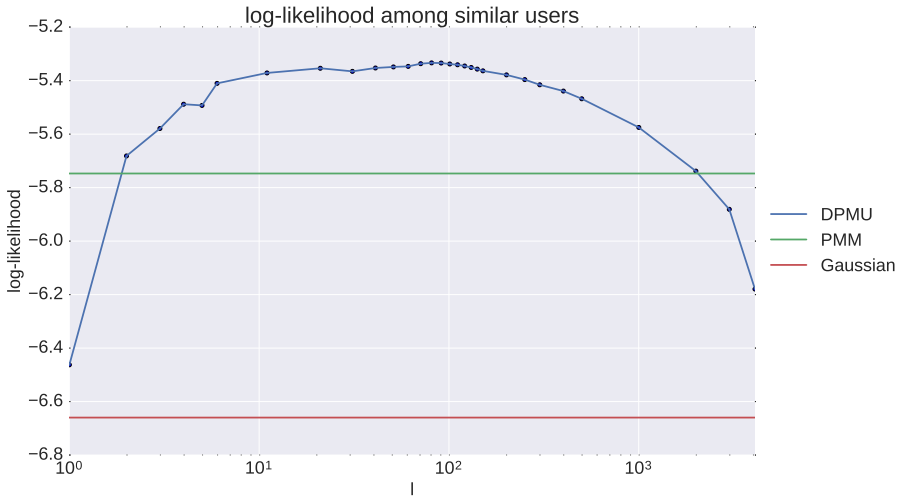


図 6.1: log-likelihood による比較。x 軸は、予測対象となるユーザを含む予測に使った類似したユーザの数。x 軸は対数軸となっている。

ることで、ユーザの存在確率の分布は平滑化されていき、Rank の値が低いものは高くなり、Rank の値が高いものは低くなる。ここで、MRR は Rank の値が低い予測を重視した評価であるため、Rank の値が高い予測が低くなることよりも、Rank の値が低い予測が高くなることの影響が大きくなり、その結果平滑化の影響で MRR が低くなると考えられる。例えば、平滑化される前は Rank が $1, 2, 9$ 位となっている 3 つのチェックインがあり、平滑後に全て Rank が 3 位となった場合、Rank の平均値は 4 から 3 となり向上する (値としては下がるが Rank では予測向上) が、MRR は $(1/1 + 1/2 + 1/9)/3 = 29/54$ から $(1/3 + 1/3 + 1/3)/3 = 1/3$ と低下する。

log-likelihood、Rank、MRR の標準偏差

ユーザによってどの程度予測にばらつきがあるかを調べるために、log-likelihood と Rank と MRR の標準偏差を調べた。標準偏差は 100 人のサンプリングしたユーザから計算した。

図 6.4 は、DPMU_l の log-likelihood の標準偏差を表している。$l = 81$ の時、log-likelihood の標準偏差は 1.06 であった。$l > 81$ の時、log-likelihood は徐々に減少しており (図 6.1)、標準偏差もまた減少している (図 6.4)。ここから、ユーザを多く集めすぎたことで、大き

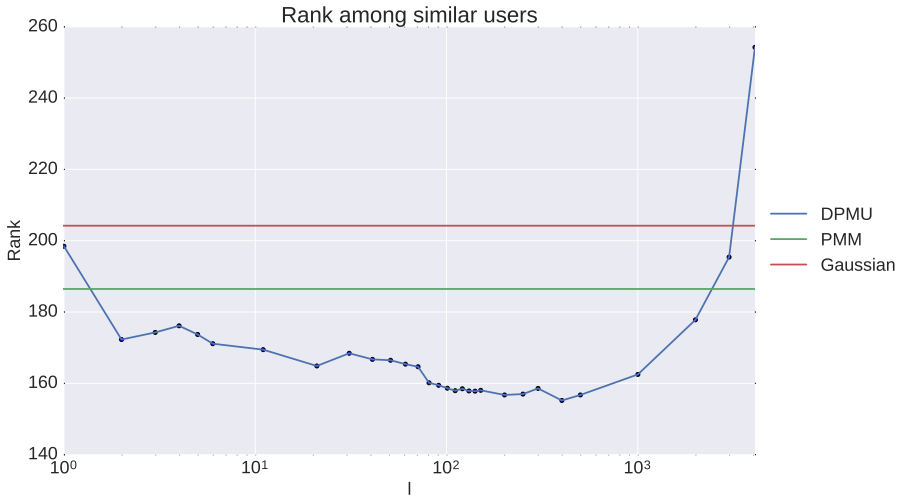


図 6.2: Rank による比較。x 軸は、予測対象となるユーザを含む予測に使った類似したユーザの数。x 軸は対数軸となっている。

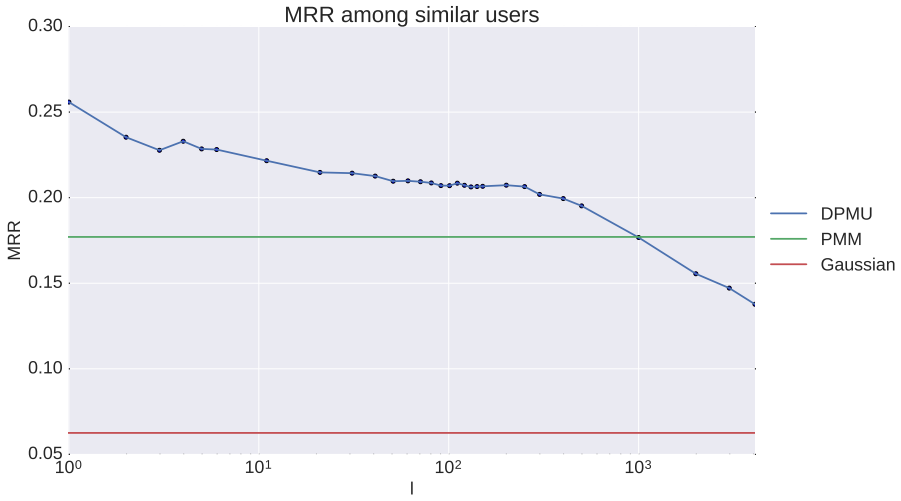


図 6.3: MRR による比較。x 軸は、予測対象となるユーザを含む予測に使った類似したユーザの数。x 軸は対数軸となっている。

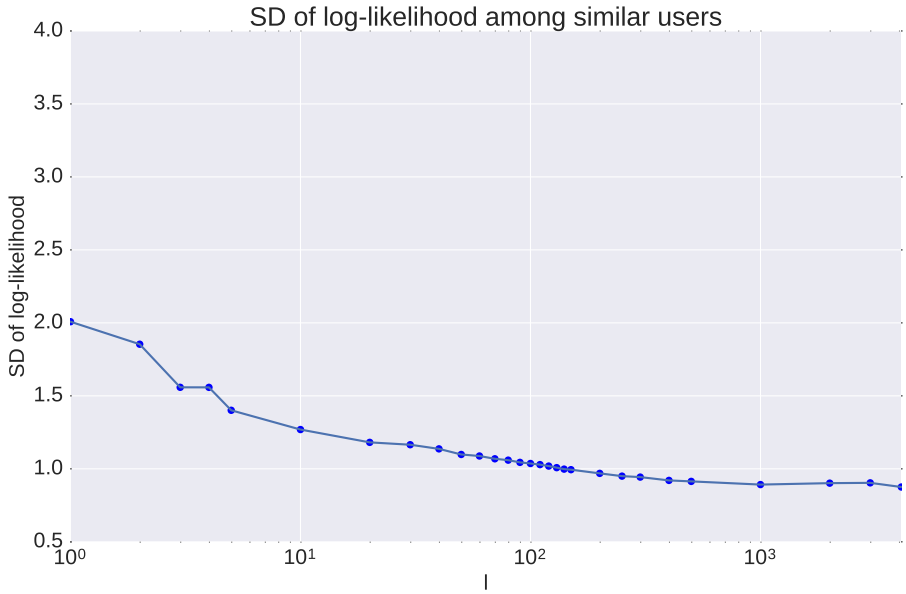


図 6.4: サンプリングした 140 人のユーザの log-likelihood の標準偏差。x 軸は、予測対象となるユーザを含む予測に使った類似したユーザの数。x 軸は対数軸となっている。

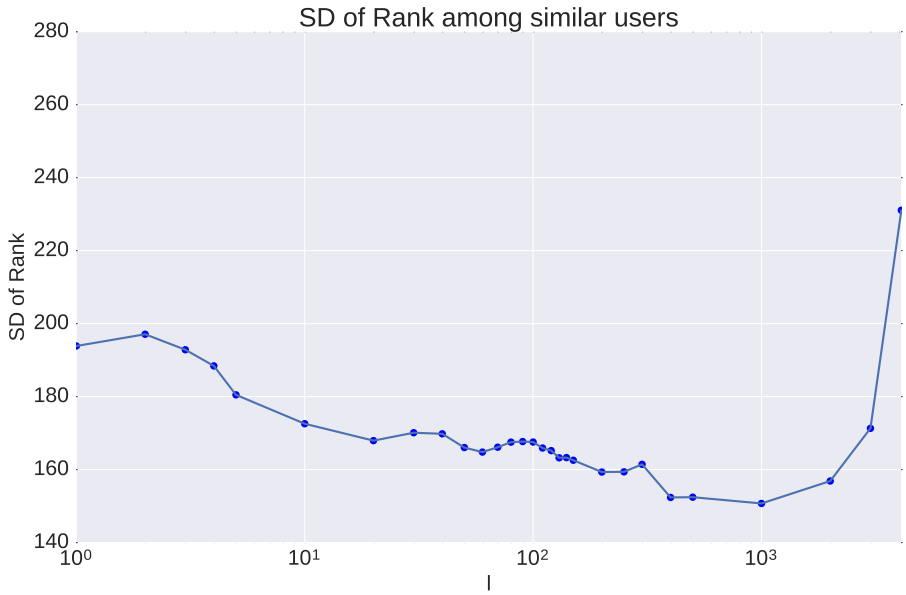


図 6.5: サンプリングした 140 人のユーザの Rank の標準偏差。x 軸は、予測対象となるユーザを含む予測に使った類似したユーザの数。x 軸は対数軸となっている。

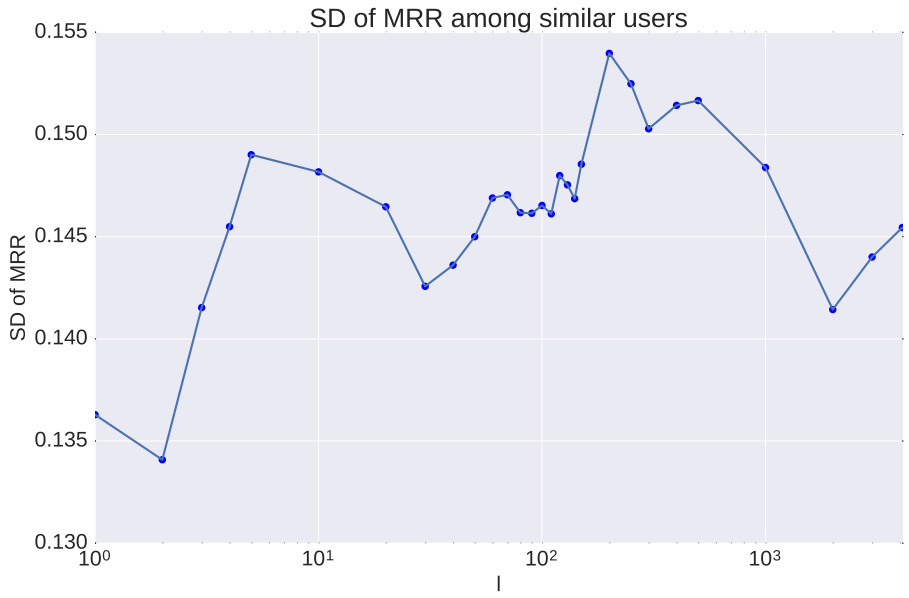


図 6.6: サンプリングした 140 人のユーザの MRR の標準偏差。x 軸は、予測対象となるユーザを含む予測に使った類似したユーザの数。x 軸は対数軸となっている。

い l では全てのユーザの位置予測が同一となり、予測するユーザの特有の行動を反映しにくい時空間分布となるため、log-likelihood が下がったと考えられる。

図 6.5 は、DPMU_l の Rank の標準偏差を表している。$l = 401$ の時、Rank の標準偏差は 152.4 であった。$l = 1001$ まで Rank の標準偏差はおおよそ下がる傾向があり、$l = 1001$ 以降になると増加している (図 6.5)。l が大きい場合、DPMU_l が類似したユーザだけでなく類似していないユーザのチェックインデータを用いるため、Rank が大きくなってしまい (図 6.1 右端の急騰)、標準偏差が非常に大きくなってしまうと考えられる。

図 6.6 は、DPMU_l の MRR の標準偏差を表している。$l = 1$ のとき MRR の標準偏差は最大値 0.153 となる。

この節で、いくつかの l で提案手法が既存手法より、よい結果が出ることを示した。類似したユーザのチェックインデータを使用することは、log-likelihood と Rank を向上させるが、MRR を低下させることが分かった。

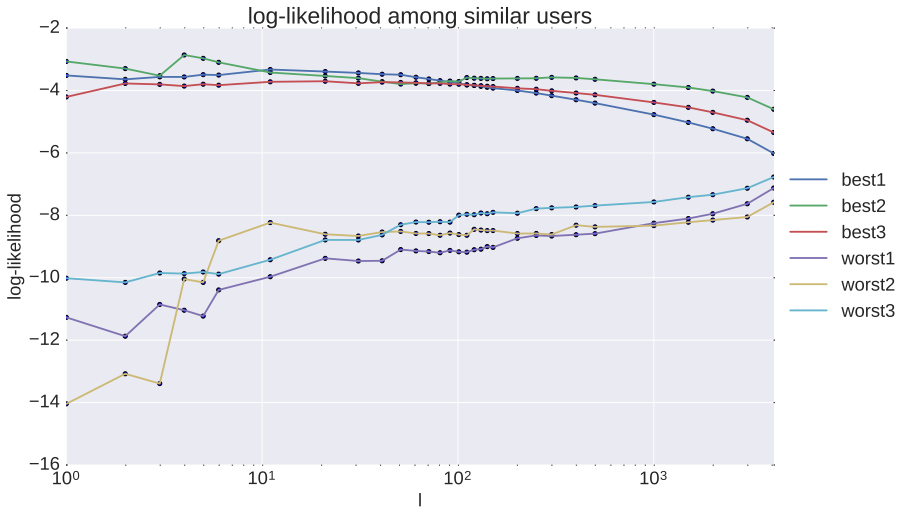


図 6.7: log-likelihood が上位 3 人と下位 3 人の log-likelihood の推移。x 軸は対数軸となっている。

6.8 ユーザによる予測の違い

6.8.1 log-likelihood、Rank、MRR の高いユーザと低いユーザに対する l の影響

6.7 節において、log-likelihood、Rank、MRR の標準偏差が大きかったことより、ユーザによって予測結果のばらつきが大きいことが分かった。そこで、本節では、予測がうまくいっているユーザとうまくいっていないユーザが、どんな性質を持ったユーザであるかを調べるために、各指標で上位と下位のユーザの log-likelihood、Rank、MRR を調べた。

log-likelihood

図 6.7 に log-likelihood の全体平均が大きくなる $l = 81$ のときに、log-likelihood の上位 3 人と下位 3 人のユーザの、l による log-likelihood の変化を示した。l が小さいときは、上位ユーザと下位ユーザのばらつきはかなり大きいが、l が大きくなるに連れ、その差が小さくなっていることが分かる。これは標準偏差で見た場合と同じような傾向と言える。上位ユーザは $l = 100$ で、上位 3 人のユーザの log-likelihood が近づくが、$l > 100$ にてば

らつきが大きくなっており、全体として l を大きくすると、log-likelihood が悪くなっていると言える。逆に、下位 3 人のユーザは、l を大きくすることで log-likelihood を上げていることが分かる。よって、log-likelihood が他のユーザに比べて低いユーザに対しては、類似したユーザのチェックインデータを使用することが、log-likelihood を向上させるのに有効であると考えられる。実際、下位 20 人のユーザを調査したところ、20 人中 16 人が類似したユーザのチェックインデータを使用することで、log-likelihood が向上していた。

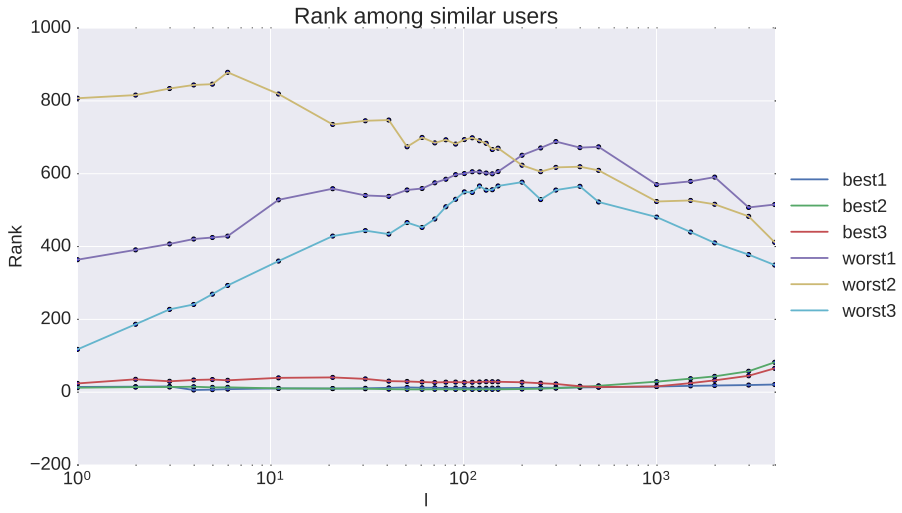


図 6.8: Rank が上位 3 人と下位 3 人の Rank の推移。x 軸は対数軸となっている。

Rank

図 6.8 に Rank の全体平均が大きくなる $l = 401$ のときに、Rank が上位 3 人と下位 3 人のユーザの、l による Rank の変化について示した。上位 3 人は、l によって多少変化が見られるものの、下位 3 人のユーザに比べて変化が少ない。下位ユーザは、下位 2 位、下位 3 位のユーザが $l = 401$ 付近まで Rank が悪くなり、逆に $l > 401$ から Rank が良くなっている。ただし、$l = 0$ のときに比べ、どちらも Rank が悪くなっている。下位 1 位のユーザは、概ね l を大きくすることによって Rank の向上が見られる。下位 3 人は l が大きくなることで、それぞれの Rank の値は近づいたが、ユーザによって Rank の上下する傾向が異なり、全体として共通する傾向は見られなかった。

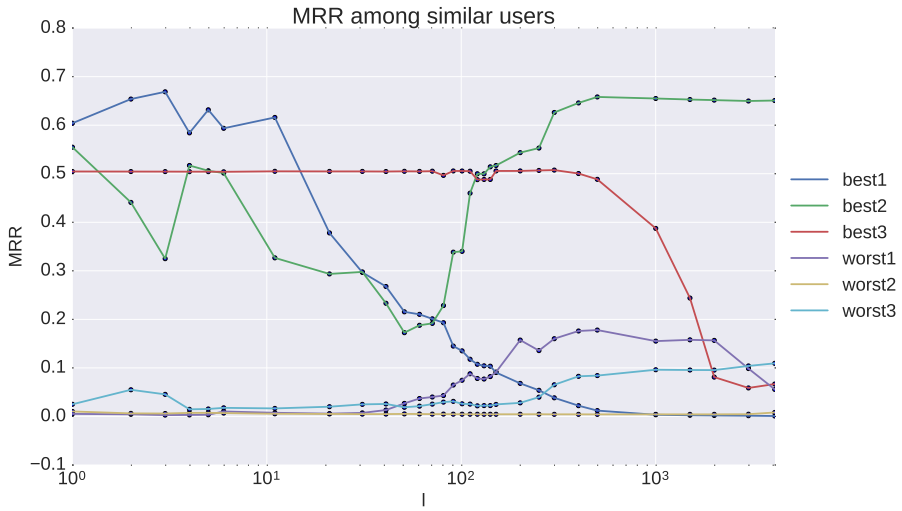


図 6.9: MRR が上位 3 人と下位 3 人の MRR の推移。x 軸は対数軸となっている。

MRR

図 6.9 に、MRR の全体平均が大きくなる $l = 1$ のときに、MRR が上位 3 人と下位 3 人のユーザの、l による MRR の変化について示した。l を大きくすると、上位 1 位と上位 3 位のユーザは、概ね MRR が下がる傾向にあった。上位 2 位のユーザは、$l = 51$ までは下がる傾向にあり、$l > 51$ で MRR が上がっていき、最終的に元の MRR より高くなった。下位ユーザは、下位 2 位のユーザを除き $l > 100$ で MRR の向上が見られた。ユーザによってかなり傾向のばらつきがあるものの、類似したユーザのチェックインデータを使用することで、MRR の向上が期待できる場合があることが分かった。

	latitude	longitude
best1	0.039426	0.066180
best2	0.029400	0.095591
best3	0.103189	0.108112
worst1	0.101451	0.231835
worst2	0.110385	0.135389
worst3	0.071641	0.132098

表 6.1: log-likelihood が上位 3 人、下位 3 人のチェックインの位置の標準偏差

6.8.2 log-likelihood、Rank、MRR の高いユーザと低いユーザの実際のチェックインの位置の分布

実際のチェックインの位置の分布の log-likelihood、Rank、MRR に対する影響を調べるため、ユーザのチェックインの位置のばらつきについて議論する。

log-likelihood

図 6.10 に log-likelihood が上位 3 人、下位 3 人のユーザのチェックインの分布を、表 6.1 にそのチェックインの位置の標準偏差を示した。全体として、上位ユーザに比べ下位ユーザのチェックインの位置はばらつきが大きくなっていることが分かる。図 6.10 より、上位ユーザは下位ユーザに比べて狭い範囲でチェックインしているようである。実際、上位のユーザは、下位のユーザよりチェックインの標準偏差が小さい傾向にある (表 6.1)。特に、経度方向の標準偏差は、上位 3 人とも下位 3 人より小さい範囲を動いていると言える。移動が多いユーザに対して、この DPMU は log-likelihood が悪くなる傾向があると考えられる。

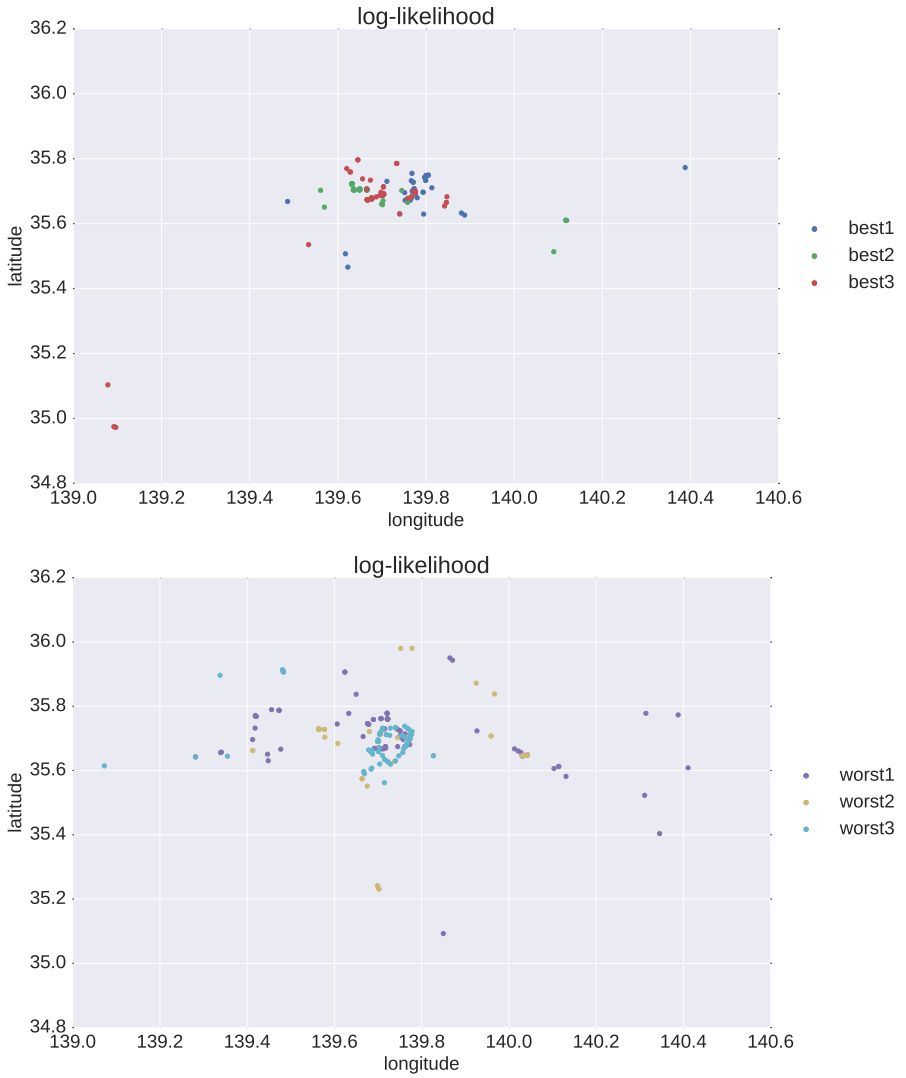


図 6.10: log-likelihood が上位 3 人、下位 3 人のチェックイン座標の分布。

	latitude	longitude
best1	0.103189	0.108112
best2	0.029400	0.095591
best3	0.026405	0.077940
worst1	0.110385	0.135389
worst2	0.089132	0.093235
worst3	0.101451	0.231835

表 6.2: Rank が上位 3 人、下位 3 人のチェックインの位置の標準偏差

Rank

図 6.11 に Rank が上位 3 人、下位 3 人のユーザのチェックインの位置の分布を、表 6.2 にそのチェックインの位置の標準偏差を示した。表 6.2 より、log-likelihood と異なり、Rank ではチェックインの位置の標準偏差が大きくても Rank は高く出ている。実際、図 6.11 を見ると、Rank が高かったユーザは、標準偏差は高いものの、チェックインの位置は狭い範囲にまとまっていることが観察できる。つまりこのユーザは、複数のチェックインの位置が集中する拠点を持っているユーザであると考えられる。下位 2 位のユーザは、上位 1 位のユーザに比べて、チェックインの位置の標準偏差自体は小さくなっているが、チェックインの位置が集中する拠点が複数に分布しており、Rank が下がってしまっていると考えられる。また、log-likelihood と同様に、上位ユーザは経度方向の標準偏差が下位ユーザに比べて小さい傾向にあるようであるが、下位 2 位のユーザのように必ずしも標準偏差が小さいからといって Rank が高くなるわけではないようである。

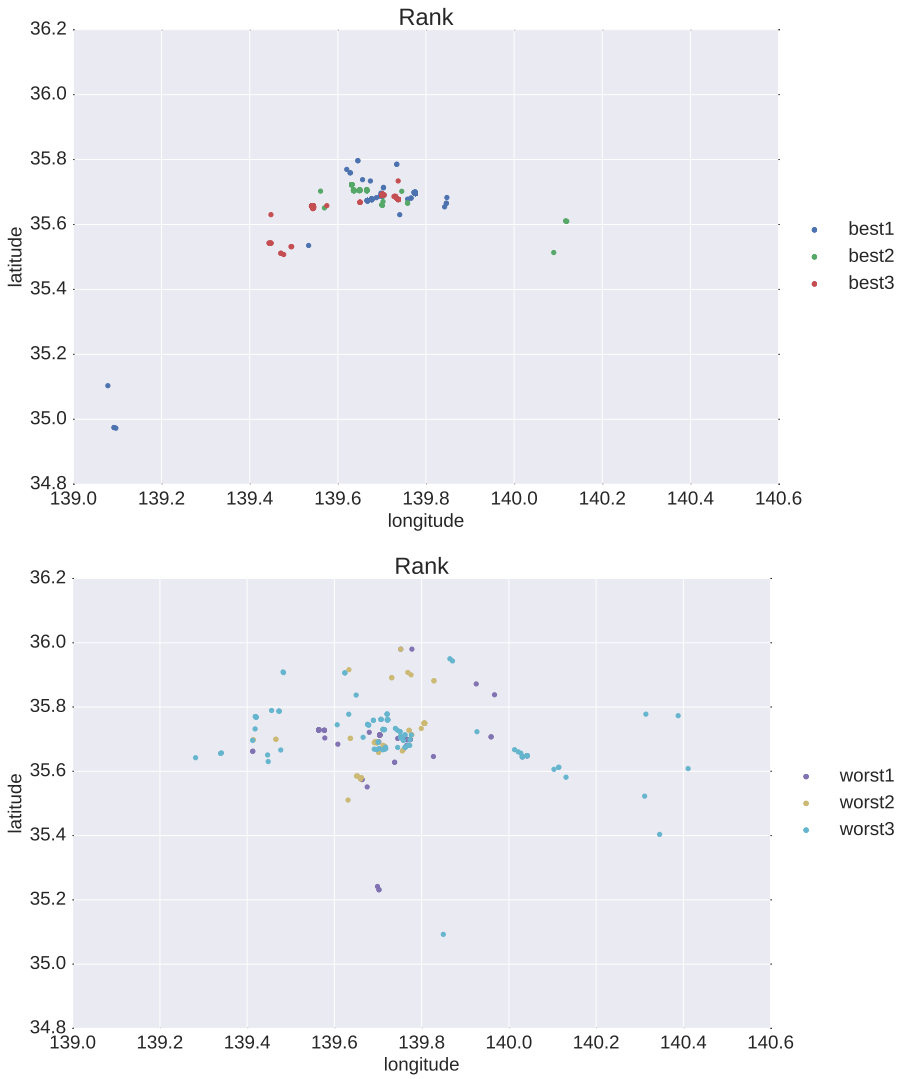


図 6.11: Rank が上位 3 人、下位 3 人のチェックイン座標の分布。

	latitude	longitude
best1	0.122534	0.112291
best2	0.067498	0.125712
best3	0.103189	0.108112
worst1	0.084838	0.136797
worst2	0.101451	0.231835
worst3	0.106873	0.165115

表 6.3: MRR が上位 3 人、下位 3 人のチェックインの位置の標準偏差

MRR

図 6.10 に MRR が上位 3 人、下位 3 人のユーザのチェックインの位置の分布を、表 6.3 にそのチェックインの位置の標準偏差を示した。表 6.3 より、MRR は Rank と同様に、チェックインの位置の標準偏差が大きくても MRR が高く出ていることが分かる。N 35.6°-35.8°、E 139.6°-139.8° の範囲によくチェックインするユーザが、log-likelihood、Rank において MRR が上位に来ていたが、MRR では上位 2 位のユーザのように、その範囲以外のユーザ (N 35.4°、E 139.3° 付近にチェックインするユーザ) でも高い MRR となっていた (図 6.9)。

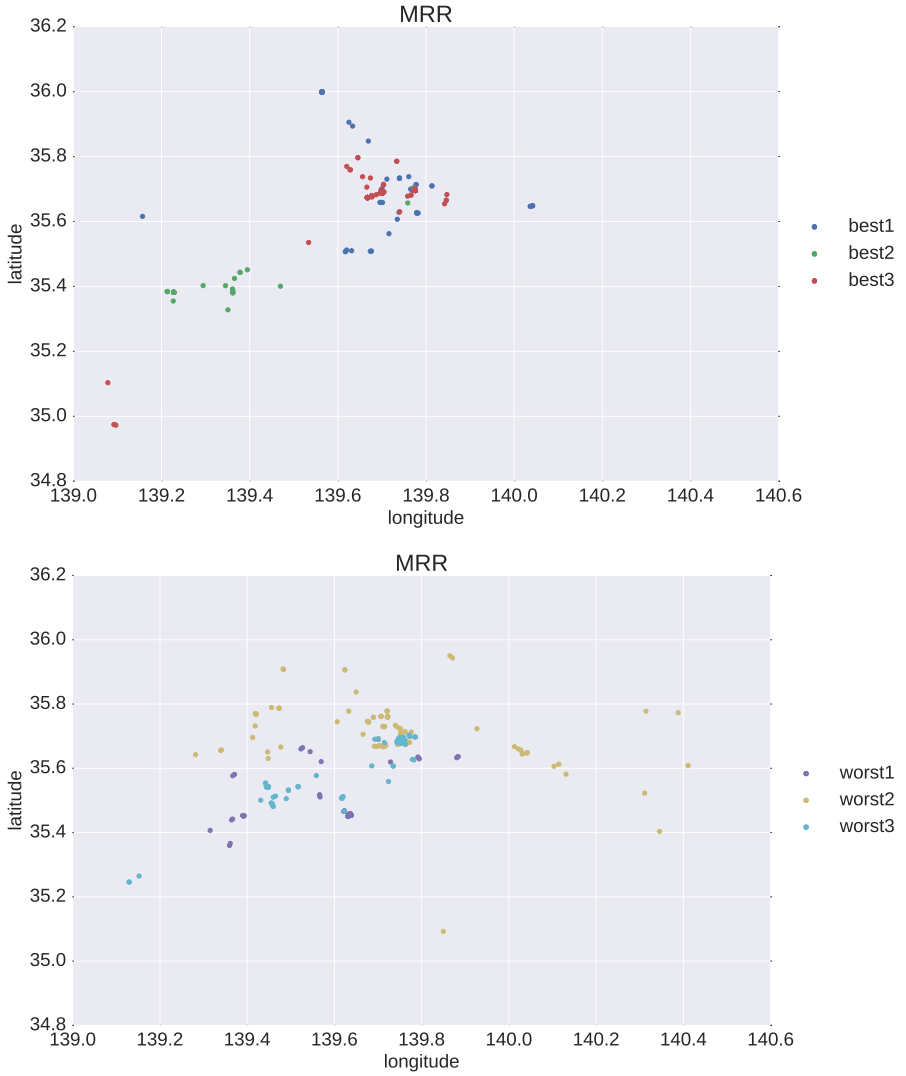


図 6.12: MRR が上位 3 人、下位 3 人のチェックイン座標の分布。

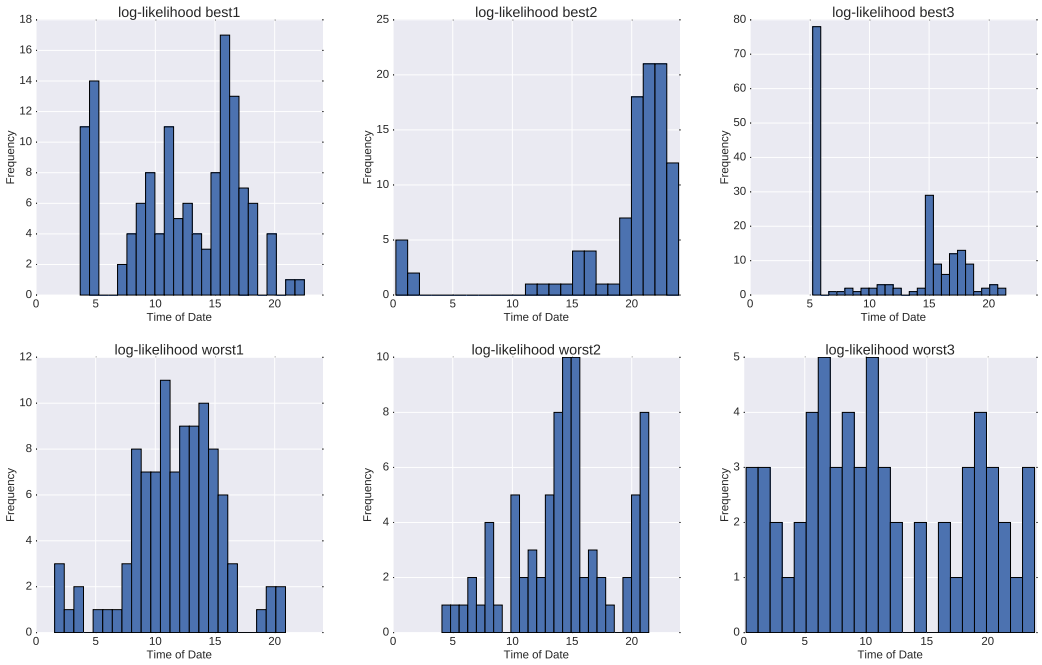


図 6.13: log-likelihood が上位 3 人、下位 3 人のチェックイン時間の分布。

6.8.3 log-likelihood、Rank、MRR が高いユーザと低いユーザの実際のチェックイン時刻の分布

本節では、log-likelihood、Rank、MRR が高いユーザと低いユーザのチェックインの時刻について議論をする。

log-likelihood

図 6.13 に log-likelihood が上位 3 人、下位 3 人のユーザのチェックインの時刻の分布を、表 6.4 にチェックイン時刻の平均値と標準偏差を示した。上位ユーザ、下位全ユーザとしてチェックインの時刻は様々に分布しており、共通した傾向は見られなかった。上位ユーザは、下位ユーザに比べてチェックインのピーク時のチェックイン回数が多い。これは同時刻にチェックインすることでユーザの傾向を抽出しやすくなっているからと考えられる。

	mean	std
best1	12.168757	4.829618
best2	19.293322	5.664249
best3	11.327550	5.451912
worst1	11.699651	3.838814
worst2	14.287676	4.266091
worst3	11.267354	6.809869

表 6.4: log-likelihood が上位 3 人、下位 3 人のチェックイン時刻の平均値と標準偏差

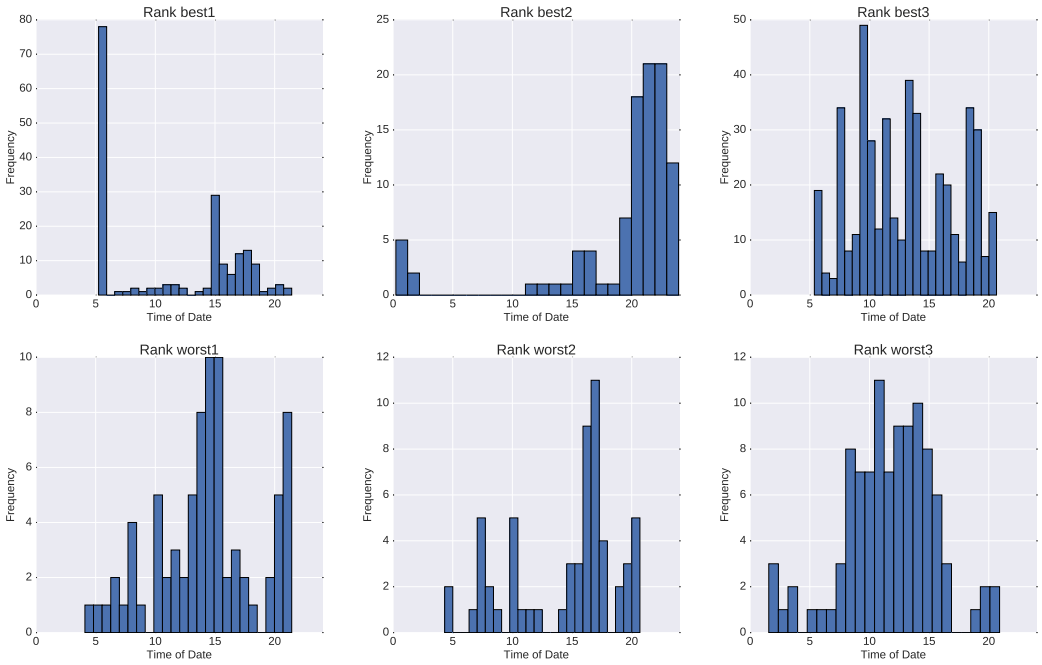


図 6.14: Rank が上位 3 人、下位 3 人のチェックイン時間の分布。

Rank

図 6.14 に Rank が上位 3 人、下位 3 人のユーザのチェックインの時刻の分布を、表 6.5 にチェックイン時刻の平均値と標準偏差を示した。Rank では、上位 1 位、上位 2 位のユーザは、それぞれ朝と夜の特定の時間帯に集中してチェックインを行っているユーザであった。特定の時間帯に特定の位置でチェックインしたほうが、Rank としてぶれが小さくなるからと考えられる。上位 3 位のユーザは、チェックインの時刻がばらついているが、表 6.2 に示したように、チェックインする位置の範囲が狭くなっているため、Rank が高くなっていると考えられる。log-likelihood と同様に、下位ユーザは上位ユーザに比べてチェックインのピークが低くなっている。

	mean	std
best1	11.327550	5.451912
best2	19.293322	5.664249
best3	13.015835	4.094960
worst1	14.287676	4.266091
worst2	14.533542	4.432380
worst3	11.699651	3.838814

表 6.5: Rank が上位 3 人、下位 3 人のチェックイン時刻の平均値と標準偏差

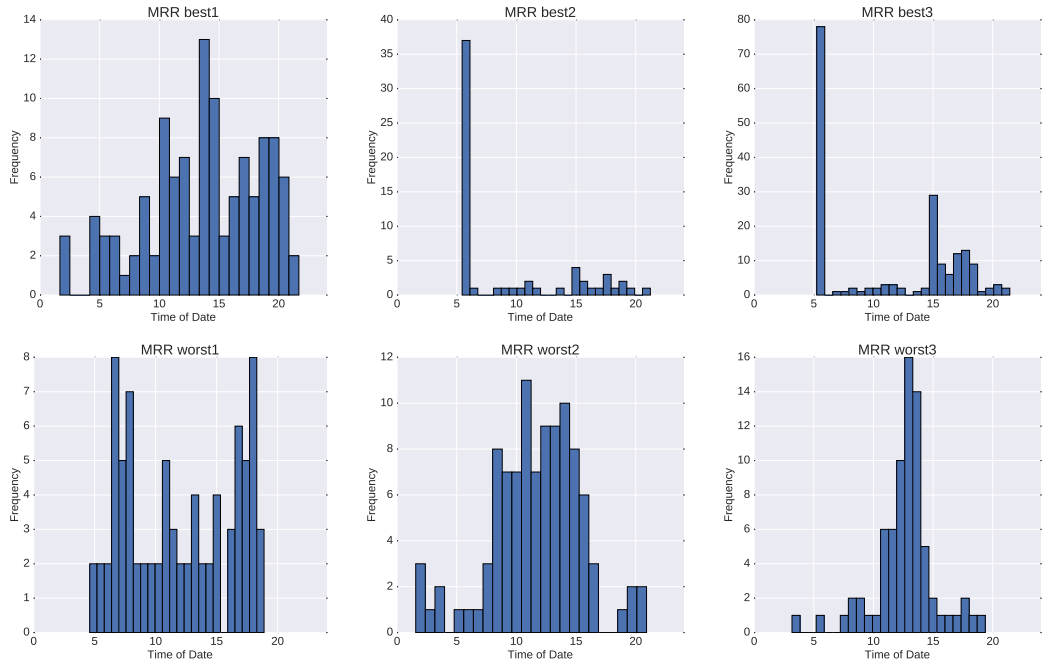


図 6.15: MRR が上位 3 人、下位 3 人のチェックイン時間の分布。

MRR

図 6.15 に MRR が上位 3 人、下位 3 人のユーザのチェックインの時刻の分布を、表 6.6 にチェックインの時刻の平均値と標準偏差を示した。MRR では、上位 2 位、上位 3 位のユーザは特定の時間帯にチェックインが集中しているようである。log-likelihood、Rank と同様に、下位ユーザは上位ユーザに比べてチェックイン回数のピークが低くなっている。

	mean	std
best1	13.609012	4.827200
best2	9.220909	5.106852
best3	11.327550	5.451912
worst1	12.009682	4.437101
worst2	11.699651	3.838814
worst3	12.685567	2.530708

表 6.6: MRR が上位 3 人、下位 3 人のチェックイン時刻の平均値と標準偏差

6.9 時空間分布の次元削減の影響

6.9.1 次元削減を用いた予測モデル RDPMU (Reduced Diffusion-type Periodic Model with similar Users)

推定したユーザの時空間分布には、第 5 章で行ったように PCA を適用することができる。PCA を適用し成分の次元を削減することにより、ノイズを取り除くことができることが期待される。本節では、これを用いて時空間分布を次元削減を利用した DPMU の拡張した RDPMU (Reduced Diffusion-type Periodic Model with similar Users) を提案する。

DPMU で利用していた l 人のユーザから推定した時空間分布を、3.3 節で行ったように PCA で成分分解し (式 (3.8))、寄与率の上位 M 個を残し、時空間分布の次元削減を行う。3.3 節の固有値の平方根 ν_m、時系列パターン V_m、空間パターン U_m を用いて、次元削減をした時空間分布は、式 (6.14) で定義する。

$$P\left(\boldsymbol{x}|t\right) = \sum_{m=1}^{M} \nu_m U_m\left(\boldsymbol{x}\right) V_m\left(t\right) \tag{6.14}$$

式 (6.1) を用いて 24 時間毎にこの時空間分布を平均したものを $P_{RDPMU}(\boldsymbol{x}|\omega, t)$ と定義する。

6.9.2 次元削減による予測の変化

図 6.16、図 6.17、図 6.18 に、それぞれ次元削減したときの log-likelihood、Rank、MRR の値のヒートマップを示した。x 軸が次元数 M を表し、y 軸が l 人のユーザを用いて時空間分布を推定したことを表す。また、色の濃さで各指標の値を表すので、log-likelihood と MRR では色が濃いほうが予測がよく、逆に Rank では色が薄いほうが予測がよい。

log-likelihood は、$l = 11, M = 6$ のとき最も良い -5.26 となった (図 6.16)。DPMU では、$l = 81$ のとき log-likelihood が最もよい -5.33 であったので、次元削減により log-likelihood は若干向上しているが、その差は小さいのでほとんど変わらなかったといえる。$l = 1$ では、M を大きくすることで log-likelihood が上がることがわかったが、$l = 1$ 以

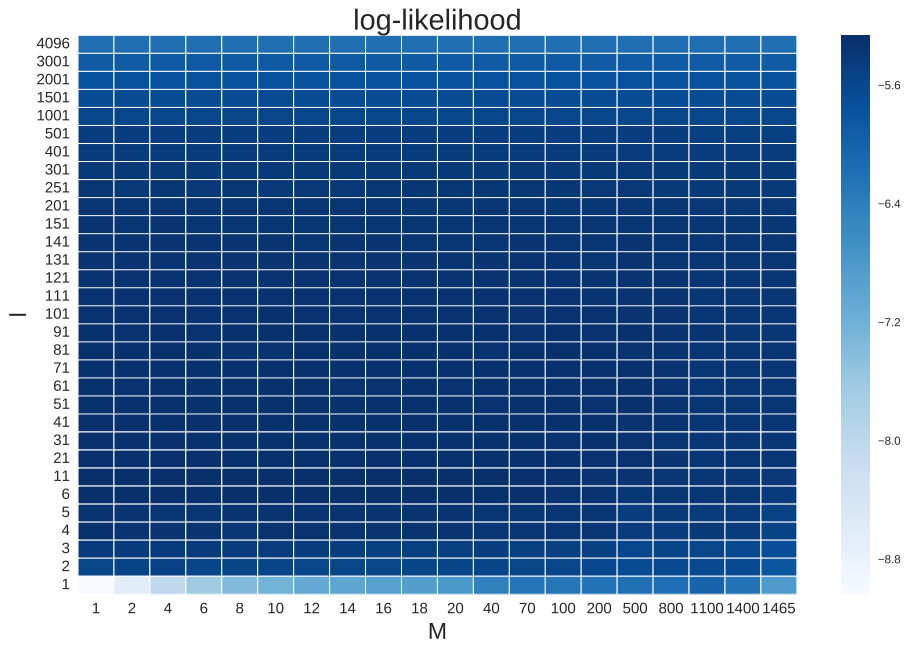


図 6.16: 次元削減の log-likelihoood への影響。x 軸が寄与率の高い順に成分を M 個使用したことを表し、y 軸が次元削減する前の時空間分布を作成するときに使用したユーザ l 人を表す。青色の濃さが log-likelihoood における値の大きさを表す。

外では、log-likelihood は M によってあまり変化をしなかった。Rank は、$l = 11, M = 1$ のとき最も良い 153.9 となった (図 6.17)。DPMU では、$l = 401$ のとき Rank が最も良い 155.2 であったので、Rank も次元削減により若干向上したが、log-likelihood 同様に差が微小であるので、ほとんど変わらなかったといえる。MRR は、$l = 1, M = 4$ のときに最も良い 0.285 となった (図 6.18)。DPMU では、$l = 1$ のとき MRR が最も良い 0.256 であったので、log-likelihood、Rank と同様に若干の向上が見られたが、差が小さいのでほとんど変わらなかったといえる。

次元削減によって、log-likelihood、Rank、MRR の全てにおいて若干の向上が見られたが、次元削減しない場合のベストパフォーマンスとの差は全て微小であった。

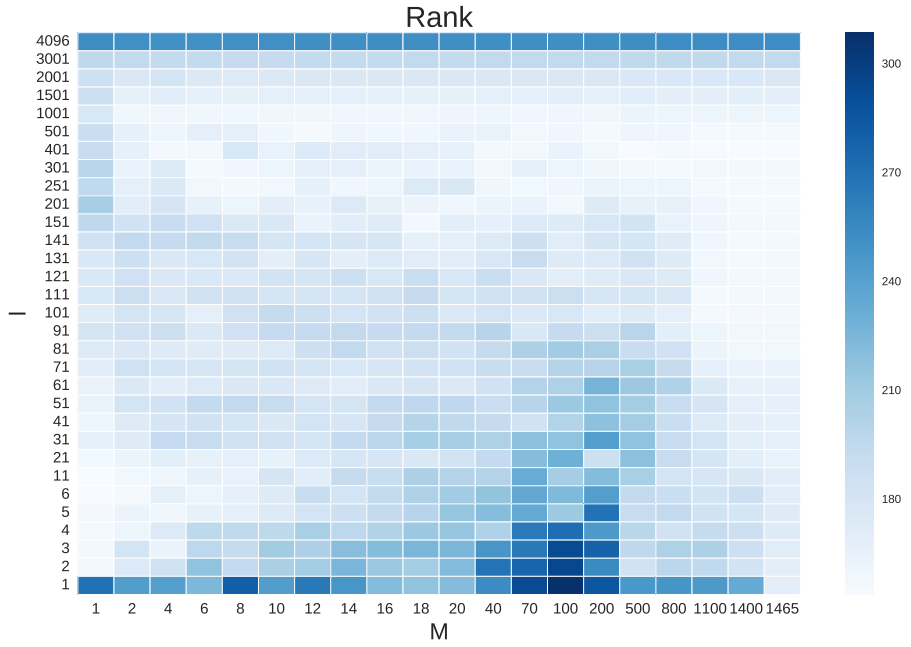


図 6.17: 次元削減の Rank への影響。x 軸が寄与率の高い順に成分を M 個使用したことを表し、y 軸が次元削減する前の時空間分布を作成するときに使用したユーザ l 人を表す。青色の濃さが Rank における値の大きさを表す。

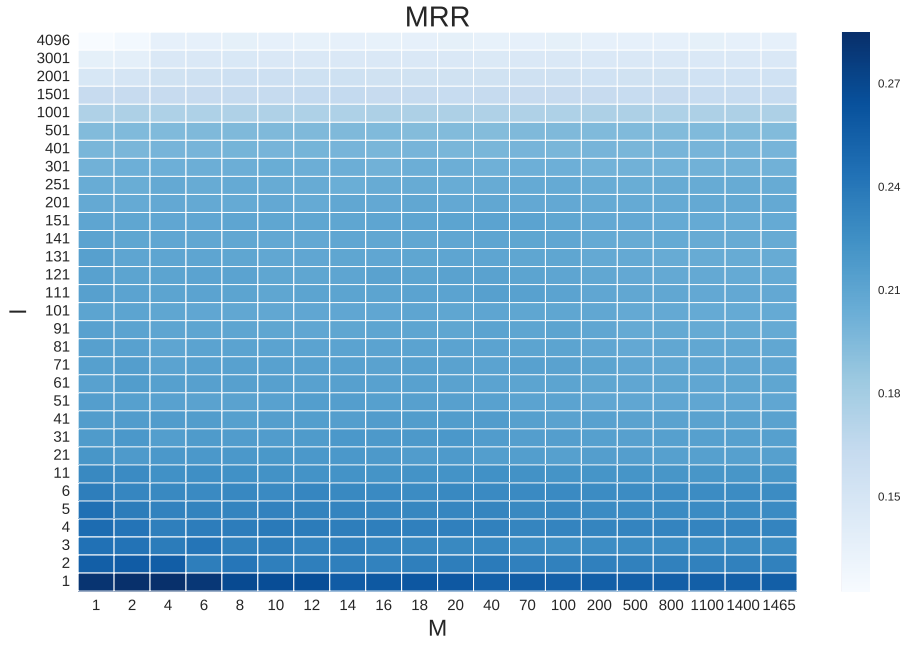


図 6.18: 次元削減の MRR への影響。x 軸が寄与率の高い順に成分を M 個使用したことを表し、y 軸が次元削減する前の時空間分布を作成するときに使用したユーザ l 人を表す。青色の濃さが MRR における値の大きさを表す。

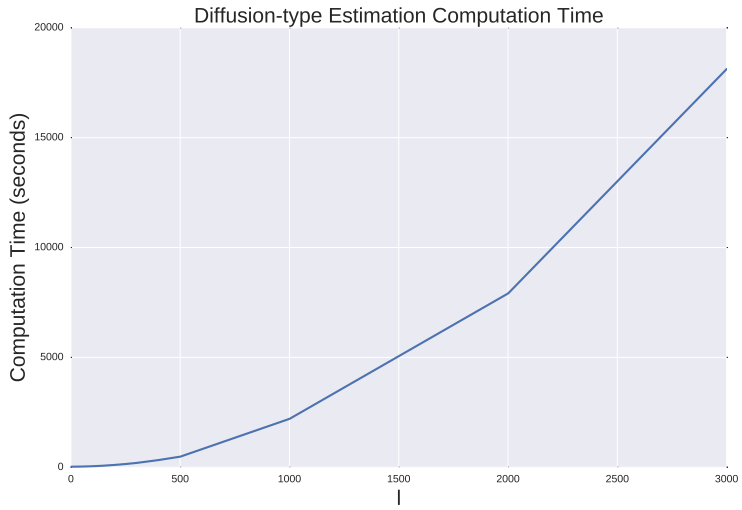


図 6.19: 次元削減した場合の時空間分布推定の計算時間。x 軸が時空間分布を推定するときに使用したユーザ l 人を表し、y 軸が計算時間 (秒) を表す

6.9.3　次元削減にともなう時空間分布の計算時間

時空間分布の推定において、DPMU では l 人の時空間分布を推定するので $O(l)$ である。一方、次元削減するモデルでは、l 人のユーザから共分散行列を求める必要があるため $O(l^2)$ と考えられる。そこで本節にて、実際に RDPMU での時空間分布の推定にかかる時間について議論する。図 6.19 に、l 人のユーザを用いて時空間分布を計算したときにかかった時間を示した。$l = 51$ の場合に約 34 秒、$l = 501$ の場合に約 8 分、$l = 1000$ の時に約 37 分、$l = 3001$ の時に約 5 時間の計算が必要となっており、予想した $O(l^2)$ に近い動きとなっていることが、図 6.19 から観察された。よって、RDPMU は、l が大きい場合に時空間分布の計算コストが非常に大きくなると言える。また、6.9.2 節で示したように、若干の log-likelihood、Rank、MRR の向上が期待できるが、向上する M は実験的にしか決定できない。計算コストの大きさと log-likelihood、Rank、MRR の向上量を考えると、DPMU の方が RDPMU に比べ優れているということが言える。

第7章　結論

本論文では、位置情報 SNS のスパース性の高いチェックインデータを用いて、全ユーザのチェックインデータおよび類似したユーザのチェックインデータから時空間分布を推定し、そこから定常的なパターン、周期的なパターン、突発的なパターンなどの時空間パターンの抽出を目的とした。更にこの結果を用いてユーザのチェックインデータの末尾時刻以降の位置を予測するモデル構築を目指した。

7.1　全ユーザの時空間パターンの抽出

第 3 章では、位置情報 SNS のチェックインデータの欠点である、チェックイン以外の時刻にユーザがどこにいたかが分からないという問題を、拡散方程式を用いてユーザの時空間分布を推定することで解決した。ユーザの時空間分布は、時間軸および空間軸を離散化することなく推定することができた。また、推定したユーザの時空間分布に、ICA を適用することで、ユーザの時空間パターンとして、定常的なパターンや、周期的なパターン、突発的なパターンを抽出できた。

7.2　スケールパラメータ λ の推定

拡散方程式には、ユーザの移動距離の二乗の時間変化の逆数の次元を持つスケールパラメータ λ がある。スケールパラメータ λ は全ユーザに共通であるような粗いモデルであるので、推定された分布は λ の値に敏感ではないが、それでも外れ値などで大きく変化すれば分布が変化してしまう。そこで、第 4 章にて、逆ガンマ分布を事前分布として利用することにより、外れ値などの影響を受けにくい λ の推定方法を提案した。

交差検定などの手法でスケールパラメータ λ を推定するのと比較して、この事前分布を用いた推定方法では λ を高速に推定することが可能であった。

7.3 類似ユーザの時空間パターンの抽出

時空間分布とヘリンジャー距離を用いることにより、ユーザ間の距離を第 5 章で定義した。先行研究では、チェックインの位置や時刻を離散化する必要があったが、第 5 章で定義したユーザ間の距離は、時空間分布から直接計算することができた。また、ヘリンジャー距離がユークリッド距離であるという性質から、凝縮型のクラスタリングで分類度が高いと言われている Ward 法を用いて、ユーザをクラスタリングすることができた。このクラスタを用いることで特徴のある時空間パターンを抽出することができた。

7.4 特定周期の時空間パターンを用いたユーザの位置予測

第 6 章では、第 5 章で定義した、ユーザの時空間分布と時空間分布から計算したユーザ間距離を用いて、ユーザの周期性をベースとした、ユーザのチェックインデータの末尾以降の位置を予測する DPM と DPMU を提案した。予測対象のユーザのチェックインデータだけでなく、類似したユーザのチェックインデータを用いて時空間分布を推定することで、log-likelihood、Rank、MRR などの指標による評価を先行研究よりも向上させることが出来た

PCA による時空間分布の次元削減を利用した予測手法 RDPMU では、寄与率の低いパターンのノイズを減らすことにより、log-likelihood、Rank、MRR の全てを若干向上できることが分かった。次元削減のために共分散行列を作成する必要があり計算量が増加するので、位置予測モデルとしては、次元削減をしない DPMU のほうが実用的であった。

7.5 今後の展望

ユーザの存在確率の時空間分布は、チェックインデータを時間帯、領域ごとにまとめるなどの離散化をすることなく推定することができた。しかし、PCA や ICA にて時空間パターンをユーザの時空間分布から抽出するため、時間軸を離散化して、共分散行列を計算する必要があった。そのため、離散化によりチェックインデータの情報が、一部失われてしまっていると考えられる。離散化は今回 1 時間区切りにしたが、1 日区切りにしたり、1 分区切りにすることで、抽出されるパターンが変わる可能性も考えられる。そこで、時間軸を離散化せずに、拡散方程式で推定した時空間パターンのままでパターン抽出することで、離散化の区切りによらず、またチェックインデータの持つ情報を失うことなく、時空間パターンを抽出する方法を適用したい。これは今後の課題である。

位置予測では、本論文で用いなかった、チェックイン時のツイート内容や、場所のカテゴリ、ユーザの SNS 上での影響度の強さなどの情報を使用することで、より高い精度の、位置の予測ができるようになると考えられる。このようなチェックインの付加情報を利用した手法も検討したい。

位置予測は、物理的制約が大きい施設推薦システムに適用することで、施設推薦の質を上げることが可能と考えられる。施設推薦システムは、推薦システムの一つであり、主にユーザがどの施設に行くべきかを推薦する手法であるが、ユーザのいる位置によって行ける範囲が限定されるため、物理的制約を大きく受ける手法である。一般に、推薦システムは、ユーザが好む商品や施設などを推定して、ユーザに推薦を行う手法であり、協調フィルタリング [33]、内容ベース推薦 [30] が、推薦システムとして広く使われている。また、Matrix Factorization [20][28]、LDA [3] なども、行列を分解し特徴量を抽出する手法であるが、推薦システムの一部としてよく利用されている手法である。これらの手法と第 6 章で提案した位置予測モデルを組み合わせることで、より良い場所を推薦することができるようになると考えられる。

位置予測は他にも、ユーザの現在地に応じた情報 (広告など) を出すサービスなどにも適用可能と考えられる。ユーザが目的地の入力することなく、目的地を情報を得ることができることは、位置予測の一つの利点である。数時間後に移動する場所の「お得な」情報

を前もって受け取ることができると、ユーザは移動時間中に、手に入れた情報を元に目的地での行動を計画することができるようになる。このように位置予測は、様々なアプリケーションへの応用が期待でき、その可能性は幅広い。今後も位置予測とその応用の研究を行っていきたい。

参考文献

[1] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

[2] Derya Birant and Alp Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] Huiping Cao, Nikos Mamoulis, and David W Cheung. Mining frequent spatio-temporal sequential patterns. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[5] Huiping Cao, Nikos Mamoulis, and David W Cheung. Discovery of periodic patterns in spatiotemporal sequences. *Knowledge and Data Engineering, IEEE Transactions on*, 19(4):453–467, 2007.

[6] Chen Cheng, Haiqin Yang, Irwin King, and Michael Lyu. Fused matrix factorization with geographical and social influence in location-based social networks, 2012.

[7] Zhiyuan Cheng, James Caverlee, Krishna Yeswanth Kamath, and Kyumin Lee. Toward traffic-driven location-based web search. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 805–814, New York, NY, USA, 2011. ACM.

[8] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z Sui. Exploring millions of footprints in location sharing services. *ICWSM*, 2011:81–88, 2011.

[9] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[10] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[11] Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(11):1057–1066, 2009.

[12] R.P. Feynman and A.R. Hibbs. *Quantum mechanics and path integrals*. International series in pure and applied physics. McGraw-Hill, 1965.

[13] M. Fire, D. Kagan, R. Puzis, L. Rokach, and Y. Elovici. Data mining opportunities in geosocial networks for improving road safety. In *Electrical Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of*, pages 1–4, 2012.

[14] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330–339. ACM, 2007.

[15] Bo Hu, Mohsin Jamali, and Martin Ester. Spatio-temporal topic modeling in mobile social media for location recommendation. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1073–1078. IEEE, 2013.

[16] Chi-Min Huang, J Jia-Chin Ying, and V Tseng. Mining users behavior and environment for semantic place prediction. In *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated task*, pages –, 2012.

[17] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[18] Juyoung Kang and Hwan-Seung Yong. Mining trajectory patterns by incorporating temporal properties. In *Proceedings of the 1st International Conference on Emerging Databases*, pages 63–68, 2009.

[19] Risi Kondor and Tony Jebara. A kernel between sets of vectors. In *ICML*, pages 361–368, 2003.

[20] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[21] Takeshi Kurashima, Tomoharu Iwata, Takahide Hoshide, Noriko Takaya, and Ko Fujimura. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 375–384. ACM, 2013.

[22] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, page 34. ACM, 2008.

[23] Defu Lian and Xing Xie. Collaborative activity recognition via check-in history. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pages 45–48. ACM, 2011.

[24] Hechen Liu and Markus Schneider. Similarity measurement of moving object trajectories. In *Proceedings of the Third ACM SIGSPATIAL International Workshop on GeoStreaming*, pages 19–22. ACM, 2012.

[25] K. S. Lomax. Business failures: Another example of the analysis of failure data. *Journal of the American Statistical Association*, 49(268):847–852, dec 1954.

[26] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[27] Nikos Mamoulis, Huiping Cao, George Kollios, Marios Hadjieleftheriou, Yufei Tao, and David W Cheung. Mining, indexing, and querying historical spatiotemporal data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 236–245. ACM, 2004.

[28] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[29] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. An empirical study of geographic user activity patterns in foursquare. *ICWSM*, 11:70–573, 2011.

[30] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[31] David Pollard. *A user's guide to measure theoretic probability*, volume 8. Cambridge University Press, 2002.

[32] Adam Rae, Vanessa Murdock, Adrian Popescu, and Hugues Bouchard. Mining the web for points of interest. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 711–720, New York, NY, USA, 2012. ACM.

[33] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[34] Frederic Paik Schoenberg, David R. Brillinger, and Peter Guttorp. *Point Processes, Spatial-Temporal*, pages –. John Wiley & Sons, Ltd, 2006.

[35] Ilias Tsoukatos and Dimitrios Gunopulos. *Efficient mining of spatiotemporal patterns*. Springer, 2001.

[36] Joe H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[37] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. Finding similar users using category-based location history. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 442–445. ACM, 2010.

[38] Gökhan Yavaş, Dimitrios Katsaros, Özgür Ulusoy, and Yannis Manolopoulos. A data mining approach for location prediction in mobile environments. *Data & Knowledge Engineering*, 54(2):121–146, 2005.

[39] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–613. ACM, 2013.

[40] Jia-Dong Zhang and Chi-Yin Chow. igslr: Personalized geo-social location recommendation: A kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, pages 334–343, New York, NY, USA, 2013. ACM.

[41] Yu Zheng and Xing Xie. Learning travel recommendations from user-generated gps traces. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):2, 2011.

謝辞

本論文は、筆者が東京大学大学院総合文化研究科山口和紀研究室で山口和紀教授のご指導のもとで、研究をまとめた論文である。研究の全般において学部時代から長期に渡って熱心なご指導ご鞭撻を頂いた、山口和紀教授に感謝いたします。お忙しい中、本論文の審査を引き受けて下さった山口泰教授、田中哲朗准教授、金子知適准教授、森畑明昌准教授には、審査以外だけでなく合同ゼミなどで多種多様なアドバイスを頂き、感謝いたします。また、山口和紀研究室の OB である松田源立さんには、本論文の根本となる推定モデルアルゴリズムの構築において、ご指導を頂き感謝いたします。最後に、本論文をまとめるまでに温かく見守ってくれていた家族に感謝の意を表して謝辞と致します。