【論文紹介】
人間社会的流行における数理モデルの提案
-イノベータ理論と時間遅れの方程式を用いて-

武藤 克弥 (Katsuya Mutoh)
u255018@st.pu-toyama.ac.jp

富山県立大学 大学院 電子・情報工学専攻 情報基盤工学部門

June 23, 2023

はじめに[1]

背景

- マーケティングにおいて，新商品やサービス開発のためにトレンド予測をすることが欠かせない
- 病気の流行を微分方程式で表した SIR モデルを拡張し，人間社会の流行をモデル化する研究が行われている

目的

- SIR モデルにイノベータ理論を組み込み，新しい流行モデルを提案
- Twitter や口コミを基に，あるトレンドの流行り廃りを提案した数理モデルで再現し，高精度なフィッティングを目指す

[1] 太田 靖, 水谷 直樹., 2021.

概要

SIR モデル

- 未感染者 (S)，感染者 (I)，免疫者 (R) に分類し，それぞれの増減を連立非線形微分方程式で表現
- α：感染率，β：治癒速度

$$
\begin{cases}
\dfrac{dS(t)}{dt} = -\alpha S(t)I(t) \\[2ex]
\dfrac{dI(t)}{dt} = \alpha S(t)I(t) - \beta I(t) \\[2ex]
\dfrac{dR(t)}{dt} = \beta I(t)
\end{cases}
$$

はじめに
概要
概要
概要
提案手法
数値実験
結果
まとめ
自分の研究に対して

イノベータ理論

- 流行を取り入れる早さは人それぞれ
- 流行には流れがある=時間的な遅れがある 現象をモデル化

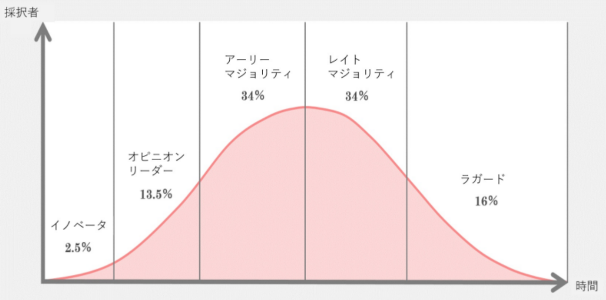


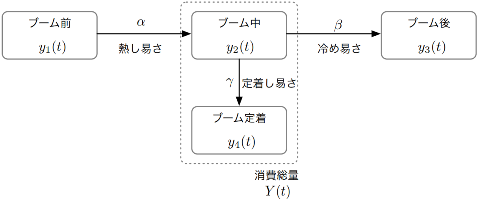
図 1: イノベータ理論による流行取入のタイミング

概要

従来モデルの問題点

- 同様に SIR モデルを拡張
- 単調な流行の爆発・衰退しか再現できなかった

$$
\begin{cases}
\dfrac{dy_1(t)}{dt} = -\alpha y_1(t) \\
\dfrac{dy_2(t)}{dt} = \alpha y_1(t) - (\beta + \gamma) y_2(t) \\
\dfrac{dy_3(t)}{dt} = \beta y_2(t) \\
\dfrac{dy_4(t)}{dt} = \gamma y_2(t)
\end{cases}
$$



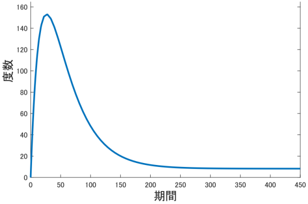
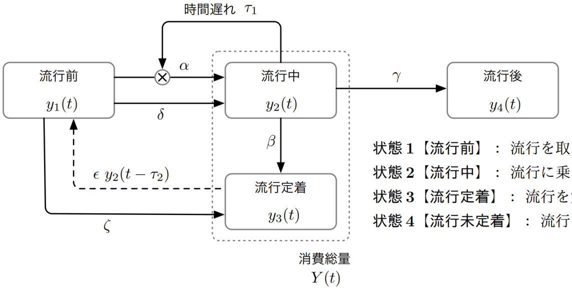


図 2: 従来モデル (一部抜粋)

提案手法(モデル)



状態 1【流行前】：流行を取り入れる可能性がある状態
状態 2【流行中】：流行に乗っている状態
状態 3【流行定着】：流行を定着させた状態
状態 4【流行未定着】：流行を取り止めた状態

$$
\begin{cases}
\dfrac{dy_1(t)}{dt} = -\alpha y_1(t) y_2(t-\tau_1) - \delta y_1(t) + \varepsilon y_2(t-\tau_2) - \zeta \\
\dfrac{dy_2(t)}{dt} = \alpha y_1(t) y_2(t-\tau_1) - (\beta+\gamma) y_2(t) + \delta y_1(t) \\
\dfrac{dy_3(t)}{dt} = \beta y_2(t) - \varepsilon y_2(t-\tau_2) + \zeta \\
\dfrac{dy_4(t)}{dt} = \gamma y_2(t)
\end{cases}
$$

解 $y_2(t) + y_3(t)$ を実データにフィッティング

α：接触して流行前→流行中になる確率
β：流行中→流行定着になる確率
γ：流行中→流行語 (未定着) になる確率

δ：接触せず流行前→流行中になる確率 (新規生産)
ε：流行定着→流行前になる確率 (インフルエンサーによる再起)
ζ：流行前→流行強制定着の確率 (商品の無料取得など)

図 3: 提案モデル (一部改変)

提案手法

解の導出とパラメータ推定

- モデル式は解析解を必ずしも得られない
- ベイズ推定によって数値解からパラメータ推定

ベイズ推定

- データごとにパラメータがばらつく=確率分布に従っている
  としてパラメータの分布を推定する手法
- 事前分布の設定，尤度関数の導出を行い，事後分布 (パラメータが従うべき分布) を求める

提案手法

ベイズ推定 (事後分布の式)

$f(\theta|\boldsymbol{Y})$：事後分布, $f(\boldsymbol{Y}|\theta)$：尤度関数, $f(\theta)$：事前分布
$f(\boldsymbol{Y})$：観測データ Y が与えられたときの正規化定数

$$f(\theta|\boldsymbol{Y}) = \frac{f(\boldsymbol{Y}|\theta)f(\theta)}{f(\boldsymbol{Y})} \tag{1}$$

本研究への適用

事前分布=一様分布： $f(\theta) = U_{[-\theta_0,\theta_0]}$ (θ_0 は大きい正の定数)

・m：観測点の数
・$\boldsymbol{Y} = (Y(t_1), Y(t_2), \cdots, Y(t_m))$：実データベクトル
・$\boldsymbol{F^\theta} = (y_2^\theta(t_1) + y_3^\theta(t_1), y_2^\theta(t_2) + y_3^\theta(t_2), \cdots, y_2^\theta(t_m) + y_3^\theta(t_m))$
→パラメータ θ のときの数値解

・観測誤差：$\varepsilon = \boldsymbol{Y} - \boldsymbol{F^\theta}$ $\quad (\varepsilon \sim N(0, \Sigma_\theta^2))$
・尤度関数：$f(\boldsymbol{Y}|\theta) = \exp\left\{-\frac{(\boldsymbol{Y}-\boldsymbol{F^\theta})(\boldsymbol{Y}-\boldsymbol{F^\theta})^T}{2\Sigma_\varepsilon^2}\right\}$
・事後分布：$f(\theta|\boldsymbol{Y}) \propto \exp\left\{-\frac{(\boldsymbol{Y}-\boldsymbol{F^\theta})(\boldsymbol{Y}-\boldsymbol{F^\theta})^T}{2\Sigma_\varepsilon^2}\right\}$

提案手法 パラメータ (事後分布) の更新

マルコフ連鎖モンテカルロ法 (MCMC)

- 疑似乱数シミュレーションと 1 つ前の状態を利用する更新方法
- 尤度関数の最大化パラメータを偏微分で解析的に解けないときに使用

メトロポリス-ヘイスティングズ法 (M-H 法)

- MCMC の一種
- 提案分布を設定．事後分布に等しくなるようパラメータサンプリング

提案分布=正規分布 (中心 θ_k，標準偏差 σ_θ)

1. $N(\theta_k, \sigma_\theta^2)$ から θ' を抽出
2. θ' から微分方程式の数値解 $\boldsymbol{F}^{\boldsymbol{\theta}'}$ を取得 (MATLAB dde23 使用)
3. **事後分布 $f(\theta'|\boldsymbol{Y})$ を導出**
4. 事後分布比率からパラメータ採択確率を計算
$$\alpha(\theta', \theta_k) = min\left(1, \frac{f(\theta'|\boldsymbol{Y})}{f(\theta_k|\boldsymbol{Y})}\right) \tag{2}$$
5. (乱数値) $< \alpha(\theta', \theta_k)$ のとき $\theta_{k+1} = \theta'$(更新),
False なら $\theta_{k+1} = \theta_k$(未更新)

数値実験の流れ パラメータ推定とフィッティング

1. ある話題のツイートデータを用意
2. パラメータ 5 種 α〜ε(6p) の初期値を決定
3. 実データから ζ, τ_1, τ_2 を決定
4. M-H 法を 500,000 回実行し，α〜ε を決定
5. 実データとのフィッティング．決定係数の計算

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(E_i - \bar{E})^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \quad (E_i = Y_i - F_i^{\theta}) \tag{3}$$

6. ζ, τ_1, τ_2 の微調整
7. 4〜6 を繰り返し，最終的なパラメータ決定

数値実験 1 先行研究との比較

データ

- 原発に関するツイート (日数ごとの) 件数
- 先行研究 1,2 モデルとの当てはまりを比較

原発(提案モデル)			
α	0.00132	$y_1(0)$	29000
β	0.00541	$y_2(0)$	0
γ	0.02723	$y_3(0)$	0
δ	0.05132	$y_4(0)$	0
ε	0.01183	τ_1	45
ζ	0.35	τ_2	150

原発 (中桐らのモデル)			
α	0.05340	$y_1(0)$	29000
β	0.00808	$y_2(0)$	0
γ	0.02520	$y_3(0)$	0
		$y_4(0)$	0

原発 (植田らのモデル)			
α	0.04680	$y_1(0)$	29000
β	0.00576	$y_2(0)$	0
γ	0.02251	$y_3(0)$	0
δ	-0.00066	$y_4(0)$	0

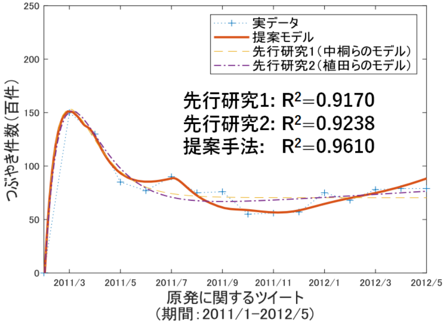


図 4: 先行研究との比較 (一部改変)

データ

- 一過性 (単調な) 流行，再起性 (波のある) 流行データで評価
- 一過性=Pokémon GO 口コミ件数
  - 2016/6〜2017/4 Google トレンドの口コミ数
  - 微分方程式の解の単位時間=1 日
  - 初期値 $\tau_1 = 16$ (約 2 週間間前に海外の先行配信を考慮)
- 再起性=本麒麟ツイート件数
  - 2019/2/23〜2019/3/12 新品キャンペーンの Twitter データ ()
  - 微分方程式の解の単位時間=1 時間
  - 初期値 $\tau_1 = 13$ (「イベント半日前から話題浮上」と仮定)
- 初期値 τ_2=(再起の山の変化点からピークまでの時間)

結果1　一過性流行 (Pokémon GO)

解 $y_2(t) + y_3(t)$ を実データにフィッティング

Pokemon Go			
α	0.00369	$y_1(0)$	550
β	0.00348	$y_2(0)$	0
γ	0.04591	$y_3(0)$	0
δ	0.01259	$y_4(0)$	0
ε	0.00201	τ_1	16
ζ	0	τ_2	155

$$R^2 = 0.9979$$

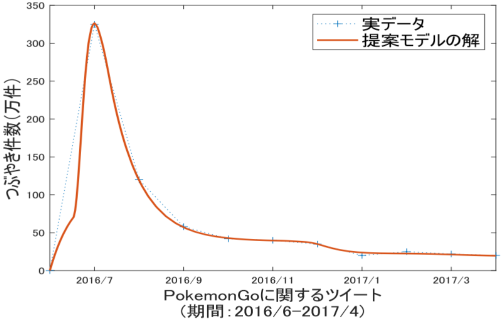


図 5: 決定パラメータとフィッティング結果

結果２　再起性流行 (本麒麟)

- 解 $y_2(t) + y_3(t)$ を実データにフィッティング
- 再起の波も表現
- 若干の過剰な振動が見られた

本麒麟			
α	0.00017	y₁(0)	3500
β	-0.00034	y₂(0)	0
γ	0.03170	y₃(0)	199
δ	0.03593	y₄(0)	0
ε	0.01962	τ₁	12
ζ	7	τ₂	144

$$R^2 = 0.9643$$

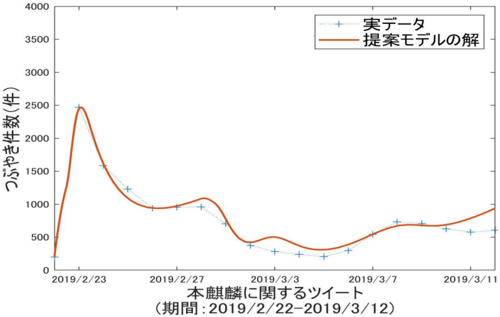


図 6: 決定パラメータとフィッティング結果 (一部改変)

考察

- 一過性 (Pokémon GO，君の名は。)，再起性 (本麒麟，KitKat)
  いずれもパラメータ γ が一番高い傾向
  →いずれも初めの爆発的な波の後に止める人が多いことを表現
- ε について再起性の方が高い
  →インフルエンサーが再度流行 (波) を起こした結果の現れ

まとめ

- 病気の流行を表すSIRモデルとマーケティングのイノベータ理論を組み合わせ，社会的流行を表現する数理モデルを設計
- 先行研究と比較して再起する流行 (微細振動) を表現可能にした

今後の課題

- 一過性，再起性以外のデータ，国外の流行データでの検証
- マーケティングの知見をさらに取り入れて，τ_1，τ_2，ζ の決め方を改善

自分の研究への応用

やりたいと思っていること

- トピック (キーワード等) に対する話題の流行変化をモデリング
  → 1 つではなく複数 (キーワード A → B への流行変化)
- キーワード A を発端として共起されるキーワードのうち，どれが流行しそうかを予測？

本紹介論文が活かせそうな部分

- 6 ページの微分方程式
  →微分方程式 (1 キーワード ver) を複数キーワード ver に拡張
- 微分方程式の中に共起ネットワークを入れて，キーワード遷移を表現？

やってみたこと

微分方程式数値解の導出

- 論文と同じ MATLAB dde23(遅延微分方程式) ライブラリ使用
- 最終パラメータを代入して実行

sir_rogers.m

```
function dydt = sir_rogers(t,y,Z,p)
ylag1 = Z(:,1);
ylag2 = Z(:,2);
Y2tau_1 = ylag1(2);
Y2tau_2 = ylag2(2);
Y1 = y(1);
Y2 = y(2);
Y3 = y(3);
Y4 = y(4);

dY1dt = - p.alpha * Y1 *Y2tau_1 - p.delta * Y1 ...
        + p.epsilon * Y2tau_2 - p.zeta;

dY2dt = p.alpha * Y1 * Y2tau_1 ...
        - (p.beta + p.gammma) * Y2 + p.delta * Y1;

dY3dt = p.beta * Y2 - p.epsilon * Y2tau_2 + p.zeta;

dY4dt = p.gammma * Y2;

dydt = [dY1dt; dY2dt; dY3dt; dY4dt];
end
```

sir_rogers_sol_yourname.m

```
p.alpha = 0.00369;
p.beta = 0.00348;
p.gammma = 0.04591;
p.delta = 0.01259;
p.epsilon = 0.00201;
p.zeta = 0;

tau = [16, 155];
Y1_0 = 550;
Y2_0 = 0;
Y3_0 = 0;
Y4_0= 0;
history = [Y1_0, Y2_0, Y3_0, Y4_0];

tspan = [0,365];
sol = dde23(@(t,y,Z) sir_rogers(t,y,Z,p), tau, history, tspan);

plot(sol.x,sol.y(2,:)+sol.y(3,:))
xlabel('Time t')
ylabel('y(2)+y(3)')
```

図 7: 微分方程式のコード

やってみたこと

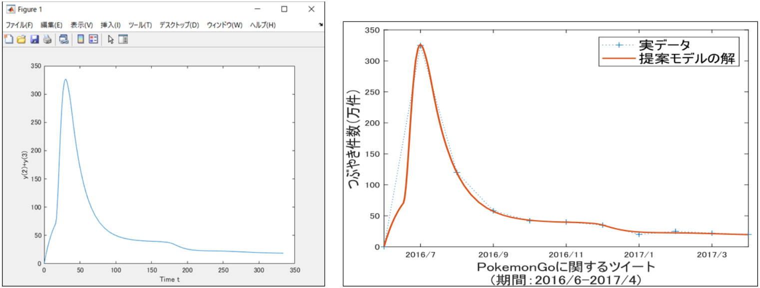


図 8: Pokémon GO の数値解

- dde23 →微分方程式の構築+解の導出に使えそう
- M-H 法→ PyMC3(Python ライブラリ) で実装可能
- MATLAB → Python から動かせる
- 論文の再現可能？