研究進捗

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

November 22, 2023

はじめに

著者への質問

研究に用いようとしている意思決定モデルの再現が難しいため，著者への質問資料を作成した.

3p-23p が質問内容 (進捗と発生している問題)

はじめに

参考にさせていただいた論文

主に [1] の再現を中心に行い，[2]〜[4] は適宜離散化やアルゴリズム実装時に参考させていただきました.

[1] 長江剛志, 赤松隆：連鎖的な意思決定構造を持つプロジェ クトの動学的評価法：オプション・グラフ・モデルとその 解法, 土木学会論文集, No. 772/IV-65, pp. 185–202, 2004

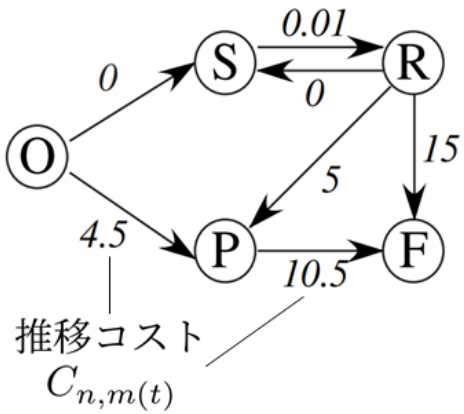
[2] 赤松隆・長江剛志：社会基盤整備・運用事業の経済リスク管理問題に対するファイナンス工学的アプローチ, 土木計画学研究・論文集, No. 23, pp. 1-21, 2006

[3] Akamatsu, T. and Nagae, T.: A network of options: Evaluating complex interdependent decisions under uncertainty, Journal of Economic Dynamics and Control, Vol. 35, pp. 714-729, 2011.

[4] Nagae, T. and Akamatsu, T.: Dynamic Revenue Management of Toll Road Projects under Transportation Demand Uncertainty, Networks and Spatial Economics, Volume 6, Numbers 3-4, pp. 345-357, 2006.

現在行っていること

[1] 8 章の例題の再現を (12p アルゴリズムを用いて) 行っております



アクティビティ: $n \in \{O, S, R, P, F\}$ $V_n(t, P)$

(1) 利潤: $\pi_n(t, P)$, (2) 満期利潤, (3) 推移コスト: $C_{n,m(t)}$

O: 事前評価 $\pi_{\mathrm{O}}(t, P) = -M_{\mathrm{O}}$

S: 計画凍結 $\pi_{\mathrm{S}}(t, P) = 0$

R: 再評価 $\pi_{\mathrm{R}}(t, P) = -M_{\mathrm{R}}$

P: 1区間供用 $\pi_{\mathrm{P}}(t, P) = X_{\mathrm{P}}P - E_{\mathrm{P}}$

F: 2区間供用 $\pi_{\mathrm{F}}(t, P) = X_{\mathrm{F}}P - E_{\mathrm{F}}$

$T = 20,\ r = 5\%,\ \alpha = 1\%,\ \sigma = 40\%$

$M_{\mathrm{O}} = 0.02,\ M_{\mathrm{R}} = 0.01$

$X_{\mathrm{P}} = 0.5,\ E_{\mathrm{P}} = 0.6,\ X_{\mathrm{F}} = 1,\ E_{\mathrm{F}} = 1$

離散化

時間 t と状態 P の離散化

・以下の設定
・「P がどの i においても P_0 から $P_0 + J\Delta P$ まで変化する」と仮定

$$(t^i, P^j) = (i\Delta T, P_{min} + j\Delta P)$$
$$(i = 0, 1, \cdots, I),\ (j = 0, 1, \cdots, J)$$
$$I = J = 400$$

時間: $t \in [0, T] \quad (T = 20)$
状態空間: $[P_{\min}, P_{\max}] \in \mathcal{R}$
$P_{\min} = P_0 = 0$
$P_{\max} = P^J = 8$

$$\Delta T = \frac{T - 0}{400} = \frac{1}{20}$$
$$\Delta P = \frac{P^J - P_0}{400} = \frac{1}{50}$$

懸念 1.

$\Delta T,\ \Delta P$ の設定について

$$P^j = P_0 + j\Delta P$$

$=$

P^j	$j = 0$	$j = 1$	$j = 2$	$\cdots$	$j = J$
$i = 0$ $\vdots$ $i = I$	P_0	$P_0 + \Delta P$	$P_0 + 2\Delta P$	$\cdots$	$P_0 + J\Delta P$

離散化

懸念 2.

P^j 離散化の設定について

$$P^j = P_0 + j\Delta P$$

$$= \begin{array}{|c|c|c|c|c|c|}\hline P^j & j=0 & j=1 & j=2 & \cdots & j=J \\ \hline i=0 \\ \vdots \\ i=I & P_0 & P_0+\Delta P & P_0+2\Delta P & \cdots & P_0+J\Delta P \\ \hline \end{array} \quad \text{---(※)}$$

$P(t)$ は幾何ブラウン運動に従う ([1], p15)

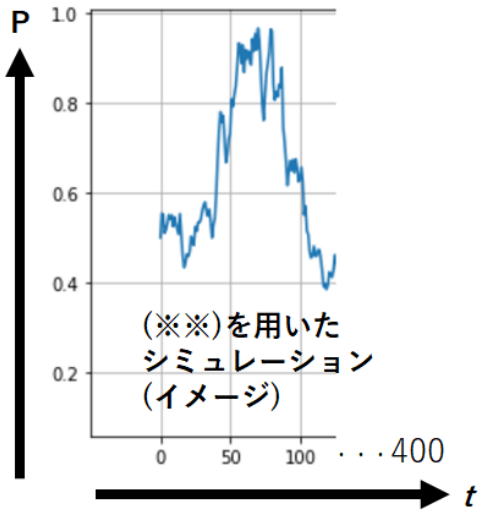
$$\boxed{\mathrm{d}P(t)/P(t) = \alpha\,\mathrm{d}t + \sigma\,\mathrm{d}Z(t)} \quad \text{---(※※)}$$

懸念 2.

(※) のように変化する場合,
P^j は幾何ブラウン運動 (※※) に従わないと形になると思われる

2.1. ドリフト・拡散項を使用する意味の記述であるのか？

$$\alpha(t,P,n) = \alpha P$$
$$\sigma(t,P,n) = \sigma P \quad [2]$$



2.2. ※※をプログラムでシミュレートして得られる「$[0,8]$ で変化する $P(t)$」を用いることを意味した記述か？

離散化

- 2.1. の想定で $\alpha(t,P)$, $\sigma(t,P)$, $\pi_n^i(t,P)$ を離散化

$$\alpha(t,P) = \alpha P$$

$$\simeq \begin{array}{|c||c|c|c|c|c|}\hline \alpha P^j & j=0 & j=1 & j=2 & \cdots & j=J \\ \hline\hline \begin{array}{c} i=0 \\ \vdots \\ i=I \end{array} & \alpha P_0 & \alpha(P_0+\Delta P) & \alpha(P_0+2\Delta P) & \cdots & \alpha(P_0+J\Delta P) \\ \hline \end{array}$$

$$\sigma(t,P) = \sigma P \simeq \sigma P^j \ \text{(同様)}$$

$$\pi_F(t,P) = P - 1$$

$$\simeq \boldsymbol{\pi}_F^i = \pi(t^i, P^j) = P^j - \mathbf{1} = \begin{bmatrix} P_0 - 1 \\ P_0 + \Delta P - 1 \\ P_0 + 2\Delta P - 1 \\ \vdots \\ P_0 + J\Delta P - 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} (J\ \text{次元}) \\ \\ \\ \\ \\ \end{matrix}$$

$$\boldsymbol{\pi}_P^i = 0.5P^j - 0.6 \cdot \mathbf{1} \qquad \boldsymbol{\pi}_S^i = \mathbf{0}$$

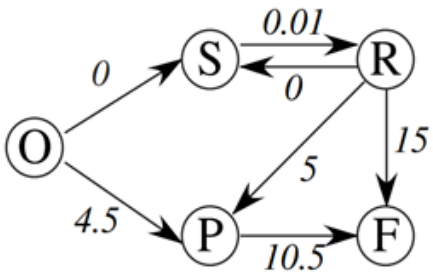
$$\boldsymbol{\pi}_R^i = -0.01 \cdot \mathbf{1} \qquad \boldsymbol{\pi}_O^i = -0.02 \cdot \mathbf{1}$$

アルゴリズムの再現

[1]12p のアルゴリズムを再現し，"O"，"S"，"R"，"P"，"F" の価値を導出

[Alg-Option Graph]

(終端条件)
for all $n \in N$ **do**
$V_n^{\mathrm{I}} := \max.\left[F_n,\ \max_{m \in O(n)} \left\{ V_m^{\mathrm{I}} - \mathbf{1} C_{n,m} \right\} \right];$
end for
(時刻についての逆向き帰納法)
for $i := \mathrm{I} - 1$ to 0 step -1 **do**
(初期化)
for all $n' \in N^{\mathrm{E}}$ **do**
差分方程式 (28) を解いて $V_{n'}^{i}$ を求める;
end for
$A(n) := |O(n)|$ **for all** $n \in N$;
$B(u) := |O(u)|$ **for all** $c \in C$;
$\hat{N} := N^{\mathrm{E}}$



(グラフ構造ごとに分解された問題の求解)
while $\hat{N} \neq \emptyset$ **do**
$\hat{N}$ の先頭要素を n とし，n を $\hat{N}$ から取り除く;
for all $m \in I(n)$ **do**
$A(m) := A(m) - 1$
if $A(m) = 0$ **then**
[Alg-Merit] を用いて問題 [LCP^i-m] を解き，V_m^i を求める;
m を $\hat{N}$ の末尾に挿入する;
end if
end for
for all $c \in I_C(n)$ **do**
$B(c) := B(c) - 1$
if $B(c) = 0$ **then**
[Alg-Merit] を用いて問題 [NCP^i-c] を解き，V_c^i を求める;
N_c 内の全要素を $\hat{N}$ の末尾に挿入する;
end if
end for
end while
end for

アルゴリズムの再現

終端条件について

懸念 3.

終端条件 $\boldsymbol{V}_F^I, \boldsymbol{V}_P^I, \boldsymbol{V}_R^I, \boldsymbol{V}_S^I, \boldsymbol{V}_O^I$ は全て 0 であるか？
(終端ペイオフは，$F_n = 0$ と仮定 ([3] 7 章))

(終端条件)
for all $n \in N$ **do**

$$\boldsymbol{V}_n^{\mathrm{I}} := \max.\left[\boldsymbol{F}_n,\ \max_{m \in O(n)}.\left\{\boldsymbol{V}_m^{\mathrm{I}} - \mathbf{1}C_{n,m}\right\}\right];$$

end for

---(※※※)

3.1. [2](p12) では $\boldsymbol{V}_n^i = \mathbf{0}$

3.2. [1],[3] では $\boldsymbol{V}_n^i$ の明記がみられなかったが，(※※※) を用いて求める形か？
→計算済み $m \in O(n)$ の価値を用いるため，
相互に価値未決定のサイクル構造 $(\boldsymbol{V}_R^I, \boldsymbol{V}_S^I)$ の計算不可？

現状 $\boldsymbol{V}_n^I = \mathbf{0}$ として実装

終端アクティビティ"F"の価値導出

[1] に従い，偏微分方程式を離散化

価値 V を求める式 (終端アクティビティの場合)

$$\frac{\partial V(t,P)}{\partial t}+\alpha(t,P)\frac{\partial V(t,P)}{\partial P}+\frac{1}{2}\{\sigma(t,P)\}^2\frac{\partial^2 V(t,P)}{\partial P^2}-rV(t,P)+\pi(t,P)=0$$

差分化 (Crank-Nicolson法) [4]

$$\begin{bmatrix} b_1^i & c_1^i & 0 & 0 & \cdots & 0 \\ a_2^i & b_2^i & c_2^i & 0 & \cdots & 0 \\ 0 & a_3^i & b_3^i & c_3^i & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{J-3}^i & b_{J-3}^i & c_{J-3}^i & 0 \\ 0 & \cdots & 0 & a_{J-2}^i & b_{J-2}^i & c_{J-2}^i \\ 0 & \cdots & 0 & 0 & a_{J-1}^i & b_{J-1}^i \end{bmatrix} \begin{bmatrix} V_1^i \\ V_2^i \\ V_3^i \\ \vdots \\ V_{J-3}^i \\ V_{J-2}^i \\ V_{J-1}^i \end{bmatrix} + \begin{bmatrix} d_1^i & c_1^i & 0 & 0 & \cdots & 0 \\ a_2^i & d_2^i & c_2^i & 0 & \cdots & 0 \\ 0 & a_3^i & d_3^i & c_3^i & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{J-3}^i & d_{J-3}^i & c_{J-3}^i & 0 \\ 0 & \cdots & 0 & a_{J-2}^i & d_{J-2}^i & c_{J-2}^i \\ 0 & \cdots & 0 & 0 & a_{J-1}^i & d_{J-1}^i \end{bmatrix} \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ V_3^{i+1} \\ \vdots \\ V_{J-3}^{i+1} \\ V_{J-2}^{i+1} \\ V_{J-1}^{i+1} \end{bmatrix} + \begin{bmatrix} \pi_1^i \\ \pi_2^i \\ \pi_3^i \\ \vdots \\ \pi_{J-3}^i \\ \pi_{J-2}^i \\ \pi_{J-1}^i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$\boldsymbol{L}_n^i$ ($J\times J$ 次元)　$\boldsymbol{V}_n^i$ (J 次元)　$\boldsymbol{M}_n^i$ ($J\times J$ 次元)　$\boldsymbol{V}_n^{i+1}$　(J 次元)

行列内の要素 [4]

$$\begin{aligned} a_j^i &= -\frac{\alpha_j^i}{4\Delta P}+\frac{(\sigma_j^i)^2}{4(\Delta P)^2} \\ b_j^i &= -\frac{1}{\Delta T}-\frac{(\sigma_j^i)^2}{2(\Delta P)^2}-r \\ c_j^i &= \frac{\alpha_j^i}{4\Delta P}+\frac{(\sigma_j^i)^2}{4(\Delta P)^2} \\ d_j^i &= \frac{1}{\Delta T}-\frac{(\sigma_j^i)^2}{2(\Delta P)^2} \end{aligned} \qquad \begin{aligned} &(\alpha_j^i=\alpha P^j) \\ &(\sigma_j^i=\sigma P^j) \end{aligned}$$

$$\boldsymbol{L}^i \begin{bmatrix} V_1^i \\ V_2^i \\ V_3^i \\ \vdots \\ V_{J-3}^i \\ V_{J-2}^i \\ V_{J-1}^i \end{bmatrix} + \boldsymbol{M}^i \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ V_3^{i+1} \\ \vdots \\ V_{J-3}^{i+1} \\ V_{J-2}^{i+1} \\ V_{J-1}^{i+1} \end{bmatrix} + \boldsymbol{\pi}^i = \boldsymbol{0}$$

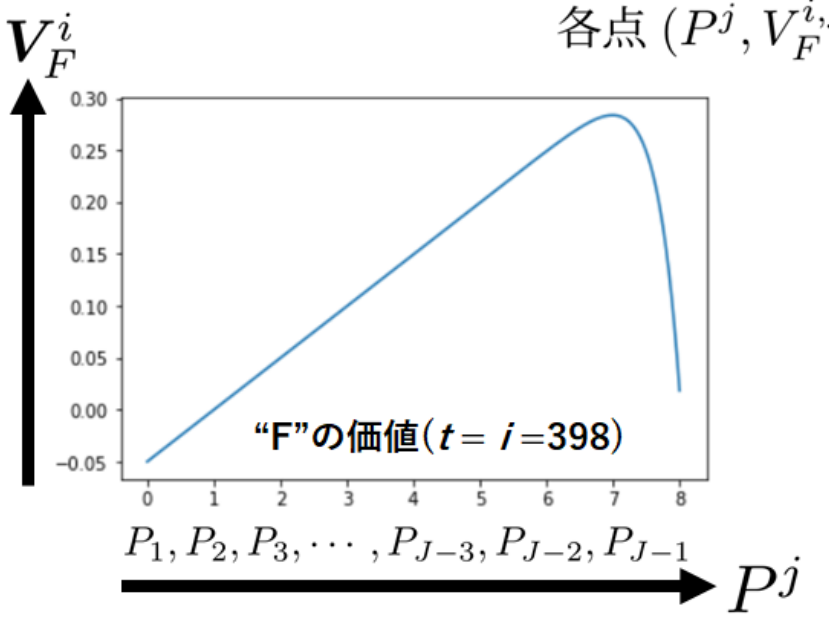
$i=I\to i=0$:
V_n^i 逐次導出

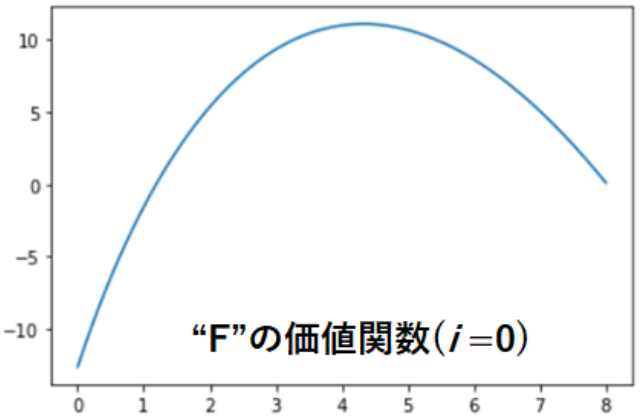
V_F^i の結果確認

先に全時刻で V_F^i を出力

```
for i in range(I-1):
    # 終端アクティビティFの価値計算
    i_rev = I-1-i
    act_now["F"][i_rev-1,:] = ( -np.linalg.inv(L[i_rev-1])
        @ (M[i_rev-1] @ act_now["F"][i_rev,:].T + act_pi["F"][i_rev-1,:].T) ).T
```

各点 $(P^j, V_F^{i,j})$ をプロット





懸念 4. [1] 各図は exp 型の増加だが，結果は増加→減少型の曲線となったこと

Merit 関数による $X_n^i(V_n^i)$ の更新

【解釈】最適 X_c^i (X_n^i) を求めることで，(※) から V_n^i が求められる

求めたいもの ([LCP^i-n] も同様)

[NCP^i-c] Find X_c^i such that

$$X_c^i \cdot G_c^i(X_c^i) = 0, \quad \text{and} \quad X_c^i \geq \mathbf{0},\ G_c^i(X_c^i) \geq \mathbf{0}.$$

[Alg-Merit]

Step 0 初期可能解 $X_c^{i(1)} \in \mathcal{R}_+$，$k := 1$.

Step 1 降下方向ベクトルの決定.

$$d^{(k)} := H(X_c^{i(k)}). \tag{34}$$

Step 2 ステップ・サイズ α を，以下の一次元探索問題の解として求める.

$$\alpha^* = \arg.\min_{\alpha\in[0,1]} \Phi\left(X_c^{i(k)} + \alpha d^{(k)}\right). \tag{35}$$

Step 3 解の改訂. $X_c^{i(k+1)} := X_c^{i(k)} + \alpha^* d^{(k)}$

Step 4 収束判定：収束していれば停止，そうでなければ $k := k+1$ として Step 1 へ.

解釈

$G(X_c^i)$ を求める処理

$$L_n^i V_n^{i(k)} = -X_n^{i(k)} - M_n^i V_n^{i+1} - \pi_n^i \quad \text{----------(※)}$$

$$G_n^{i(k)} := \min.\left[\min_{m\in O_c(n)}\left\{V_n^{i(k)} - V_m^{i(k)} - \mathbf{1}C_{n,m}\right\},\right.$$

$$\left.\min_{m'\in \hat{O}_c(n)}\left\{V_n^{i(k)} - V_{m'}^{i} - \mathbf{1}C_{n,m'}\right\}\right],\ \forall n \in N_c.$$

Step1: $H(X_c^i)$ を求める処理

$$H(X_c^i) \equiv \left[X_c^i - G(X_c^i)\right]_+ - X_c^i$$

Step2: $\Phi(X_c^i)$ を求める処理

$$\Phi(X_c^i) \equiv -G(X_c^i)\cdot H(X_c^i) - \frac{1}{2}H(X_c^i)\cdot H(X_c^i)$$

Merit 関数の実装

$\boldsymbol{H(X_n^i)}$, $\boldsymbol{G(X_n^i)}$, $\boldsymbol{\Phi(X_n^i)}$ を呼び出し関数として実装

・$\boldsymbol{H(X_n^i)}$ を求める処理

$$\boldsymbol{H}(\boldsymbol{X}_n^i) = \left[\boldsymbol{X}_n^i - \boldsymbol{G}(\boldsymbol{X}_n^i)\right]_+ - \boldsymbol{X}_n^{i(k)}$$

→【解釈】要素 $X^{i,j}$ は正または 0

・$\boldsymbol{G(X_n^i)}$ を求める処理

$$\boldsymbol{L}_n^i \boldsymbol{V}_n^{i(k)} = -\boldsymbol{X}_n^{i(k)} - \boldsymbol{M}_n^i \boldsymbol{V}_n^{i+1} - \boldsymbol{\pi}_n^i$$

$$\boldsymbol{G}_n^{i(k)} = \min\left[\min_{m\in O_c(n)}\left\{\boldsymbol{V}_n^{i(k)} - \boldsymbol{V}_m^{i(k)} - \mathbf{1}C_{n,m}\right\}, \underline{\min_{m'\in \hat{O}_c(n)}\left\{\boldsymbol{V}_n^{i(k)} - \boldsymbol{V}_{m'}^{i(k)} - \mathbf{1}C_{n,m'}\right\}}\right]$$

サイクル構造の場合

懸念 5. ベクトルの最小比較について

J 次元の各要素どうしで最小比較されると仮定

例) アクティビティ"O"の場合, $\min\left[\boldsymbol{V}_O^{i(k)} - \boldsymbol{V}_S^{i(k)} - \mathbf{1}C_{O,S},\quad \boldsymbol{V}_O^{i(k)} - \boldsymbol{V}_P^{i(k)} - \mathbf{1}C_{O,P}\right]$

$$\boldsymbol{G}_O^{i(k)} = \begin{array}{ll} \boldsymbol{V}_O^{i(k),0} - \boldsymbol{V}_S^{i(k),0} - C_{O,S} & j=0 \\ \boldsymbol{V}_O^{i(k),1} - \boldsymbol{V}_P^{i(k),1} - C_{O,P} & j=1 \\ \boldsymbol{V}_O^{i(k),2} - \boldsymbol{V}_P^{i(k),2} - C_{O,P} & j=2 \\ \vdots & \vdots \\ \boldsymbol{V}_O^{i(k),J} - \boldsymbol{V}_S^{i(k),J} - C_{O,S} & j=J \end{array}$$

"S"と"P"の要素が入り混じると想定

Merit 関数の実装

[Alg-Merit] を以下のように解釈して実装

・$\Phi(\boldsymbol{X}_n^i)$ を求める処理

$$\Phi(\boldsymbol{X}_n^i) = -\boldsymbol{G}(\boldsymbol{X}_n^i)\cdot\boldsymbol{G}(\boldsymbol{X}_n^i)^\top - \frac{1}{2}\boldsymbol{H}(\boldsymbol{X}_n^i)\cdot\boldsymbol{H}(\boldsymbol{X}_n^i)^\top$$

（内積）　（内積）

Step0: 初期状態 $\boldsymbol{X}_n^{i(1)} = \boldsymbol{0}$ (J 次元ベクトル)

Step1: $\boldsymbol{d}^{(k)} = \boldsymbol{H}(\boldsymbol{X}_n^i)$

Step2: $\alpha \in [0,1] \quad \rightarrow \quad \alpha = \frac{l}{1000},\ (l = 0, 1, \cdots, 1000)$

for l (0 to 1000): $\boldsymbol{G}(\boldsymbol{X}_n^{i(k)} + \frac{l}{1000}\boldsymbol{d}^{(k)}) \rightarrow \boldsymbol{H}(\boldsymbol{X}_n^{i(k)} + \frac{l}{1000}\boldsymbol{d}^{(k)})$
$\rightarrow \Phi(\boldsymbol{X}_n^{i(k)} + \frac{l}{1000}\boldsymbol{d}^{(k)})$ 導出 $\rightarrow$ リストに格納

リスト中で $\Phi(\boldsymbol{X}_n^{i(k)} + \frac{l}{1000}\boldsymbol{d}^{(k)})$ が最小のインデックス l^* を求める

Step3: $\boldsymbol{X}_n^{i(k+1)} = \boldsymbol{X}_n^{i(k)} + \frac{l^*}{1000}\boldsymbol{d}^{(k)}$

Step4: $\boldsymbol{X}_n^{i(k+1)} = \boldsymbol{X}_n^{i(k)}$ なら停止,
それ以外は $k = k+1$

懸念 6. 収束判定について

$\boldsymbol{X}_n^{i(k+1)}$ と $\boldsymbol{X}_n^{i(k)}$,
J 個の要素が全て一致する必要があるのか？

Merit 関数の実装 (コーディング)

$\boldsymbol{V}_F^i$ から $\boldsymbol{V}_P^i$ を求める過程を実装 (Python)

Gx処理定義(Python)

```python
# G(X_n^i)を求める処理(非サイクル構造)
def getGxFromViForNc(X, n):
    act_v[n][i_rev-1,:] = -(np.linalg.inv(L[i]) @ (X.T + M[i] @ act_v[n][i_rev,:].T
                                                   + act_pi[n][i_rev-1,:].T)).T
    On = list(act_trans[n])
    Gn_ks = []
    for ix in On:
        Gn_ks.append(act_v[n][i_rev-1,:] - act_v[ix][i_rev-1,:]
                     - np.matrix(np.full((1,J), act_cost[n][ix]), dtype='float'))
    return np.minimum.reduce(Gn_ks)

# G(X_n^i)を求める処理(サイクル構造)
def getGxFromViForC(X, nc):
    act_v[nc[0]][i_rev-1,:] = -(np.linalg.inv(L[i]) @ (X.T + M[i] @ act_v[nc[0]][i_rev,:].T
                                                       + act_pi[nc[0]][i_rev-1,:].T)).T
    act_v[nc[1]][i_rev-1,:] = -(np.linalg.inv(L[i]) @ (X.T + M[i] @ act_v[nc[1]][i_rev,:].T
                                                       + act_pi[nc[1]][i_rev-1,:].T)).T
    On = list(act_trans[nc[0]])
    On2 = list(act_trans[nc[1]])

    Gn_ks = []
    for ix in On:
        Gn_ks.append(act_v[nc[0]][i_rev-1,:] - act_v[ix][i_rev-1,:]
                     - np.matrix(np.full((1,J), act_cost[nc[0]][ix]), dtype='float'))

    Gn_ks2 = []
    for iy in On2:
        Gn_ks2.append(act_v[nc[1]][i_rev-1,:] - act_v[iy][i_rev-1,:]
                      - np.matrix(np.full((1,J), act_cost[nc[1]][iy]), dtype='float'))
    Gn_ks2.append(np.minimum.reduce(Gn_ks))
    return np.minimum.reduce(Gn_ks2)
```

Merit関数(Python)

```python
for i in range(I-1):
    # 終端アクティビティFの価値計算
    i_rev = I-1-i
    act_v["F"][i_rev-1,:] = ( -np.linalg.inv(L[i_rev-1])
                              @ (M[i_rev-1] @ act_v["F"][i_rev,:].T + act_pi["F"][i_rev-1,:].T) ).T
    ###### Vpを求める処理 ######
    # 初期値X_n^{ik}, d^k
    Xn_ik = np.matrix(np.full((1,J), 0), dtype='float')
    d_k = np.matrix(np.full((1,J), 0), dtype='float')

    k = 1
    isExist = True
    while isExist:  # Xn_ikが更新される限りkを回す
        ###### Step1 #####
        d_k = np.maximum(Xn_ik - getGxFromViForNc(Xn_ik, "P"),
            np.matrix(np.full((1,J), 0)), dtype='float') - Xn_ik
        ###### Step2 #####
        phi = []
        L = 1000  # nの分母
        for l in range(L + 1):
            Xn_re = Xn_ik + (l / L) * d_k
            G_x = getGxFromViForNc(Xn_re, "P")
            H_x = np.maximum(Xn_re - G_x, np.matrix(np.full((1,J), 0))) - Xn_re
            phi0 = -G_x @ H_x.T - (1 / 2) * H_x @ H_x.T
            phi.append(phi0[0,0])  # リスト格納
        als = np.argmin(phi)  # 最小のインデックス1*
        ##### Step3 #####
        Xn_ik_re = Xn_ik + (als / L) * d_k
        ##### Step4 #####
        if (Xn_ik_re == Xn_ik).all():
            isExist = False  # 収束したのでkを止める
        else:
            k = k + 1
            Xn_ik = Xn_ik_re
```

$I = J = 400$

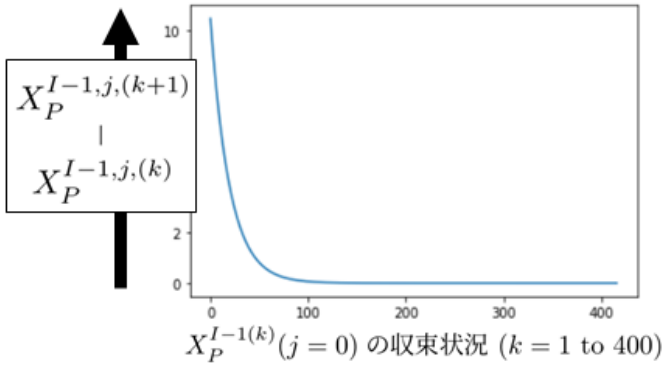
アクティビティ"P"の価値導出 ($\boldsymbol{V}_P^i$ の結果確認)

あらかじめ全時刻 i で $\boldsymbol{V}_F^i$ から $\boldsymbol{V}_P^i$ のみを計算

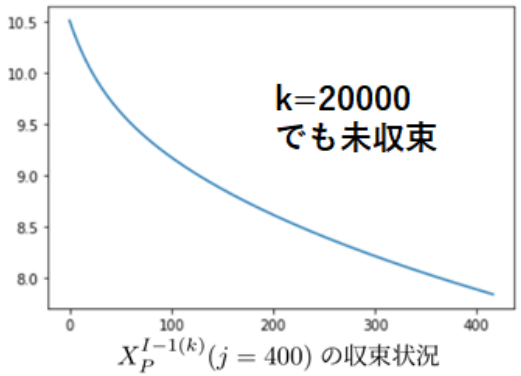
$\boldsymbol{X}_P^{I-1}$ の時点で収束しなかった ($\rightarrow k = 400$ (約 1 時間半) 経過)

起こっている現象

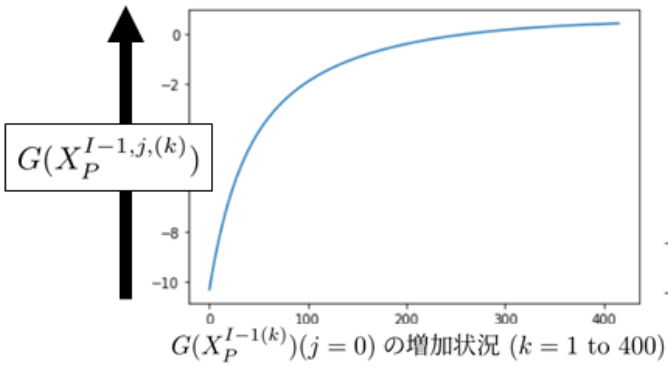
- どの i でも，$\alpha^* = 1(l^* = 1000)$ が選ばれる
- $\boldsymbol{X}_P^i$ の要素 ($j = 390$ まで) は収束ぎみ，391 以降は時間がかかる
- $\boldsymbol{G(X_P^i)}$ の要素 ($j = 390$ まで) は $\fallingdotseq 0$，391 以降はゆっくり増加 (負$\rightarrow 0$)



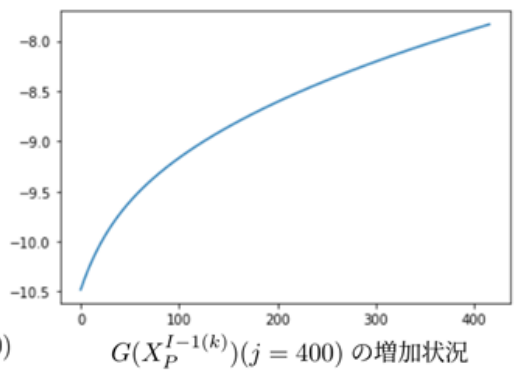
$X_P^{I-1(k)}(j=0)$ の収束状況 ($k = 1$ to 400)



$X_P^{I-1(k)}(j=400)$ の収束状況



$G(X_P^{I-1(k)})(j=0)$ の増加状況 ($k = 1$ to 400)



$G(X_P^{I-1(k)})(j=400)$ の増加状況

問題 ①

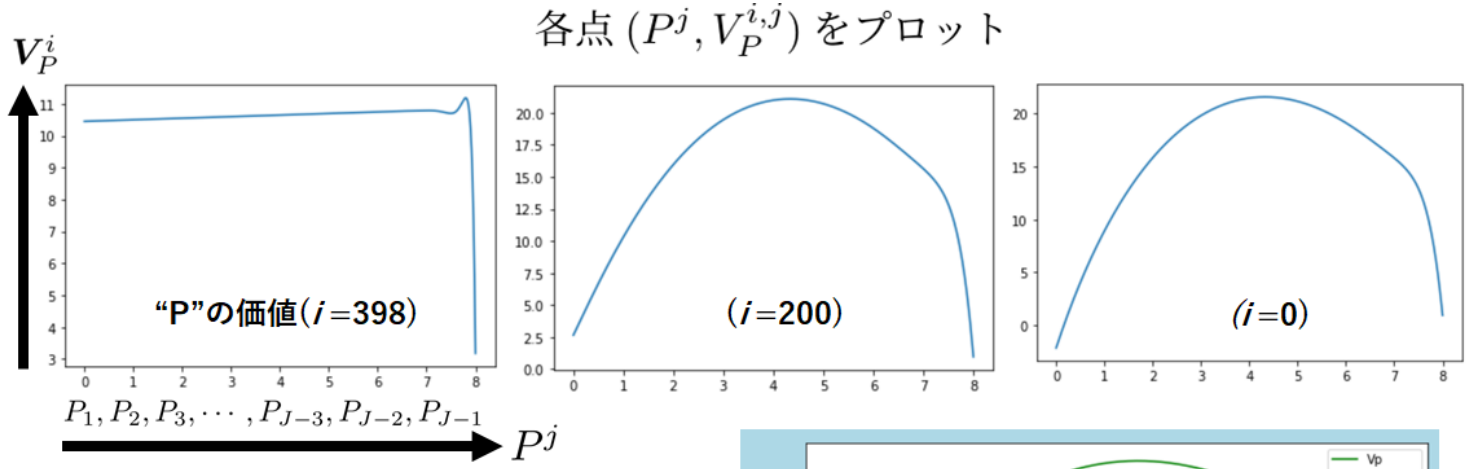
全時刻 i で $\boldsymbol{X}_n^i$ を収束させ，$\boldsymbol{V}_F^i, \boldsymbol{V}_P^i, \boldsymbol{V}_R^i, \boldsymbol{V}_S^i, \boldsymbol{V}_O^i$ を求めたい

アクティビティ"P"の価値導出 (V_P^i の結果確認)

各種考察

- ※ 前ページ = 収束判定「J 個の要素全て一致」の場合
- 初期値: $\boldsymbol{X}_n^{i(1)} = -\mathbf{10}$ や $\boldsymbol{X}_n^{i(1)} = \mathbf{100}$ でも収束状況は変わらなかった

収束判定「少なくとも要素 1 つが一致」の場合でも適切な結果でなかった (下図)

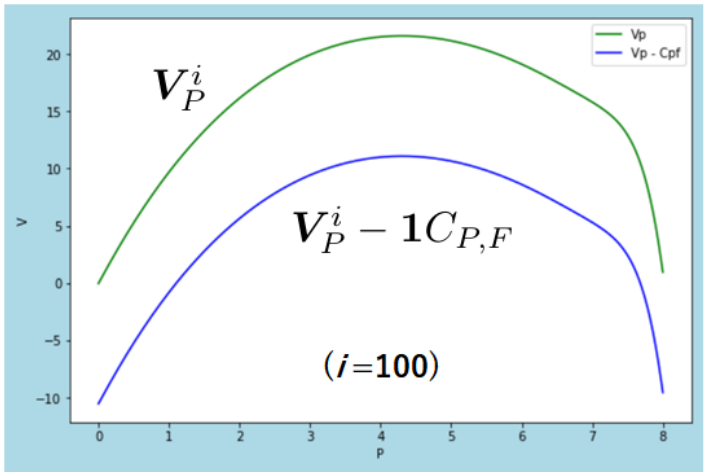


$i = 0$: $k = 653$,

$i = 1 \sim 10$: $k = 100 \sim 50$,

i=10 以降: 常に $k = 63$ で収束

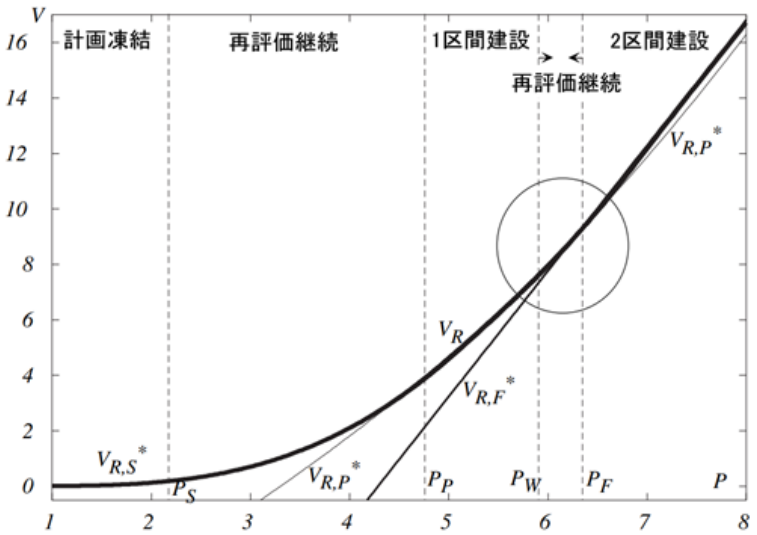
$\boldsymbol{V}_P^i$ を全て計算できたが,
$\boldsymbol{V}_P^i - \mathbf{1}C_{P,F}$ と接しなかった



アクティビティ"P"の価値導出 ($\boldsymbol{V}_P^i$ の結果確認)

懸念 4(再).

[1] 各図は exp 型の増加だが,
結果の $\boldsymbol{V}_F^i$, $\boldsymbol{V}_P^i$ が増加→減少の曲線になっている点



"R"の価値関数 t=15 (≒ i = 300)

問題②

[1] の各図と同様な形
($\boldsymbol{V}_n^i - \mathbf{1}C_{n,m}$ と $\boldsymbol{V}_n^i$ が接するような形と思われる) の再現

問題の解決

- 問題１と２を解決するために６つの懸念や各所の解釈違いが関わっていると考えられる

[1](28) 式の導出

- Crank-Nicholson 法による差分化→行列化の手順を確認

$$\frac{\partial V(t,P)}{\partial t}+\alpha(t,P)\frac{\partial V(t,P)}{\partial P}+\frac{1}{2}\{\sigma(t,P)\}^2\frac{\partial^2 V(t,P)}{\partial P^2}-rV(t,P)+\pi(t,P)=0$$

差分化 (Crank-Nicolson法)

[4] $$\frac{V_j^{i+1}-V_j^i}{\Delta T}+\frac{1}{2}\alpha_j^i\left(\frac{V_{j+1}^i-V_{j-1}^i}{2\Delta P}+\frac{V_{j+1}^{i+1}-V_{j-1}^{i+1}}{2\Delta P}\right)+\frac{1}{2}\frac{(\sigma_j^i)^2}{2}\left(\frac{V_{j+1}^i-2V_j^i+V_{j-1}^i}{(\Delta P)^2}+\frac{V_{j+1}^{i+1}-2V_j^{i+1}+V_{j-1}^{i+1}}{(\Delta P)^2}\right)$$
$$-rV_j^i+\pi_j^i=0$$

$$\left(-\frac{\alpha_j^i}{4\Delta P}+\frac{(\sigma_j^i)^2}{4(\Delta P)^2}\right)V_{j-1}^i+\left(-\frac{1}{\Delta T}-\frac{(\sigma_j^i)^2}{2(\Delta P)^2}-r\right)V_j^i+\left(\frac{\alpha_j^i}{4\Delta P}+\frac{(\sigma_j^i)^2}{4(\Delta P)^2}\right)V_{j+1}^i$$
$$+\left(-\frac{\alpha_j^i}{4\Delta P}+\frac{(\sigma_j^i)^2}{4(\Delta P)^2}\right)V_{j-1}^{i+1}+\left(\frac{1}{\Delta T}-\frac{(\sigma_j^i)^2}{2(\Delta P)^2}\right)V_j^{i+1}+\left(\frac{\alpha_j^i}{4\Delta P}+\frac{(\sigma_j^i)^2}{4(\Delta P)^2}\right)V_{j+1}^{i+1}+\pi_j^i=0$$

項の置き換え

$$a_j^i=-\frac{\alpha_j^i}{4\Delta P}+\frac{(\sigma_j^i)^2}{4(\Delta P)^2} \qquad c_j^i=\frac{\alpha_j^i}{4\Delta P}+\frac{(\sigma_j^i)^2}{4(\Delta P)^2} \qquad (\alpha_j^i=\alpha P^j)$$
$$b_j^i=-\frac{1}{\Delta T}-\frac{(\sigma_j^i)^2}{2(\Delta P)^2}-r \qquad d_j^i=\frac{1}{\Delta T}-\frac{(\sigma_j^i)^2}{2(\Delta P)^2} \qquad (\sigma_j^i=\sigma P^j)$$

[1](28) 式の導出

$$a_j^i V_{j-1}^i + b_j^i V_j^i + c_j^i V_{j+1}^i + a_j^i V_{j-1}^{i+1} + d_j^i V_j^{i+1} + c_j^i V_{j+1}^{i+1} + \pi_j^i = 0 \qquad (1)$$

$\underline{a_j^i V_{j-1}^i + b_j^i V_j^i + c_j^i V_{j+1}^i}$ について $j = 1$ から $j = J-1$ まで展開

$$\begin{aligned}
j &= 1 && a_1^i V_0^i + b_1^i V_1^i + c_1^i V_2^i \\
j &= 2 && a_2^i V_1^i + b_2^i V_2^i + c_2^i V_3^i \\
j &= 3 && a_3^i V_2^i + b_3^i V_3^i + c_3^i V_4^i \\
&\vdots && \vdots \\
j &= J-3 && a_{J-3}^i V_{J-4}^i + b_{J-3}^i V_{J-3}^i + c_{J-3}^i V_{J-2}^i \\
j &= J-2 && a_{J-2}^i V_{J-3}^i + b_{J-2}^i V_{J-2}^i + c_{J-2}^i V_{J-1}^i \\
j &= J-1 && a_{J-1}^i V_{J-2}^i + b_{J-1}^i V_{J-1}^i + c_{J-1}^i V_J^i
\end{aligned}$$

[1](28) 式の導出

同様の V_j^i をまとめることで行列化

$$\text{for } j \text{ in } (1 \le j \le J-1) \qquad a_j^i V_{j-1}^i + b_j^i V_j^i + c_j^i V_{j+1}^i = \begin{bmatrix} b_1^i & c_1^i & 0 & 0 & \cdots & 0 \\ a_2^i & b_2^i & c_2^i & 0 & \cdots & 0 \\ 0 & a_3^i & b_3^i & c_3^i & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{J-3}^i & b_{J-3}^i & c_{J-3}^i & 0 \\ 0 & \cdots & 0 & a_{J-2}^i & b_{J-2}^i & c_{J-2}^i \\ 0 & \cdots & 0 & 0 & a_{J-1}^i & b_{J-1}^i \end{bmatrix} \begin{bmatrix} V_1^i \\ V_2^i \\ V_3^i \\ \vdots \\ V_{J-3}^i \\ V_{J-2}^i \\ V_{J-1}^i \end{bmatrix} + \begin{bmatrix} a_1^i V_0^i \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ c_{J-1}^i V_J^i \end{bmatrix}$$

※ (1) 式 第 4 項以降も同様に展開

まとめると以下の式が得られる

$$\begin{bmatrix} b_1^i & c_1^i & 0 & 0 & \cdots & 0 \\ a_2^i & b_2^i & c_2^i & 0 & \cdots & 0 \\ 0 & a_3^i & b_3^i & c_3^i & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{J-3}^i & b_{J-3}^i & c_{J-3}^i & 0 \\ 0 & \cdots & 0 & a_{J-2}^i & b_{J-2}^i & c_{J-2}^i \\ 0 & \cdots & 0 & 0 & a_{J-1}^i & b_{J-1}^i \end{bmatrix} \begin{bmatrix} V_1^i \\ V_2^i \\ V_3^i \\ \vdots \\ V_{J-3}^i \\ V_{J-2}^i \\ V_{J-1}^i \end{bmatrix} + \begin{bmatrix} a_1^i V_0^i \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ c_{J-1}^i V_J^i \end{bmatrix} + \begin{bmatrix} d_1^i & c_1^i & 0 & 0 & \cdots & 0 \\ a_2^i & d_2^i & c_2^i & 0 & \cdots & 0 \\ 0 & a_3^i & d_3^i & c_3^i & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{J-3}^i & d_{J-3}^i & c_{J-3}^i & 0 \\ 0 & \cdots & 0 & a_{J-2}^i & d_{J-2}^i & c_{J-2}^i \\ 0 & \cdots & 0 & 0 & a_{J-1}^i & d_{J-1}^i \end{bmatrix} \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ V_3^{i+1} \\ \vdots \\ V_{J-3}^{i+1} \\ V_{J-2}^{i+1} \\ V_{J-1}^{i+1} \end{bmatrix} + \begin{bmatrix} a_1^{i+1} V_0^{i+1} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ c_{J-1}^{i+1} V_J^{i+1} \end{bmatrix} + \begin{bmatrix} \pi_1^i \\ \pi_2^i \\ \pi_3^i \\ \vdots \\ \pi_{J-3}^i \\ \pi_{J-2}^i \\ \pi_{J-1}^i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\boxed{a_j^i V_{j-1}^i \quad + b_j^i V_j^i \quad + c_j^i V_{j+1}^i} \qquad \boxed{a_j^i V_{j-1}^{i+1} \quad + d_j^i V_j^{i+1} \quad + c_j^i V_{j+1}^{i+1}}$$

懸念事項

- 別の項が出現したため，$\boldsymbol{V}^i$ を簡単に求められない
- $\mathbf{0}$ とみなし，論文と同じ形で計算したこと

[1](28)式

$$\boldsymbol{L}^i \begin{bmatrix} V_1^i \\ V_2^i \\ V_3^i \\ \vdots \\ V_{J-3}^i \\ V_{J-2}^i \\ V_{J-1}^i \end{bmatrix} + \boldsymbol{M}^i \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ V_3^{i+1} \\ \vdots \\ V_{J-3}^{i+1} \\ V_{J-2}^{i+1} \\ V_{J-1}^{i+1} \end{bmatrix} + \boldsymbol{\pi}^i = \mathbf{0}$$

懸念 7

- $\mathbf{0}$ とみなせる要因があるのか？
- 何かしらの近似が行われるのか？

$\mathbf{0}$ として計算

自身で導出した式

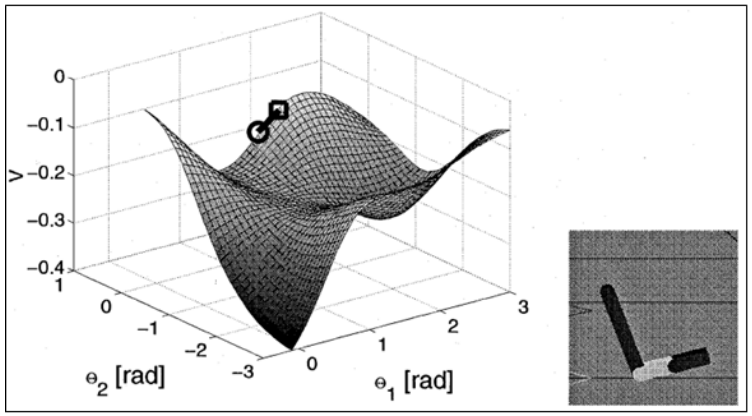
$$\boldsymbol{L}^i \begin{bmatrix} V_1^i \\ V_2^i \\ V_3^i \\ \vdots \\ V_{J-3}^i \\ V_{J-2}^i \\ V_{J-1}^i \end{bmatrix} + \begin{bmatrix} a_1^i V_0^i \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ c_{J-1}^i V_J^i \end{bmatrix} + \boldsymbol{M}^i \begin{bmatrix} V_1^{i+1} \\ V_2^{i+1} \\ V_3^{i+1} \\ \vdots \\ V_{J-3}^{i+1} \\ V_{J-2}^{i+1} \\ V_{J-1}^{i+1} \end{bmatrix} + \begin{bmatrix} a_1^{i+1} V_0^{i+1} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ c_{J-1}^{i+1} V_J^{i+1} \end{bmatrix} + \boldsymbol{\pi}^i = \mathbf{0}$$

現在行いたいこと

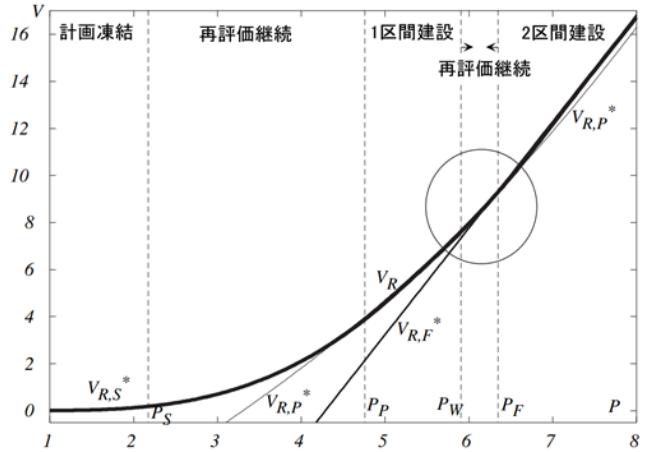
回答をいただくまでに，自分の研究部分 (後半部分) を考える

案 (評価関数推定) の棄却

- 採用した「意思決定モデル」と説明できることが被るため
- 「意思決定モデル」を別の方向に用いる



“評価の高い行動”をとる



太曲線(最適値関数)
で戦略の切り替わりが分かる

“評価の高い戦略”に遷移

テーマの変更

「キーワードの流行り廃り分析」を再度行う

背景的なもの

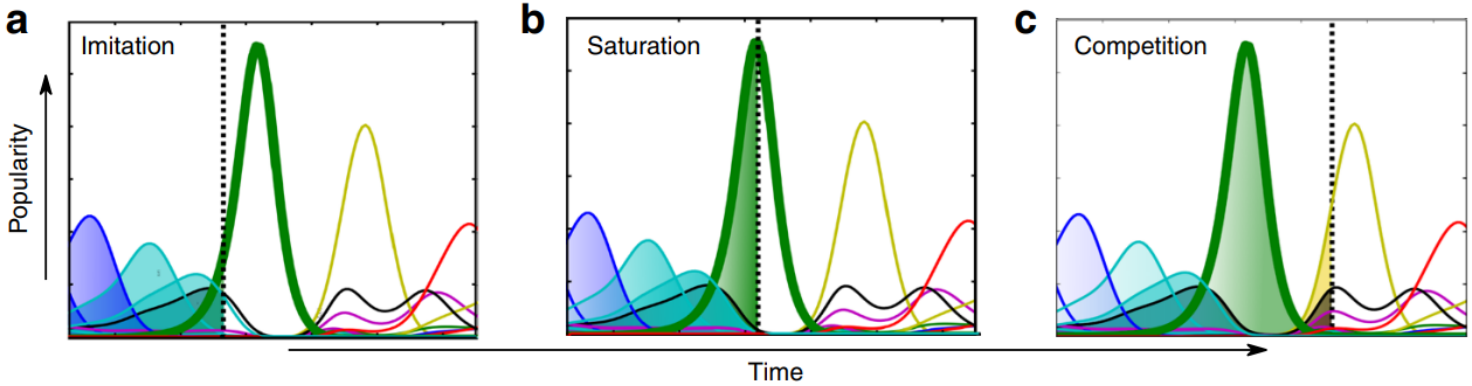
・「流行の流行り廃りを高い精度で説明できるモデルの開発」を行う研究いくつか
→「単語が" 人気" という食料を奪い合う」食物連鎖モデルによる説明 (下図)

ロトカ・ヴォルテラ -注目モデル

$$\frac{dL_i(t)}{dt} = r_p L_i(t)\left(1 - \frac{r_c}{K} Y_i(t) - c\sum_{j=1, j\neq i}^{N} L_j(t)\right)$$

$$\frac{dY_i(t)}{dt} = L_i(t) - \alpha Y_i(t).$$

$L_i(t)$：ハッシュタグ i の出現回数
Y：補助変数
N トピック数
K：成長モデルの収容力
c：重複コンテキストの量
r_p, r_c：1 トピック当たりの成長率



Lorenz-Spreen, P., Mønsted, B.M. et al., 2019

背景的なもの

流行り廃りモデルの目的

- パラメータ推定でモデルが実データに近くなるようにチューニング
- 現実のデータにどれほどフィットするかの確認
- (将来的に) 高精度モデルで将来の流行りを予測

「キーワード注目度の時系列変化の観測」に焦点

行われていなさそうなこと

「キーワードがいつ注目状態 or 風化状態に切り替わるのか」根拠的な説明

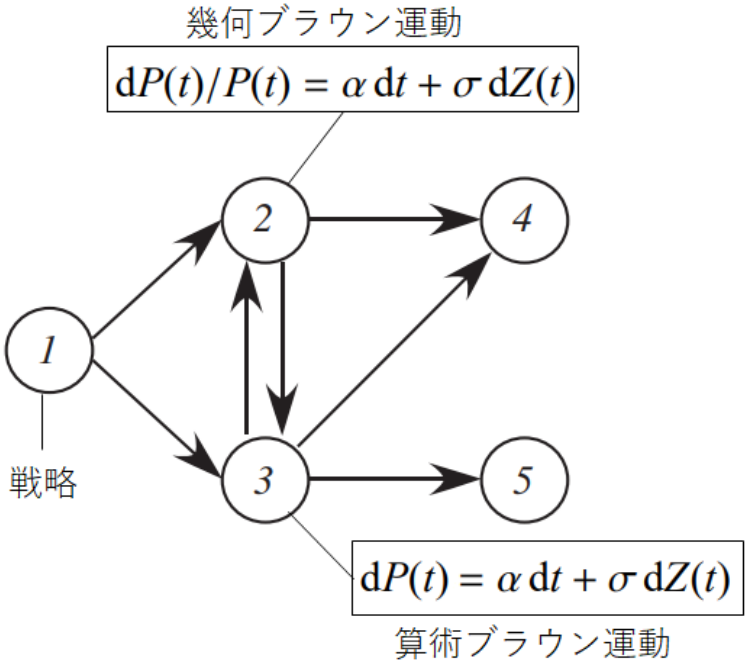
提案手法的なもの

提案

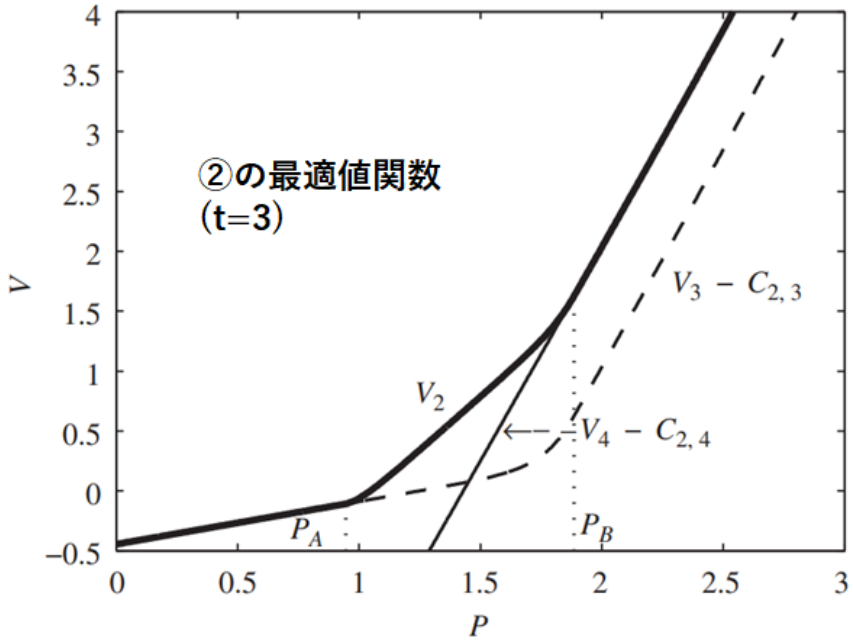
何らかのキーワード遷移ネットワークを仮定し，単語の注目が変動する現象をより説明可能にする

オプション・グラフ・モデルで説明できること

各時刻 t，需要量 P に応じて「戦略を維持する」or「切り替える」を選択する瞬間
戦略＝単語　需要量＝注目度
→ 1 単語と他単語の注目度 (確率変動) に応じて単語が「流行状態」「廃り状態」が切り替わる瞬間を観測できる



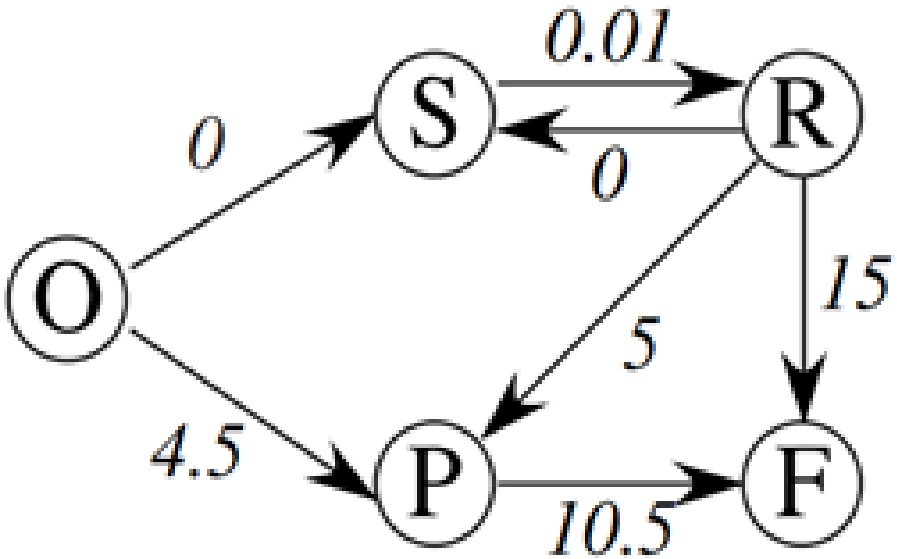
“P”が異なる確率過程でも適用可



単語1の人気需要がXのとき，
人気需要がYの単語2に乗り換えた方が評価が高い

課題

- 単語が一定のネットワークの中で状態遷移すると仮定する必要
- 単語間を遷移するときのコストを設定する必要



まとめ

これからの展望

- 流行遷移分析の研究としての的確な落としどころを見つける