ベイズ推論による沖縄県のオゾン濃度モデリング

田崎盛也

Ozone Concentration Modeling in Okinawa using Bayesian Inference

Moriya TASAKI

要旨： 沖縄県の中部に位置する沖縄局と与那城局のポテンシャルオゾン濃度（PO）の変動を日内変動パターンと季節依存の変動幅の積によって表現される日内変動成分と季節変動成分の和で表すモデル構築を試みた．このモデルを構築するためのパラメータ推定のためにベイズ推論に用いられる手法であるマルコフ連鎖モンテカルロ法（MCMC）を用いて推定を行った．PO の季節変動成分は春に高く，夏に濃度低下し，秋から冬にかけて上昇する傾向がみられた．PO の日変動パターンは 6 時から上昇した濃度は 12－13 時ごろに最大値を示した後徐々に減少するという動きであった．また，日変動パターンの変動幅は沖縄局で 5 ppb 程度であり，与那城局はその半分程度であった．季節によるオゾン生成能の変化について，沖縄局では明瞭な変化が確認できず，与那城局では夏季に増大する傾向があるものの，推定値の不確実性が高いためはっきりとした結論は出せなかった．また，モデルによる予測分布と観測データの分布を比較するとデータの大まかな傾向を捉えてはいるものの，分布の多峰性までは再現できておらず，更なる改良の余地があることが分かった．

Key words: オゾン，ポテンシャルオゾン（PO），ベイズ推論，マルコフ連鎖モンテカルロ法（MCMC）

Ⅰ　はじめに

光化学オキシダント（以下，Ox）は光化学反応によって生じるオゾンなどの酸化性物質の総称である．近年の沖縄県の大気常時監視測定結果では $PM_{2.5}$ も含めほとんどの環境基準は達成できる傾向にあるが，Ox のみは依然として環境基準を達成することが厳しい状況にある．Ox の環境基準達成状況は全国的にも低水準であり，2018 年度（平成 30 年度）では全国の測定局 1,183 局のうち環境基準を達成した局は 1 局のみとなっている [1]．沖縄県の Ox の環境基準超過の要因は越境大気汚染によるものがその主な原因と考えられているが，越境大気汚染の影響が定量的にどの程度なのか把握することは困難であった．板野・高倉(2011)の既報 [2]によると，都市大気のオゾン濃度についてバックグラウンド濃度（越境大気汚染を含む季節の濃度変動）と時間変動（ローカルな Ox 生成）に切り分けるベイズモデルが報告されている．なおオゾンの濃度は一酸化窒素（NO）との反応により減少するため，既報 [2]の解析では観測されたオゾンの値をそのまま用いるのではなく，この減少を打ち消したポテンシャルオゾン（PO）を用いている．これを参考にして沖縄県のオゾン濃度の解析を行ったのでその結果を報告する．

Ⅱ　方法

１．ベイズ推論

ベイズ推論とは観測データ Y と未知のパラメータ X について同時分布 $p(X,Y)$を構築し，データ Y が与えられた時のパラメータ X の条件付き分布 $p(X|Y)$を求めるという枠組みのことである [3]．同時分布 $p(X,Y)$をパラメータ X について積分すると次式のように周辺分布 $p(Y)$が得られる．

$$p(Y) = \int p(X,Y)\, dX \tag{1}$$

条件付き分布 $p(X|Y)$は同時分布 $p(X,Y)$と周辺分布 $p(Y)$によって次式のように表せる．

$$p(X|Y) = \frac{p(X,Y)}{p(Y)} \tag{2}$$

また，同様にパラメータ X が与えられた時のデータ Y の条件付き分布 $p(Y|X)$を考えることもでき，次式のように表せる．

$$p(Y|X) = \frac{p(X,Y)}{p(X)} \tag{3}$$

式(1)－(3)の 3 つを組み合わせることでいわゆるベイズの定理と呼ばれる次の式を導くことができる．

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)p(X)}{\int p(X,Y)dX} \tag{4}$$

このとき $p(Y|X)$は尤度または確率モデル，$p(X)$は事前分布，$p(X|Y)$は事後分布と呼ばれる．また，分母の周辺分布 $p(Y)$は分子を積分計算することで得られるため，推論に必要な事後分布は尤度と事前分布を設定することで得られる．ただし，事後分布の導出は式(4)の分母の積分計算が複雑となることが多く，ほとんどの場合に解析的には求まらない．そのような場合の事後分布の近似手法とし

てマルコフ連鎖モンテカルロ法（Mankov Chain Monte Carlo：以下, MCMC）や変分ベイズ法などが用いられる．MCMC は乱数を用いて多数のサンプリングを繰り返すことにより目的の事後分布を得る方法である．MCMC は大気環境の分野でもベイズによるモデリングの際に事後分布を得る方法として用いられている [2,4-6]．変分ベイズ法は解析的には求まらない事後分布を簡単に計算可能な近似分布に置き換え，ある基準と制約の下で元の事後分布との差を最小化することで近似の事後分布を得る方法である．今回は事後分布を得る方法として，MCMC を用いる．

２．解析に用いたデータ

沖縄県の中部に位置する測定局である沖縄局と与那城局を対象とした（図 1）．対象測定局における測定項目 Ox, NO_2，NOx の 2013－2017 年度までの 1 時間値を用いた．データは国立環境研究所が公開している環境数値データベースの大気環境の時間値データ [7]を利用した．これらのデータから PO を次の式より算出した．

$$\mathrm{PO} = [\mathrm{O_3}] + [\mathrm{NO_2}] - \alpha[\mathrm{NOx}] \qquad (5)$$

α は一時排出窒素酸化物（NOx）中の二酸化窒素（NO_2）の比率であり，これまでの報告 [2,8,9]と同様に 0.1 とした．PO を算出した後は MCMC 計算の収束の向上を狙い平均が 0，分散が 1 になるように PO データを正規化した．

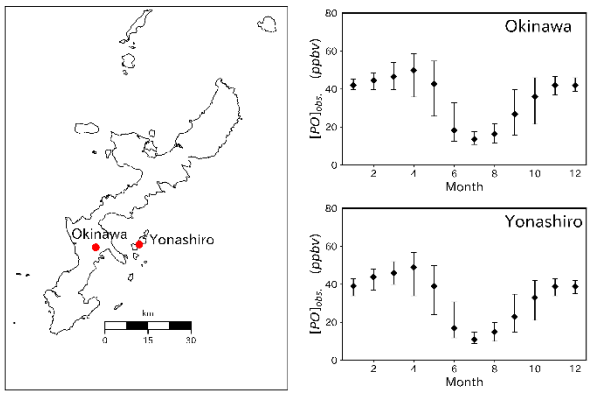

図 1．測定地点（左）と同地点におけるポテンシャルオゾン（PO）観測値（右）．◆は中央値，エラーバーは 25－75%タイル値の範囲を表す．

３．モデルの概要

板野・高倉(2011)のモデルでは観測されるオゾンの濃度は日内変動と季節変動の合成によって表されるとしている．板野・高倉(2011)では都市毎に同年度の 5 地点のデータを用いて解析を行っているが，本報告では年度毎の特徴ではなく地点の持つ特徴を見るために 1 地点の 2013 年度から 2017 年度のデータを使用して解析を行っている．

表 1．使用した Python と主なライブラリのバージョン．解析環境で使用しなかったライブラリは空欄としている．

	Anaconda	Google Colabolatry
Name	Version	Version
Pyhton	3.6.8	3.6.9
Numpy	1.16.4	
pandas	0.24.2	1.0.5
Matplotlib	3.1.0	
seaborn	0.9.0	
PyMC3	3.7	3.7
Theano	1.0.4	
ArviZ	0.5.1	0.9.0

解析には Anaconda 社が提供している Anaconda ディストリビューション(https://www.anaconda.com)により解析環境を構築し，プログラミング言語 Python（3.6.8）とベイズ推論ライブラリ PyMC3（3.7）を使用した．その他のライブラリについては表 1 を参照．ただし，使用したコンピュータのメモリ不足により計算できなかった際には Web 上で利用可能なクラウド環境 Google Colaboratory（https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja）を使用した．データの整形と MCMC を実行する際の Python コードはこの報告の付録に掲載する．PyMC3 のコードについては PyMC3 公式ドキュメント（https://docs.pymc.io/）にて使い方が確認できるほか，MCMC 計算実行についての和書は中妻（2019）の書籍が参考になる [10]．

モデルの詳細について，真の PO 濃度（C_{mh}）が月により変化する季節変動成分（A_m）と日内変動成分（$B_m \times D_h$）の和で表されるとした（式(6)）．日内変動成分について，変動パターン（D_h）は月に依存しないとし，その変動幅（B_m）は月に依存するとしてその積で表している．観測値（O_{mh}）は真の濃度C_{mh}を平均とし，σ を標準偏差とする正規分布（normal distribution, $\mathcal{N}$）に従うと仮定した（式(7)）．σ は観測誤差を考慮した誤差項である．この仮定が前述した尤度を設定したことになる．A_mとB_mは 12 次元ベクトル，D_hは 24 次元ベクトル，σはスカラー，C_{mh}は 12×24 の行列である．

$$C_{mh} = A_m + B_m \times D_h \qquad (6)$$

$$O_{mh} \sim \mathcal{N}(C_{mh}, \sigma) \qquad (7)$$

$$A_m \sim \mathcal{U}(-30, 30) \qquad (8)$$

$$B_m \sim \mathcal{U}(0, 10) \qquad (9)$$

$$D_h \sim \mathcal{U}(0, 30) \qquad (10)$$

$$\sigma \sim \mathcal{U}(0,10) \qquad (11)$$

$$(m = 1,\cdots,12, \mathrm{h} = 1,\cdots,24)$$

パラメータの事前分布について，全て一様分布（uniform distribution，$\mathcal{U}$）とし，式(8)－(11)のようにパラメータの範囲を設定した．B_m，D_h，σ は非負である必要があるため，0 未満にならないように範囲を設定した．また，板野・高倉モデルの事前分布のパラメータの範囲に比べて狭く設定しているが，これはパラメータの範囲の絶対値が大きいと収束しない系列が発生したためであり，範囲を絞ったのはその対策のためである．

以上の条件によって定義されるモデルをモデル 1 とした．また，モデル 1 に変更点を加えたモデルを次のよう定義する．モデル 2：標準偏差 σ が月毎に変動するとして，σ をスカラーではなく 12 次元ベクトルとしたモデル．モデル 3：日内変動成分の月による変動幅を表すパラメータ B_m を全て 1 と固定したモデル．モデル 4：モデル 2 とモデル 3 の変更点を合わせたモデル．

パラメータ推定には PyMC3 のデフォルト設定であり MCMC の手法の一つであるハミルトニアンモンテカルロ法 NUTS（No-U-Turn Sampler）を用いて 30,000 回の乱数を生成した．MCMC の結果得られた分布を使用するた

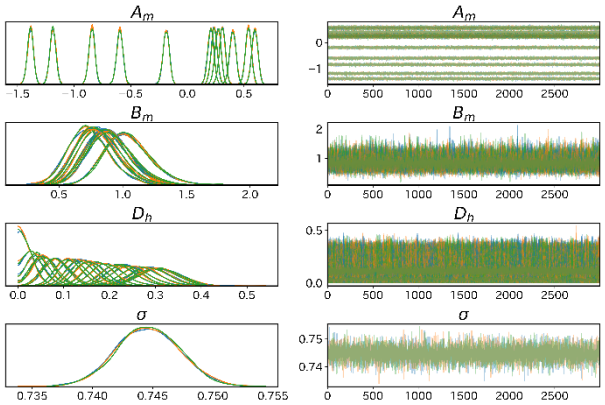

図 2．各パラメータの事後分布（左）とそのトレースプロット（右）．3 つの系列の事後分布はほぼ同じ形状であり，トレースプロットの形状には変化がない．

めには MCMC 結果が収束している必要があるが，乱数系列の初期は初期値の影響を受け収束していない可能性があるためこれを除く必要がある．

そのため先程のサンプルの前に 5,000 点の乱数を生成しバーンインとして除いている．また，MCMC サンプルはしばしば自己相関が生じるが，この緩和には間引きを行うことで対処ができる．得られた乱数 30,000 点から 10 点毎のサンプリングにより間引きし 3,000 点のデータと

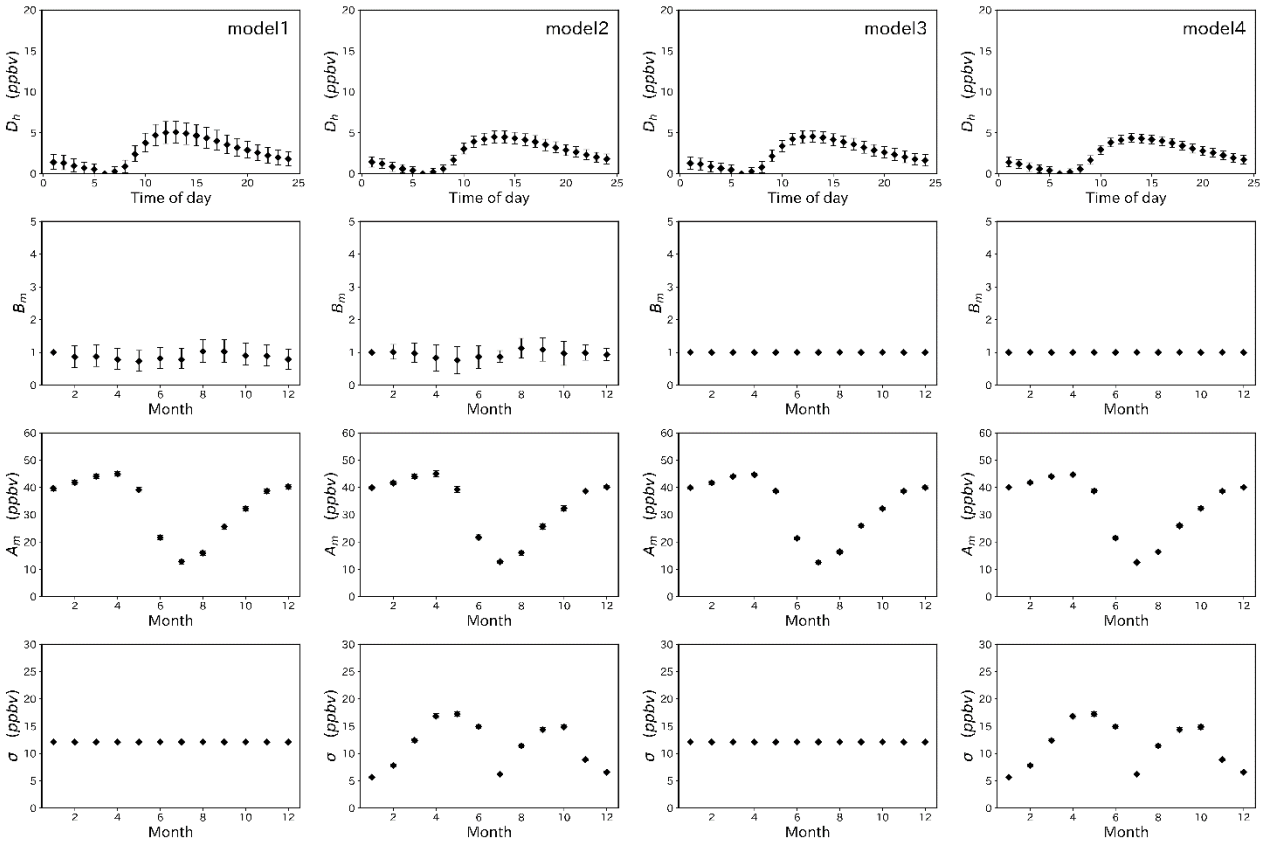

図 3．沖縄局における各パラメータの推定結果．◆は中央値，エラーバーは 95%確信区間の範囲を表す．

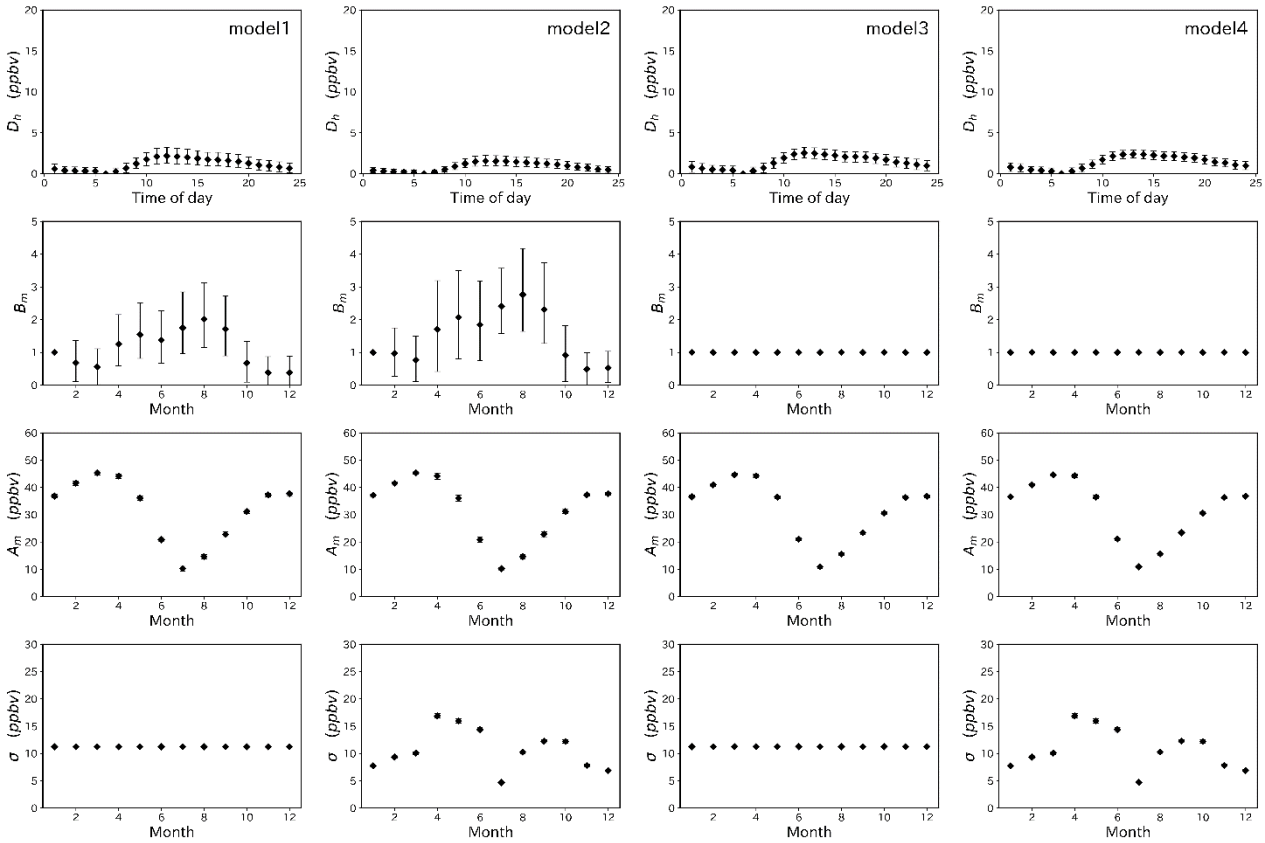

図 4．与那城局における各パラメータの推定結果．◆は中央値でエラーバーは 95%確信区間の範囲を表す．

した．これを 1 つの系列とし，同様に 3 回繰り返し全 9,000 点のデータで得られる確率分布を事後分布とした．

Ⅲ　結果および考察

１．MCMC 計算の収束および妥当性の判定

MCMC 計算の収束判定にはトレースプロットの形状の確認や Gelman-Rubin の収束判定（R-hat）が用いられる [10]．図 2 に MCMC の結果得られた沖縄局のモデル 1 の場合の事後分布の確率密度関数（左図）とトレースプロット（右図）を示す．注意点としてトレースプロットは平均 0，分散 1 と規格化されているデータによる MCMC サンプル形状に違いはなくほぼ同じ形状であり，トレースプロットもある値の周りで推移しているため収束していると判断できる．また，R-hat についてもすべてのパラメータで 1.1 以内となっているため良好に収束していると判断した．ここでは省略したが他のモデルや与那城局のデータについても同様の結果が得られている．

２．事後分布

図 3 に沖縄局のデータ，図 4 に与那城局データでのモデル 1−4 のパラメータA_m，B_m，D_h，σ の推定結果を示す．以下の議論は基本的に中央値を用いて行う．日内変動パターン（D_h）について，6 時から日中にかけて増加し，12−13 時に最大値を示した後徐々に減少するという特徴が 2 地点及び全てのモデルでみられた．沖縄局では全てのモデルで最大値を示した時間が 13 時であり，モデル 1 では 5.1 ppb，モデル 2 では 4.5 ppb，モデル 3 では 4.5 ppb，モデル 4 では 4.3 ppb と推定された．与那城局ではモデル 1−3 は 12 時に，モデル 4 は 13 時に最大値を示し，モデル 1 では 2.2 ppb，モデル 2 では 1.6 ppb，モデル 3 では 2.5 ppb，モデル 4 では 2.4 ppb と推定された．日内変動の変動幅（B_m）について，沖縄局ではモデル 1 で 5 月に最小値 0.73, 8 月に最大値 1.03 と推定され，モデル 2 では 5 月に最小値 0.75，8 月に最大値 1.12 と推定されたが，95%確信区間の幅が大きくその重なりも考慮すると季節による変動幅の変化があるとはいえない．また，与那城局ではモデル 1 で 11 月に最小値 0.38，8 月に最大値 2.01 と推定され，モデル 2 では 11 月に最小値 0.48，8 月に最大値 2.76 と推定され季節による傾向があるようにも見えるが，沖縄局に比べて与那城局は 95%確信区間の幅がさらに大きく不確実性が高いため，はっきりとした結論は出せない．本報告でB_mを固定したモデル 3，モデル 4 を導入したのはこれらの結果を受けて行ったものである．B_mの推定値が既報 [2]に比べて不確実性が

大きいのは日内変動パターン（D_h）が既報[2]では 12－13 ppb 程度あるのに対し，本報告ではその半分以下であることがその原因ではないかと考えている．季節変動成分（A_m）について，3－4 月にかけて最高値を示した後に減少に転じ，7 月に最低値を示すという変動が 2 地点及び全てのモデルでみられた．この変動は過去の沖縄県の PO の変動[11]とも一致しているため特徴をうまく抽出できたと考えられる．標準偏差 σ について，沖縄局ではスカラーと設定したモデル 1，3 の値に比べてベクトル化したモデル 2，4 では 3－6 月と 9－10 月に値が増加し，与那城局ではモデル 1，3 の値に比べてモデル 2，4 では 4－6 月と 9－10 月に値が増加した．これは増加した月の値のばらつきが大きいことを意味している．この意味については後述する．

3．WAIC によるモデルの比較

2 つの測定局に対して 4 種類のモデルを作成しそのパラメータの事後分布を確認したが，どのモデルが良いモデルといえるだろうか．一般的にモデルのパラメータを増やすとデータに対しての当てはまりは増す一方，データの要約という観点や本来当てはめを行いたくないノイズのようなものにまで過剰に適合する恐れがある．このような問題に対する一つの解決策として情報量規準がある．有名な規準として AIC（赤池情報量規準，Akaike Information Criterion）があり，データへの当てはまりとパラメータ数のバランスを満たすようなモデルの選択が可能となる．本報告ではベイズ推論によるモデル構築を行っていることから，AIC のベイズ版といえる WAIC（広く使える情報量規準，Widely Applicable Information Criterion または渡辺－赤池情報量規準，Watanabe-Akaike Information Criterion）[12,13]を用いてモデルを評価する．WAIC の計算結果を表 2 と図 5 に示す．

表 2．各モデル，測定局毎の WAIC の値とその標準誤差（SE）．

Station	Model	WAIC	SE
Okinawa	model1	92034	376
Okinawa	model2	81412	407
Okinawa	model3	92017	376
Okinawa	model4	81396	407
Yonashiro	model1	91563	375
Yonashiro	model2	82353	386
Yonashiro	model3	91598	374
Yonashiro	model4	82446	384

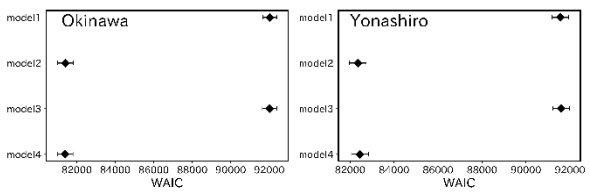

図 5．沖縄局における各モデルの WAIC の値（左）と与那城局における各モデルの WAIC の値（右）．

WAIC の観点で良いモデルとは WAIC の数値が他のモデルと比べて小さいモデルである．沖縄局ではモデル 4，与那城局ではモデル 2 が WAIC の観点から良いモデルとなった．ただし，モデル 2 と 4 の差は図 5 の標準誤差のエラーバーの範囲が重なっていることからも分かるようにその差がわずかである．そのためデータによっては WAIC の順序関係は変わる可能性がありうる．また，σ をベクトル化することは WAIC の低下に大きな影響を及ぼしていることが明らかである．また，WAIC によるモデルの選択は WAIC を適用した中での最良のモデルであり，選ばれたモデルよりもより良いモデルがないということを肯定していないことに注意が必要である．

4．予測分布

モデルの評価の際にベイズ推論の枠組みでは予測分布を確認することも重要である．予測分布は定義した事後分布の下で新たなデータ Y^*が得られた場合どのような値が観測されるのかというものであり，尤度を事後分布で平均したものがその定義となる（式(12)）．

$$p(Y^*|Y) = \int p(Y^*|X)\, p(X|Y)\, dX \qquad (12)$$

ベイズ推論ではこの予測分布が真の分布をよく近似するだろうという仮定を置いている[12]．

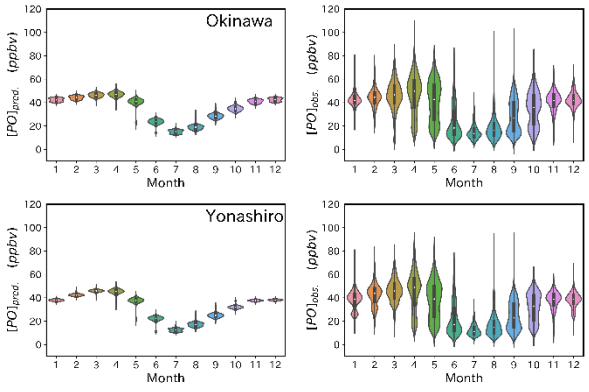

図 6．予測分布のバイオリンプロット（左）と観測データのバイオリンプロット（右）．

WAIC で良いとされたモデルについて，得られた事後分布を用いて予測分布を計算し，観測データと分布の形状の比較を行った．予測分布は 9,000 点のサンプリングを実行した．図 6 に予測分布と観測データのバイオリンプロットを示す．

予測分布と観測データのプロットの形状を比較すると中央値はある程度一致するなど大まかな傾向は再現できているといえるものの，ばらつきが大きな月のデータについては当てはまりが良くない．これは誤差に正規分布を仮定しているためモデルで表現できる値C_{mh}は単峰性となるが，ばらつきの大きな月では多峰性の傾向があるためである．前述したσの議論の際にスカラーからベクトルに変更して値が増加した月ほど観測データにおいて多峰性の特徴がみられるのが明らかである．多峰性の原因は季節の変わり目や気象条件によって生じると考えられるため，月と時刻のみを考慮したモデルアプローチではこの対処は難しい．これを解決するには，事前の情報として気象条件をモデルに導入し，大陸性または海洋性の気団の影響を状態変数としてこれを推定するようなモデルを構築すればより良い推定が可能になると思われるため今後の課題である．

Ⅳ まとめ

沖縄県の中部に位置する沖縄局と与那城局の PO データを用いてベイズ推論による PO 濃度のモデルを 4 種類構築した．パラメータの事後分布について、PO の日内変動パターン（D_h）は 2 つの地点，全てのモデルで 6 時から上昇した濃度は 12－13 時ごろに最大値を示した後徐々に減少するという動きと推定された．また，D_hの変動幅はモデルによって異なるものの，沖縄局で 4.3－5.1 ppb，与那城局は 1.6－2.5 ppb と推定された．日内変動の変動幅（B_m）について，沖縄局では明瞭な変化が確認できず，与那城局では夏季に増大する傾向があるものの，推定値の不確実性が高いためはっきりとした結論は出せなかった．季節変動成分（A_m）について，3－4 月にかけて最高値を示した後に減少に転じ，7 月に最低値を示すという変動が 2 地点及び全てのモデルでみられ，測定地点の特徴をうまく推定できたと思われる．WAIC による比較では沖縄局ではモデル 4，与那城局ではモデル 2 が良いモデルと評価された．WAIC で良いモデルとされた予測分布と PO 観測データの分布形状を比較すると，大まかな傾向は捉えているが，観測データにある多峰性の特徴が予測分布では表現できておらず，この特徴を再現するためにはモデルの改良が必要なことが示唆された．

Ⅴ 参考文献

1) 環境省．平成 30 年度　大気汚染状況について（https://www.env.go.jp/air/osen/jokyo_h30/index.html）．2020 年 8 月アクセス．
2) 板野泰之・高倉耕一（2011）ベイズ統計手法による都市大気オゾンの日内変動と季節変動の分離評価．大気環境学会誌，46(3)：179－186.
3) 須山敦志（2017）ベイズ推論による機械学習入門．講談社，pp243.
4) 久恒邦裕・山神真紀子（2015）ベイズ統計を用いた PM2.5 常時監視データの解析．大気環境学会誌，50(2)：107－116.
5) 久恒邦裕（2013）ベイズ統計を用いた PM2.5 重量濃度に与える風向の影響の解析．名古屋市環境科学調査センター年報，2：27－33.
6) 久恒邦裕・山神真紀子（2016）相関関係を考慮したパラメータ推定に基づく PM2.5 成分の年平均値推定．名古屋市環境科学調査センター年報，5：24－27.
7) 国立環境研究所. 大気環境時間値データのダウンロード（https://www.nies.go.jp/igreen/tj_down.html）．2020 年 8 月アクセス．
8) 光化学オキシダント調査検討会（2017）光化学オキシダント調査検討会報告書．（http://www.env.go.jp/air/osen/oxidant/report-201703p.pdf）．2020 年 9 月アクセス．
9) 比嘉良作・友寄喜貴・城間朝彰（2016）沖縄県における光化学オキシダントの観測結果．沖縄県衛生環境研究所報，50：92－95.
10) 中妻照雄（2019）Python によるベイズ統計学入門．朝倉書店，pp214.
11) 嘉手納恒・与儀和夫・友寄喜貴・渡具知美希子（2006）沖縄県における光化学オキシダントの現況と傾向．沖縄県衛生環境研究所報，40：99－102.
12) 渡辺澄夫（2012）ベイズ統計の理論と方法．コロナ社，pp226.
13) 浜田宏・石田淳・清水裕士（2019）社会科学のためのベイズ統計モデリング．朝倉書店，pp223.

Ⅵ 付録

データ整形と MCMC 実行のコードを掲載する．

コード 1．解析に使用するデータを整形するコード（1/2）.

```
# 使用するライブラリをインポート
import glob
import numpy as np
import pandas as pd

# 使用するデータは環境データベースの時間値データを使用
# https://www.nies.go.jp/igreen/tj_down.html
# 都道府県，年度毎に zip ファイルで取得でき，個々のファイルの概要は以下の通り
#  j**YYYY_XX.txt
#   **は都道府県コード(01-47)
#   YYYY は西暦年（年度）
#   XX は測定項目番号(01-47)
#   測定項目番号：
#   01:SO2, 02:NO, 03:NO2, 04:NOX, 05:CO, 06:OX, 07:NMHC,
#   08:CH4, 09:THC(THCP,THCM), 10:SPM(SPMB,SPMP), 11:SP,
#   12:PM2.5, 21:WD, 22:WS, 23:TEMP, 24:HUM, 25:SUN, 26:RAIN,
#   27:UV, 28:PRS, 29:NETR, 31:CAR, 41:CO2, 42:O3, 43:HCL,
#   44:HF, 45:H2S, 46:SHC, 47:UHC
# 測定局コードはそらまめくんから参照
# http://soramame.taiki.go.jp/MstItiran.php

# データフレーム作成関数の定義
def make_dataframe(pref, item, stcode):
    '''
    指定したフォルダにある全ての年度ファイルから指定した項目をデータフレームに抽出する関数
    [入力]
    pref:都道府県コード．文字列で指定．例： '01'
    item:測定項目コード．文字列で指定．例： '06'
    stcode:測定局コード．整数値で指定．例： 47211050
    [出力]
    df:コードで指定されたデータの入ったデータフレーム
    '''

    # 都道府県のリスト
    pref_list = {
    '01':'01 北海道','02':'02 青森','03':'03 岩手','04':'04 宮城','05':'05 秋田','06':'06 山形',
    '07':'07 福島','08':'08 茨城','09':'09 栃木','10':'10 群馬','11':'11 埼玉','12':'12 千葉',
    '13':'13 東京都','14':'14 神奈川','15':'15 新潟','16':'16 富山','17':'17 石川','18':'18 福井',
    '19':'19 山梨','20':'20 長野','21':'21 岐阜','22':'22 静岡','23':'23 愛知','24':'24 三重',
    '25':'25 滋賀','26':'26 京都','27':'27 大阪','28':'28 兵庫','29':'29 奈良','30':'30 和歌山',
    '31':'31 鳥取','32':'32 島根','33':'33 岡山','34':'34 広島','35':'35 山口','36':'36 徳島',
    '37':'37 香川','38':'38 愛媛','39':'39 高知','40':'40 福岡','41':'41 佐賀','42':'42 長崎',
    '43':'43 熊本','44':'44 大分','45':'45 宮崎','46':'46 鹿児島','47':'47 沖縄'}

    # 測定項目のリスト
    item_list = {
    '01':'SO2','02':'NO','03':'NO2','04':'NOX','05':'CO','06':'OX','07':'NMHC',
    '08':'CH4','09':'THC(THCP,THCM)','10':'SPM(SPMB,SPMP)','11':'SP',
    '12':'PM2.5','21':'WD','22':'WS','23':'TEMP','24':'HUM','25':'SUN','26':'RAIN',
    '27':'UV','28':'PRS','29':'NETR','31':'CAR','41':'CO2','42':'O3','43':'HCL',
    '44':'HF','45':'H2S','46':'SHC','47':'UHC'}

    # 例えば作業フォルダの階層に"47 沖縄"のフォルダがあり，
    # その下に"西暦年（年度）"のフォルダ，"測定項目番号"毎のテキストファイルがあると想定
    # このコードでは同一測定項目のテキストファイル一覧を読み込んでいる
    # ただし，ただし，西暦年（年度）を抽出する機能は持たせていないため，
    # 必要なファイルのみフォルダに入れる必要がある．
    files = glob.glob('{}¥*¥*_{}.txt'.format(pref_list[pref], item))
    list = []
    for file in files:
        # SHIFT JIS 形式のため，エンコード'cp932'を指定
        list.append(pd.read_csv(file, encoding='cp932'))
    df = pd.concat(list, sort=False)
    df.rename(columns={'測定年度':'fyear', '測定局コード':'stcode',
                       '測定月':'month', '測定日':'day'}, inplace=True)
    df.rename(columns={'01h':1, '02h':2, '03h':3, '04h':4, '05h':5, '06h':6,
                       '07h':7, '08h':8, '09h':9, '10h':10, '11h':11, '12h':12,
                       '13h':13, '14h':14, '15h':15, '16h':16, '17h':17, '18h':18,
                       '19h':19, '20h':20, '21h':21, '22h':22, '23h':23, '24h':24}, inplace=True)
    # 測定局コードを指定して抽出
    df.query('stcode == {}'.format(stcode), inplace=True)
    df.insert(5,'year', df.fyear.where(df.month > 3, df.fyear +1))
    df = pd.melt(df, id_vars=df.columns.values[:8],var_name='time',
                value_name=item_list[item]).sort_values(['year', 'month', 'day'])
    df = df.astype({'time':'int64'})
    return df
```

コード 2．解析に使用するデータを整形するコード（2/2）.

```
# データフレーム処理関数の定義
def calc_po(OX, NO2, NOX):
    '''
    3 つのデータフレームをまとめポテンシャルオゾンを算出
    [入力]
    OX:オキシダントのデータフレーム
    NO2:二酸化窒素のデータフレーム
    NOX:窒素酸化物のデータフレーム
    [出力]
    PO:ポテンシャルオゾンのデータが入ったデータフレーム
    '''

    # OX のデータフレームに NO2, NOX から抽出した列を結合
    PO = pd.concat([OX, NO2.NO2, NOX.NOX], axis=1)
    # 9998, 9999 などの欠測コードを欠測値 nan に変換
    PO = PO.replace([9997, 9998, 9999], np.nan)
    # ポテンシャルオゾンを計算し列 PO を追加
    PO = PO.assign(PO = PO.OX + PO.NO2 - 0.1 * PO.NOX)
    # 欠測値が含まれる行を削除
    PO = PO.dropna()
    return PO

# 沖縄局(47211050)のデータ作成
# OX ファイルからデータフレームを作成
df = make_dataframe(pref='47', item='06', stcode=47211050)

# NO2 ファイルからデータフレームを作成
df2 = make_dataframe(pref='47', item='03', stcode=47211050)

# NOX ファイルからデータフレームを作成
df3 = make_dataframe(pref='47', item='04', stcode=47211050)

# ポテンシャルオゾンのデータフレームを作成
df = calc_po(df, df2, df3)

# 平均・標準偏差を計算してデータを正規化し列を追加
mean = df.loc[:, 'PO'].mean()
std = df.loc[:, 'PO'].std()
df = df.assign(PO_n = (df.PO - mean) / std)

# csv 形式で必要な列のみ保存
df.to_csv('okinawa_PO.csv', columns=['year', 'month', 'day', 'time', 'PO', 'PO_n'],
          encoding='cp932', index=False)

# 与那城局(47322010)のデータ作成
df = make_dataframe(pref='47', item='06', stcode=47322010)
df2 = make_dataframe(pref='47', item='03', stcode=47322010)
df3 = make_dataframe(pref='47', item='04', stcode=47322010)
df = calc_po(df, df2, df3)

# 平均・標準偏差を計算してデータを正規化し列を追加
mean = df.loc[:, 'PO'].mean()
std = df.loc[:, 'PO'].std()
df = df.assign(PO_n = (df.PO - mean) / std)

# csv 形式で必要な列のみ保存
df.to_csv('yonashiro_PO.csv', columns=['year', 'month', 'day', 'time', 'PO', 'PO_n'],
          encoding='cp932', index=False)
```

コード 3．MCMC を実行するコード（沖縄局のモデル 1 のみ掲載）.

```
# このコードを Google Colaboratory 上で実行するには ArviZ の未インストールと
# pandas と PyMC3 およびそのバックエンドで動く ArviZ のバージョンが対応していないことが問題となるので
# !pip install pandas==0.24.2
# !pip install arviz==0.5.1
# とコード実行前にバージョンを指定してライブラリをインストールしておく必要がある.

# 使用するライブラリをインポート
import numpy as np
import pandas as pd
import theano.tensor as tt
import pymc3 as pm

# データの読み込み
df = pd.read_csv('okinawa_PO.csv', encoding='cp932')

# データを月と時間でリストにまとめる
Omh = []
for i in range(1, 13):
    sublist = []
    for j in range(1, 25):
        sublist.append(df.query('month == @i & time == @j')['PO_n'].values.tolist())
    Omh.append(sublist)

# パラメータの次元を設定
m = 12
h = 24

# モデルを定義
model = pm.Model()
with model:
    Am = pm.Uniform('Am', -30, 30, shape = m)
    pre_Bm = pm.Uniform('pre_Bm', 0, 10, shape = m-1)
    Bm = pm.Deterministic('Bm', tt.concatenate([tt.ones(1), pre_Bm]))
    pre_Dh = pm.Uniform('pre_Dh', 0, 30, shape = h-1)
    Dh = pm.Deterministic('Dh', tt.concatenate([pre_Dh[:5], tt.zeros(1), pre_Dh[5:]]))
    sigma = pm.Uniform('sigma', 0, 10)
    Cmh = pm.Deterministic('Cmh', Am[:, None] + Bm[:, None] * Dh[None, :])
    likelihood = [pm.Normal('O_{}_{}'.format(k, n), mu=Cmh[k][n],
                            sd=sigma, observed=Omh[k][n]) for k in range(m) for n in range(h)]

# MCMC の実行
n_draws = 30000
n_chains = 3
n_tune = 5000
with model:
    trace = pm.sample(draws=n_draws, chains=n_chains, tune=n_tune, nuts_kwargs=dict(target_accept=0.9))
print(pm.summary(trace[::10])) # trace[::10]は 10 点毎のサンプリングのための間引き操作

# 標準モジュール pickle をインポート
import pickle
# 後からでも作業できるように実行結果を保存
# モデルの保存
with open ('okinawa_PO_model1.pkl', 'wb',) as f:
    pickle.dump(model, f)
# 10 点置きに間引きした MCMC サンプルの保存
with open ('okinawa_PO_model1_trace.pkl', 'wb',) as f:
    pickle.dump(trace[::10], f)

# PyMC3 の summary 関数には中央値の出力機能がないので関数を作成
def trace_quantiles_50(x):
    return pd.DataFrame(pm.quantiles(x, [50]))

# summary 関数の出力結果を csv 形式で保存
summary = pm.summary(trace[::10])
median = pm.summary(trace, stat_funcs=[trace_quantiles_50])
summary = median.join(summary)
summary = summary.rename(columns={50:'median'})
summary.to_csv('okinawa_model1_summary.csv')

# 正規化の逆処理の準備
mean = df.loc[:, 'PO'].mean()
std = df.loc[:, 'PO'].std()

# 正規化と逆の処理を行い，処理後の MCMC サンプルを csv で保存
df2 = pm.trace_to_dataframe(trace[::10])
am = df2.iloc[:,0:12] * std + mean
bm = df2.iloc[:,12:35]
dh = df2.iloc[:,35:82] * std
s = df2.iloc[:,82:83] * std
cmh = df2.iloc[:,83:371] * std + mean
df3 = am.join([bm, dh, s, cmh])
df3.to_csv('okinawa_PO_model1_trace.csv')
```