

MMMMMedia Mix Model Modified

■ メディアミックスモデルの利点と限界

Katagiri, Satoshi (@ill-identified)

公開日: 2021/5/29 (Tokyo.R #92)

前置き

自己紹介

- ただの素人です
- twitter ID: @ill-identified
- ブログ
- 正直最近は stan あまり使っていない

宣伝

1. 『[R Markdown クックブック](#)』を翻訳した
 - R Markdown の活用事例集です
 - “[Bookdown](#)” や “[Definitive Guide](#)” もそのうち翻訳したい
 - “[knitr](#) のドキュメント” も翻訳しています
2. このスライドは [rmdja](#) パッケージで作成しました
 - 意外と煩雑な **rmarkdown** での日本語文書の設定をなるべく自動化することを目的にしたパッケージ
3. エッ! まだ R のグラフの文字化けで消耗してるって!?
 - [ragg](#) や [fontregisterer](#) を使うとよいことですよ?
 - 参考 [1](#), [2](#)

今日話すこと

1. Jin, Wang, et al. [5] “Bayesian methods for media mix modeling with carryover and shape effects” の解説
2. 実装と具体的な使い方
3. レポートの問題点と改善提案
 - 回帰分析の基本的な理解があったほうが良い
 - 詳しくない人は中盤の技術的な箇所は適当に聞き流してください
 - マーケティング関係の理解があったほうが良い
 - 逆に私はあまりない (言い訳)
 - stan のヘビーユーザーには物足りないかも
 - (言い訳その 2)

Media Mix Model とは

メディア・ミックス・モデル (MMM)

あなたの会社は複数のチャネル (広告媒体) に支出しプロモーションを行ってきました

では, これらの広告費は回収できているのでしょうか?

単純なメディア・ミックス・モデル

- 毎日の費用と売上で回帰分析すればいいのでは?
 - M, T : 支出したチャネルの数と期間
 - y_t : 日付 t の売上
 - $x_{m,t}$: 日付 t , メディア m への支出額
- 回帰係数 β_m が広告の限界効果!!

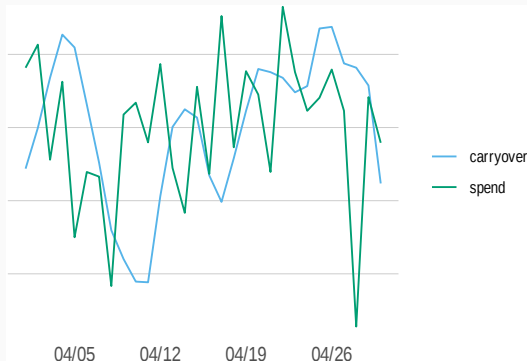
$$y_t = \beta_1 x_{1,t} + \beta_2 x_{2,t} + \dots + \beta_M x_{M,t}$$

指摘されている問題点

- [5] の言いたいこと: 「単純に同時点で回帰しただけではダメでしょ」
- 経験的に, 以下の 2 点で問題
 - 繰り越し (carryover) 効果がない
 - 飽和 (satuation) 点がない

繰り越し効果

- プロモーションの結果は少し遅れて現れる可能性
- スポットより数日間のまとまった支出のほうが効果ありそう



adstock 関数

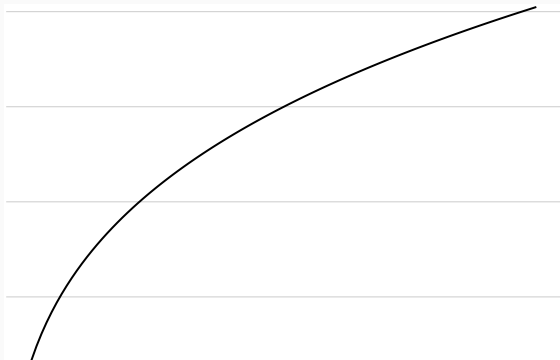
- x_t を直近 L 期間の重み付き移動平均で変換する

$$\text{adstock}(x_t; w, L) := \frac{w_m(L)x_t}{\sum w_m(1)}$$

- $w_m(L)$ の L までの各重みは非負
- 重み w の付け方は任意, 今回は幾何減衰を採用
 - $w_m^g(l; \alpha_m) := \alpha_m^l$
 - これをさらに変形して $w_m^d(l; \alpha_m, \theta_m) := \alpha_m^{(l-\theta_m)^2}$ にする
 - α_m : 持続率
 - (機械学習やノンパラメトリックモデルで使われる動経基底カーネルと同じ)

飽和点の問題

- いわゆる「サチる」
- 単純な回帰分析では「やればやるだけリターン」になる
- 非現実的ではないか？



hill 関数の導入

- 支出の増大に対して飽和点のある形状へ変換

$$\text{Hill}(x_{t,m}; K_m, S_m) := \frac{1}{1 + (x_{t,m}/K_m)^{-S_m}}$$

- メディアによって飽和点が違うと仮定
- $S_m > 0$: 傾き
- $K_m > 0$: 飽和のタイミング
- さらにスケールパラメータ β_m を掛ける

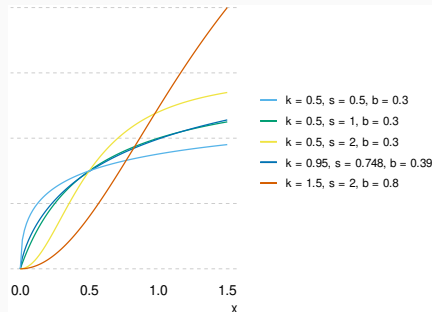


図 1: hill 関数の例

最終的にこういうモデルになる

- $x_{t,m}^*$ は adstock で変換した $x_{t,m}$
- adstock \rightarrow hill と hill \rightarrow adstock どちらが適切かは断言できない
- 2 つ目の $\sum \dots$ はその他の説明変数 (商品の価格とか)
- 売上額 y_t は対数変換したほうが当てはまりが良い
 - 誤差項 ε_t を正規分布とするため

$$\log(y_t) = \tau + \sum_m \beta_m \text{Hill}(x_{t,m}^*; K_m, S_m) + \sum_{c=1}^C \gamma_c z_{t,c} + \varepsilon_t$$

あれ? じゃあ β でリターンを計算できない?

広告支出に対するリターン

- Return on Ad Spend (ROAS) と marginal ROAS (mROAS) が計算可能

$$\text{ROAS}_m = \frac{\sum_{[t_0, t_1+L-1]} (\hat{Y}(x_{t,m}) - \hat{Y}(\underline{x}_{t,m}))}{\sum_{[t_0, t_1]} x_{t,m}}$$
$$\text{mROAS}_m = \frac{\sum_{[t_0, t_1+L-1]} (\hat{Y}(\tilde{x}_{t,m}) - \hat{Y}(x_{t,m}))}{\sum_{[t_0, t_1]} x_{t,m}}$$

- $\underline{x}_{t,m}$ は比較レベル, 通常はゼロ
 - 「**もし**広告費を出さなかったら」との比較
- $\tilde{x}_{t,m}$ は実際の値から $a\%$ 増加させた額
 - 「**もし**広告費を $a\%$ 増やし (減らし) たら」

Rでの実装

しばらく技術的に細かい話が続きます

- 著者が Stan コードを添付している
 - 構文が古い & 不要な記述が残っている
 - 最新版の機能も活用して**修正**

先行事例

- 注: 晒しものにしたいという意図はないです
- 面白いフレームワークが活用されずに埋もれるのが MOTTAINAI
- ushi-goroshi.hatenablog.com

最適化の計算が怪しい

- tjo.hatenablog.com

乱数データの生成部分が変 (X_{media} 一部しか代入してない), 最終結果の掲載が不完全

- sibylhe/mmm_stan (towardsdatascience.com, これのみ pystan)

元のモデルの簡略バージョンの結果しか検証していない

R の環境設定

- R 4.0.5 (4.1 **ではない**)
- rstan [3] は 2.21
- 必要な関数の多くはパッケージにまとめた
 - リンク先の指示に従いインストール
- チュートリアルは doc/main.nb.html を参考に
- リポジトリは **renv** [7] で設定してあるので**たぶん**再現できる
 - stan は OS/外部ライブラリ依存が強いので保証はできない
 - あと乱数だし結果の完全一致は難しい
 - 動かない等のバグ報告は歓迎

最近の rstan の注意

- なんか RStudio v1.4 だとよくクラッシュする?([参考](#))
 - `rstan_options(javascript = F)` が有効説
 - `autowrite = T` を諦めるしかない説
- あるいは **cmdStanR** を選ぶ?
 - 軽量な Stan インターフェイス
 - ただし **rstanarm**, **brms** は **rstan** 依存

原文の提案

1. 2つの効果を考えた複雑なモデルにする
2. モデルの係数を計算する
3. 係数から広告投資リターン計算
4. x を変数に最適化問題を解く
 - 最適配分がわかる

修正内容

- 著者のコードを以下の点で修正
 1. “<-” 代入演算子を “=” へ
 - 将来廃止される予定のため
 2. 不要な変数の削除
 3. バグトラッキングしやすくする
 4. (計算の効率化は時間が足りなかった)
 5. ROAS/mROAS の計算処理を追加

Stan (MCMC) にありがちなこと

- よくわからないエラーが大量に出る
 - コンパイル通ったのに実行したらエラーが出る
 - どこが原因なのか分かりづらい
- 計算終わったけど収束しない
- この辺の問題をある程度解消する

Stan のバグトラッキング

- nan になったとか無限大になったとかでエラーが出る
- (個人的な経験則) 計算時のエラーはだいたい
 1. 初期値や事前分布の値域が不適切
 2. データの値が不適切
- `print, reject` を使う (7.12)
- 変数に定義域を与える

```
row_vector<lower=0>[max_lag] X_media[N, num_media];
```

Stan のバグトラッキング: 具体的な修正

- 支出 x は負値にならない
 - 原著には書かれてないが, `shape` 関数の性質から
 - `<lower=0>` を設定
- 傾きパラメータ S (slope) や K (半飽和点) も負値にならない
 - `reject` 文を使う

```
if (slope[media] < 0) reject("slope[, media, "] must be positive; four
```

- サンプルングを `reject` するだけで中断はしない
- 対話的に修正できないので「かもしれないコーディング」を心がける

Stan のバグトラッキング

- `vb()` で変分ベイズ法で計算
 - MCMC と比べて非常に早い
 - コーディングというより運用上のテクニック
- stan 公式「あくまで近似なので MCMC でも収束するようにしてね」
- MCMC の予行演習みたいな位置づけ
 - 変分法でうまく行かなかったら MCMC でも多分失敗

動作確認

- 「都合のいい」乱数データで確認
 - ちょうどいいデータが見つからなかったため
- 実際の生データでありそうな構造のデータ -> stan 入力に変換
 - **tidyverse** [9] を駆使して変換

データの構造

- こんな構造のデータがあるという前提
 - media-* はチャネルごとの支出
 - price は共通の説明変数
 - sales は売上額

week	sales	price	media-1	media-2	media-3
2020-05-13	141.5514	2.9480380	1.7426992	0.3887749	0.1834057
2020-05-20	306.5943	2.6773373	0.0940272	0.9092068	0.3765928
2020-05-27	478.9258	3.4306232	3.1730597	0.2946631	4.0993204
2020-06-03	392.5824	3.0419810	0.0295556	0.3375569	1.5000075
2020-06-10	227.0002	0.9962776	1.4754346	0.5900986	0.0322262
2020-06-17	140.4724	-0.0503101	3.5917583	0.2150804	0.3221931 ²⁶

stan の計算実行

```
# コンパイル
model <- stan_model(
  file = "hoge hoge.stan",
  model_name = "original", verbose = T
)
# 変分法
res_vb <- vb(
  object = model_origin,
  iter = 20000, data = to_stan_data(df_out, L = 13)
)
# NUTS
res <- sampling(model, data = to_stan_data(df_out, L = 13))
```

MCMC の事後診断

- 「エラーが発生しなかった! 成功!」 **そんなことはない**
- MCMC の収束確認が必要

事後診断には bayesplot パッケージ

- 昔の参考書 [10] や私の昔のブログでは **ggmcmc** [1] を使用
- 今は **bayesplot** [2] のほうが使いやすい
 - 構文が簡単 (使用例一覧)
 - ランク正規化プロット [8]などをサポート

要点: \hat{R} だけで判定は BAD

- 従来は \hat{R} だけ見て 1.05 未満なら OK
- ランク正規化/畳み込み \hat{R} も要確認
- どれかが大きいと収束してない可能性
- トレースプロットもなるべく見る
- 詳細は Vehtari, Gelman, et al. [8], **bayesplot** のドキュメント
 - または 私の解説

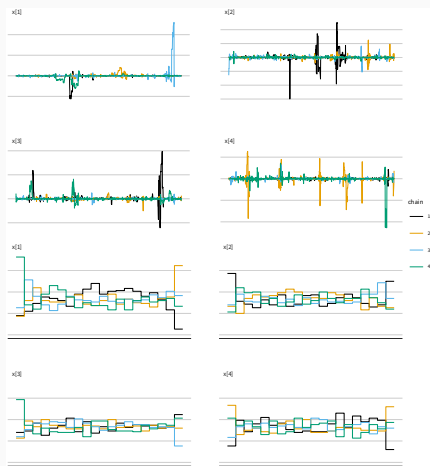


図 2: トレースプロットとランク正規化ヒストグラム

bayesplot のコード

- `res = stanfit` オブジェクト
- `regex_pars` は正規表現マッチでパラメータ取得
- `Rhat()` は 3 種類の \hat{R} の最大値を返す
 - しかしなぜか `stanfit` に対応していない

```
require(bayesplot)
mcmc_trace(res, pars = "tau")
mcmc_rank_overlay(res, regex_pars = "mu")
mcmc_rhat(
  apply(rstan::extract(res, pars = "mu", permuted = F, inc_warmup
    ↪ = T), 3, Rhat)
) + yaxis_text(hjust = 1)
```

\hat{R} の表示

- 3 種類の \hat{R} のうち最大の値を使用

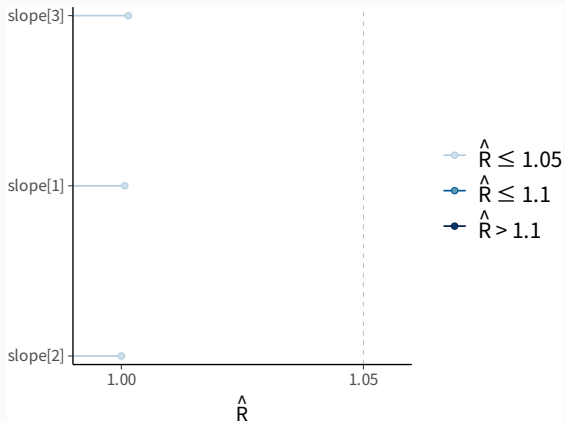


図 3: S_m の \hat{R}

ランク正規化プロット

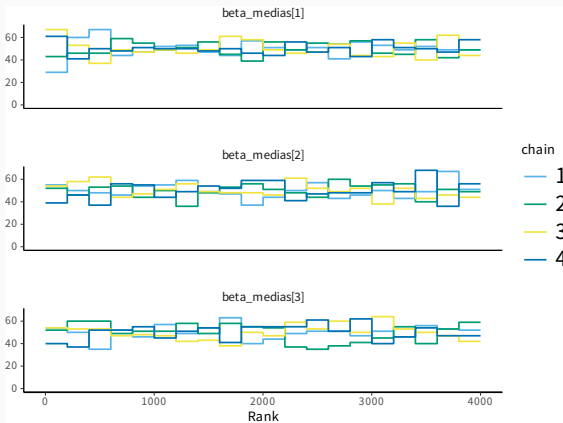


図 4: β_m のランク正規化ヒストグラム

計算結果

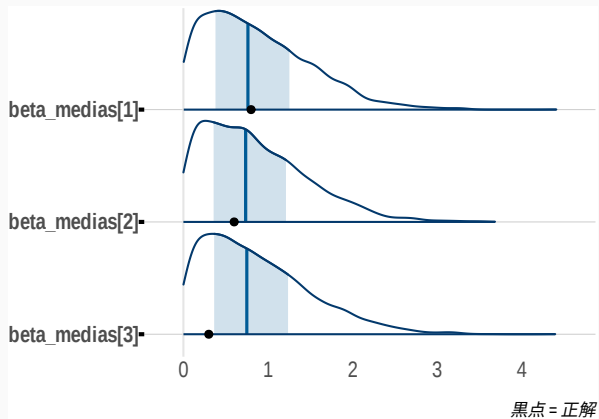


図 5: β_m のエリアプロット

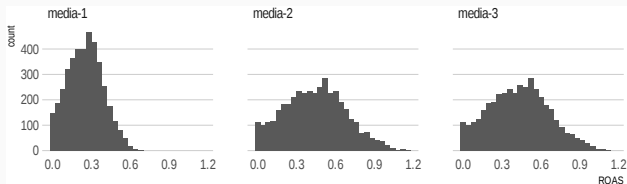
(m)ROAS の計算

- R で計算関数を作成
 - stan 内でも可能だが使いづらい
- ROAS(), mROAS() を用意
 - 現時点ではあまりスマートな設計でない...
 - L の手動入力
 - **とても遅い**

```
roas <- ROAS(res, df_out, L = 13)
mroas <- mROAS(res, df_out, L = 13)
```

Error in newdata\$X_media[nn, m,]: 添え字が許される範囲外です

結果



Error in `ggplot(mroas, aes(x = ROAS, group = media))`: オブジェクト 'mroas'

より大きな問題点

ここまではプログラム実装上の問題. マイクロマネジメント的な話
以降は実用する上でより重大な問題点に触れる

上の方ほど深刻な問題点

1. この方法で最適配分は分からない
 - いわゆる因果推論ではない
2. メディア「ミックス」できてない
 - IID を前提にしている
3. データが ARIMA (非定常) ではないように見える

最適配分を知ることは可能か

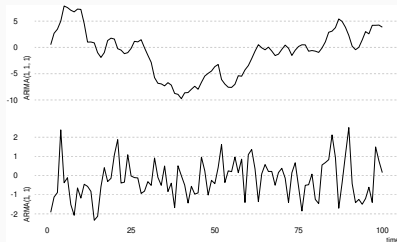
- 「推定したパラメータを固定して $\min_x \hat{Y}(x)$ から最適な予算配分 x を求める」としている
 - データにない $x_{m,t}$ を与えて売上の予測値を見ている (= 外挿)
 - 外挿がどの程度妥当かの評価は難しい
- 因果推論の観点からはかなり怪しい
 - 観察データのみで結論するのはかなり厳しい
 - 参考: 最近邦訳のでた Rosenbaum [6]
- 他の人もあまりこの方法には触れていない...

メディア「ミックス」していない

- メディア間のシナジー効果はないと仮定
 - IID の仮定につながる
 - 似たような問題意識を持っている人はいる (参考)
- モデルはあくまで近似, 目的に応じて使用
 - シナジー効果の測定に興味があるか
 - 無視したシナジー効果の実際の影響がどの程度か

データが非定常かどうか

- マーケティングでは非定常な時系列データを扱うことも多そう
 - Hanssens, Parsons, and Schultz [4] なんかを読んだ感想
- 原著では乱数を ARIMA で生成したとある.
 - しかし I がゼロ次のデータに見える
 - ランダムウォークはこうはならない
- 目視は厳密ではないが元データ非公開



使用上の注意についての要約

1. 最適配分を求める際はデータをどう取得したかに注意
 - 観察データではなく実験データを使う
 - 測定期間中に予算配分を操作しないとか
2. シナジー効果がないような状況が必要
 - 「メディア」「チャネル」の定義に注意
3. データが非定常な場合はどうなるか未検証

その他の言い訳

- 本当は後半の「より大きな問題点」のほうが優先すべきだったかも
 - しかし完成させるには時間が足りない
- モデルの当てはまりの評価もできるようにしたかった
 - BIC と bridge/path sampling ([Vehtari のコメント](#)) の計算処理とか
- 応用事例や類似モデルのサーベイもしたかった
 - Hill/Shape 変換なしの比較とか
- いろいろな乱数データで当てはまりを確認したかった
 - 都合の良いデータだけならなんとでも言える
 - どの程度ロバストなモデルかは応用で重要
- 今回上げた MMM の問題点・課題を解消する具体的方法まで考えたかった...

参考文献

- [1] Fernández-i-Marín, Xavier (2016). “ggmcmc: Analysis of MCMC Samples and Bayesian Inference”. In: Journal of Statistical Software 70.9, pp. 1–20. DOI: [10.18637/jss.v070.i09](https://doi.org/10.18637/jss.v070.i09).
- [2] Gabry, Jonah, Tristan Mahr (2021). ”bayesplot: Plotting for Bayesian Models”. R package version 1.8.0. URL: [here](#).
- [3] Guo, Jiqiang, Jonah Gabry, Ben Goodrich, Sebastian Weber (2020). ”rstan: R Interface to Stan”. R package version 2.21.2. URL: [here](#).
- [4] Hanssens, Dominique M. Leonard J. Parsons, Randall L. Schultz (2002). ”Market Response Models: Econometric and Time Series Analysis”. Kluwer Academic Publishers. ISBN: 978-0-7923-7826-6 DOI: [10.1007/b109775](https://doi.org/10.1007/b109775). URL: [here](#) (visited on 05/14/2021). 阿部誠・パワーズ恵子訳『2018』有斐閣
- [5] Jin, Yuxue, Yueqing Wang, Yunting Sun, David Chan, Jim Koehler (2017). ”Bayesian Methods for Media Mix Modeling with Carryover and Shape Effects”. Google Inc. URL: [here](#).

- [6] Rosenbaum, Paul R. (2017). "Observation and Experiment: An Introduction to Causal Inference". Harvard University Press. ISBN: 978-0-674-97557-6. 阿部貴行・岩崎学訳『統計的因果推論入門』. 2021 年. 共立出版
- . [7] Ushey, Kevin (2021). "renv: Project Environments". R package version 0.13.2. URL: [here](#).
- [8] Vehtari, Aki, Andrew Gelman, Daniel Simpson, Bob Carpenter, Paul-Christian Bürkner (2021). "Rank-Normalization, Folding, and Localization: An Improved \widehat{R} for Assessing Convergence of MCMC". In: Bayesian Analysis. ISSN: 1936-0975. DOI: [10.1214/20-BA1221](#). URL: [here](#) (visited on 04/07/2021).
- [9] Wickham, Hadley (2021). "tidyverse: Easily Install and Load the Tidyverse". R package version 1.3.1. URL: [here](#).
- [10] 松浦, 健太郎 (2016). "Stan と R でベイズ統計モデリング". 共立出版