シンプルな数理モデルで
ビジネス課題を解決する

2021/12/4
Japan.R 2021
森下光之助（@dropout009）

自己紹介

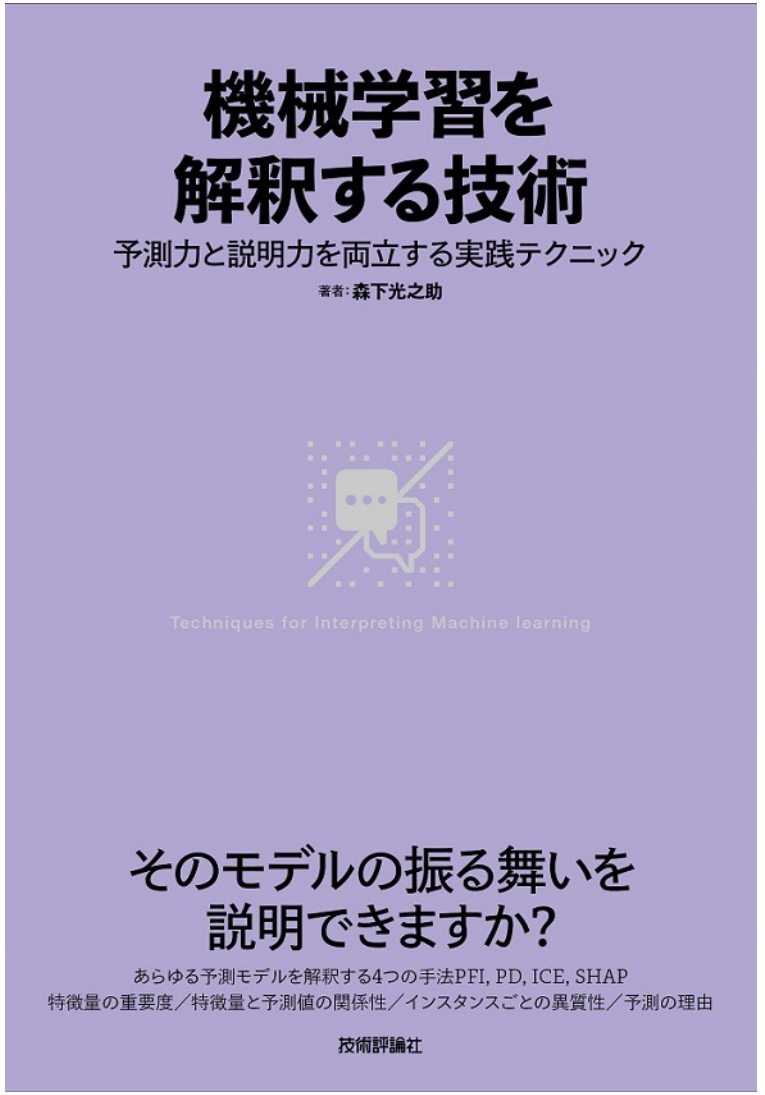
森下光之助

TVISION INSIGHTS株式会社
データサイエンティスト
執行役員（データ・テクノロジー担当）

テレビの視聴行動を分析しています

データの利活用、マネジメント、組織づくり、
因果推論、機械学習の解釈手法などに興味があ
ります

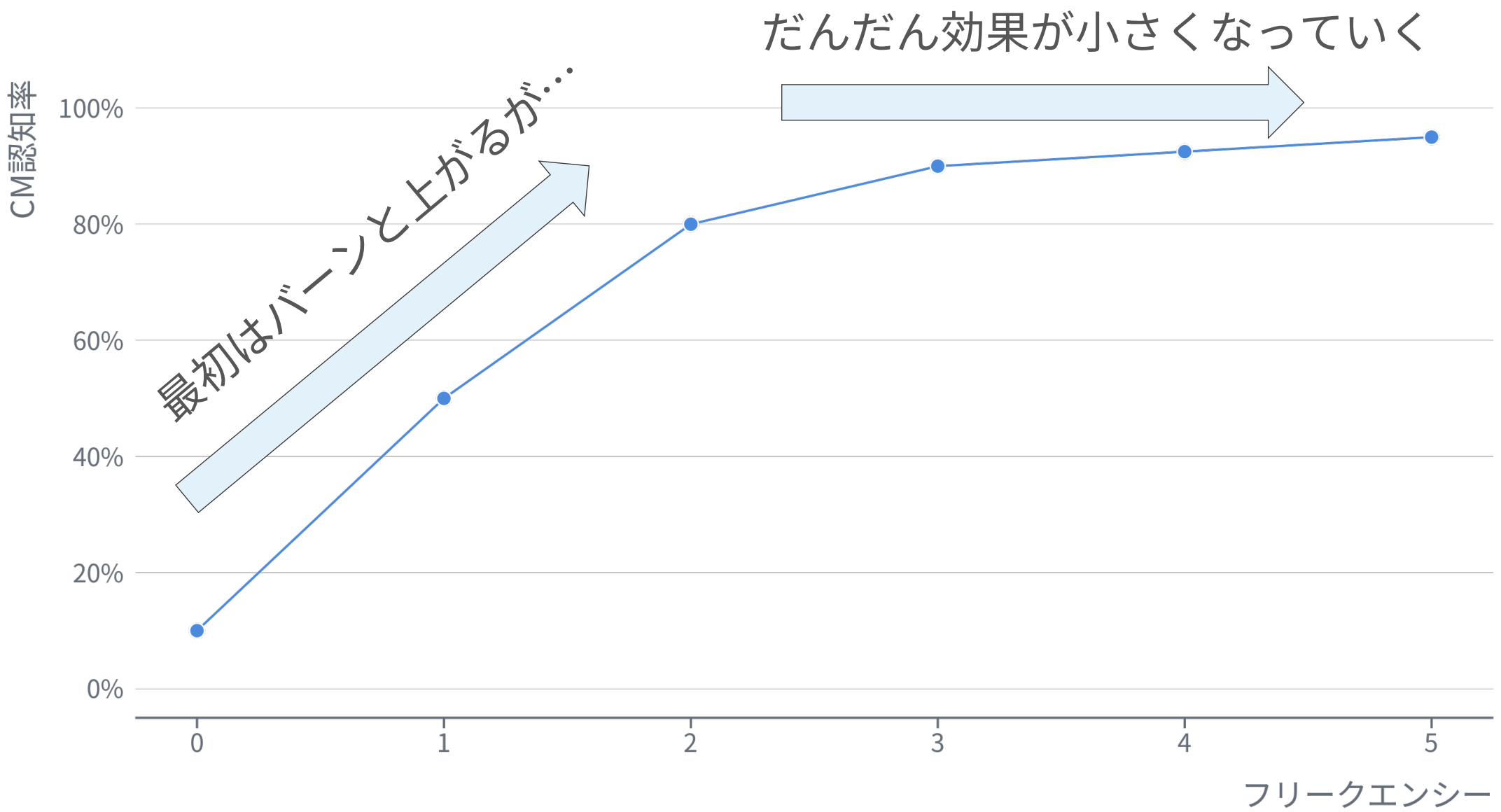
Twitter: @dropout009
Speaker Deck: dropout009
Blog: https://dropout009.hatenablog.com/



知りたいこと

CMって何回くらい
見てもらえたらいいの？

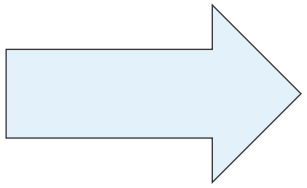
やりたいこと：CMの無駄打ちを避けたい



シングルソースでデータがとれている場合

同じ個人に対してフリークエンシーと認知がとれているなら単に集計すればいい

個人	フリークエンシー	認知してるか
森下	0	してない
伊藤	1	してない
中村	2	してる
宮本	2	してる
…	…	…



フリークエンシー	認知率
0	20%
1	50%
2	70%
3	80%
…	…

実務ではシングルソースでデータが手に入らないケースも多い

TVISION INSIGHTSはテレビの視聴行動は自動でとれるが、CM認知はとれていない

なので、シングルソースでデータを揃えるためには、パネルさんにCM認知のアンケートをとる必要があるが…

- 過去のキャンペーンに対して分析を行いたい場合、これからアンケートをとっても間に合わない
- アンケートでCM認知をとりすぎるとパネルさんがCMを意識するようになる懸念がある

実務ではマルチソースになっていることが多い

同じ個人に対してデータがとれていなくても、
フリークエンシーとCM認知率の関係を推定できるような手法を考えたい

個人	フリークエンシー
森下	0
伊藤	1
中村	2
宮本	2
…	…

別のソース

個人	認知してるか
松本	してない
慶田	してない
三野	してる
萩原	してる
…	…

簡単なモデルを考える

CMの認知率を数式を使って考えていく

CM認知率は以下のように表現できる

$$a = \frac{1}{N}\sum_{i=1}^{N} A_i$$

- i：個人を表す添字（$i = 1, \ldots, N$）
- A_i：個人iがCMを認知していれば1、していなければ0をとる確率変数
- a：CMを認知している人の割合

CM認知率の期待値を変形していくと…

$$\mathbb{E}[a] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} A_i\right]$$

$$= \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[A_i]$$

$$= \frac{1}{N}\sum_{i=1}^{N} \Pr(A_i = 1)$$

$$= \frac{1}{N}\sum_{i=1}^{N}\sum_{f=0}^{\bar{F}} \Pr(A_i = 1 \mid F_i = f)\Pr(F_i = f)$$

個人iがCMにf回接触したときにCMを認知する確率

個人iがCMにf回接触する確率

ここで、F_iは個人iのCM接触回数（$F_i = 1, \ldots, \bar{F}$）

個人差がないことを仮定する

CMにf回接触したときにCMを認知する確率と、CMにf回接触する確率が個人によらず一定であることを仮定すると、CM認知度の期待値は以下の形に変形できる

$$\mathbb{E}[a] = \sum_{f=0}^{\bar{F}} \Pr(A = 1 \mid F = f) \Pr(F = f)$$

- 平均的な関係に注目したいので悪くない仮定だと思う
  というか、マルチソースで個人差を考えることは難しい
- セグメントごとのデータがあるならセグメントで条件づけることでより妥当な分析が可能（$\mathbb{E}[a \mid S = s] = \sum_{f=0}^{\bar{F}} \Pr(A = 1 \mid F = f, S = s) \Pr(F = f \mid S = s)$）

CM認知率の推定モデル

あるCMに対して、認知度調査をT回実施しているとすると、それにあわせて
テレビ視聴データを集計して、以下のような回帰モデルを考えることができる

$$a_t = \sum_{f=0}^{\bar{F}} \beta_f s_{f,t} + \epsilon_t$$

- t：時点を表す添字（$t = 1, \ldots, T$）
- a_t：時点tでのCM認知率
- $s_{f,t}$：時点tでの接触回数がf回の人の割合
- β_f：接触回数がf回の人がCMを認知する確率
- ϵ_t：測定誤差などのノイズ

$$\mathbb{E}[a] = \sum_{f=0}^{\bar{F}} \Pr(A = 1 \mid F = f) \Pr(F = f)$$

データからの推定は制約付きの最小二乗法で可能

$$\min_{\beta_0,\ldots,\beta_{\bar{F}}} \sum_{t=1}^{T}\left(a_t - \sum_{f=0}^{\bar{F}} \beta_f s_{f,t}\right)^2$$
$$\text{s.t. } 0 \le \beta_0 \le \beta_1 \le \cdots \le \beta_{\bar{F}} \le 1$$

- β_fは確率なので、0以上1以下の値をとる制約をいれた
- フリークエンシーが大きくなっていくほどCMを認知する確率は高まっていくことが想定されるので、$\beta_0 \le \beta_1 \le \cdots \le \beta_{\bar{F}}$という単調増加の制約をいれた

関数形を特定して推定の安定化を狙う

認知度調査のデータ数が少ない場合は、CMを認知する確率の関数形を特定することで推定を安定させることができる
（特定化に失敗するとバイアスがかかるのでそこはトレードオフ）

たとえば、以下のような特定化が考えられる

$$\Pr(A = 1 \mid F = f) = 1 - (1 - \theta)(1 - \pi)^f$$

- 誤認率をθとしている
- CMに1回接触した場合にCMを認知する確率をπとしている
  f回接触してなお認知しない確率は$(1 - \pi)^f$となる
- 誤認もCM接触による認知もしない確率を1から引くと認知する確率になる

関数形を特定した場合の回帰モデル

非線形最小二乗法や
ベイズなどで推定可能

$$a_t = \sum_{f=0}^{\bar{F}} \left(1 - (1-\theta)(1-\pi)^f\right) s_{f,t} + \epsilon_t$$

$$= 1 - (1-\theta) \sum_{f=0}^{\bar{F}} (1-\pi)^f s_{f,t} + \epsilon_t$$

- t：時点を表す添字（$t = 1, \ldots, T$）
- a_t：時点tでのCM認知率
- $s_{f,t}$：時点tでの接触回数がf回の人の割合
- θ：誤認率
- π：CMに1回接触した場合にCMを認知する確率
- ϵ_t：測定誤差などのノイズ

$$\mathbb{E}[a] = \sum_{f=0}^{\bar{F}} \Pr(A = 1 \mid F = f) \Pr(F = f)$$

シミュレーションで確かめる

シミュレーションデータの生成

```r
df ← tibble(
  i = 1:n_samples,
  reach_prob = reach_prob_fn(n_samples)
) %>%
  crossing(t = 1:n_times) %>%
  mutate(is_reach = as.double(rbernoulli(n_samples * n_times, p = reach_prob))) %>%
  group_by(i) %>%
  mutate(
    fq = cumsum(is_reach),
    fq = if_else(fq ⩾ max_fq, max_fq, fq)
  ) %>%
  ungroup() %>%
  mutate(
    awareness_prob = awareness_prob_fn(fq),
    is_aware = rbernoulli(n_samples * n_times, p = awareness_prob)
  )

df_awareness ← df %>%
  group_by(t) %>%
  summarise(awareness_prop = mean(is_aware))

df_share ← df %>%
  group_by(t, fq) %>%
  summarise(fq_prop = n() / n_samples) %>%
  pivot_wider(
    id_cols = t,
    names_from = fq,
    values_from = fq_prop,
    values_fill = 0,
    names_prefix = "s_"
  )

df_aggregated ← df_awareness %>%
  left_join(df_share)
```

```
# A tibble: 20 × 8
      t awareness_prop    s_0    s_1    s_2    s_3    s_4    s_5
  <int>          <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1     1          0.446 0.500  0.500  0      0      0      0
2     2          0.639 0.247  0.499  0.254  0      0      0
3     3          0.746 0.122  0.373  0.376  0.128  0      0
4     4          0.814 0.0624 0.249  0.370  0.249  0.0692 0
5     5          0.852 0.0353 0.156  0.303  0.308  0.163  0.0342
```

ノンパラメトリックな推定

```r
# Nonparametric estimation
S ← df_aggregated %>%
  select(starts_with("s")) %>%
  as.matrix()

y ← df_aggregated %>% pull(awareness_prop)

bounds ← function(x) {
  d ← numeric(length(x) - 1)
  for (i in 2:length(x)) {
    d[i - 1] ← x[i] - x[i - 1]
  }
  return(d)
}


solution ← solnp(
  pars = seq(0.1, 0.9, length.out = max_fq + 1),
  fun = function(x) sum((y - S %*% x)^2),
  ineqfun = bounds,
  ineqLB = rep(0, max_fq),
  ineqUB = rep(1, max_fq),
  LB = rep(0, max_fq + 1),
  UB = rep(1, max_fq + 1),
  control = list(tol = 1e-10)
)
```

$$\min_{\beta_0,\ldots,\beta_{\bar{F}}} \sum_{t=1}^{T}\left(a_t - \sum_{f=0}^{\bar{F}} \beta_f s_{f,t}\right)^2$$
$$\text{s.t. } 0 \le \beta_0 \le \beta_1 \le \cdots \le \beta_{\bar{F}} \le 1$$

関数形を特定してNLSで推定

```r
get_nls_formula = function(max_fq) {
  map_chr(0:max_fq, ~ glue("(1 - theta) * (1 - pi)^{.} * s_{.}")) %>%
    c("awareness_prop ~ 1", .) %>%
    glue_collapse(sep = " - ") %>%
    as.formula()
}
```

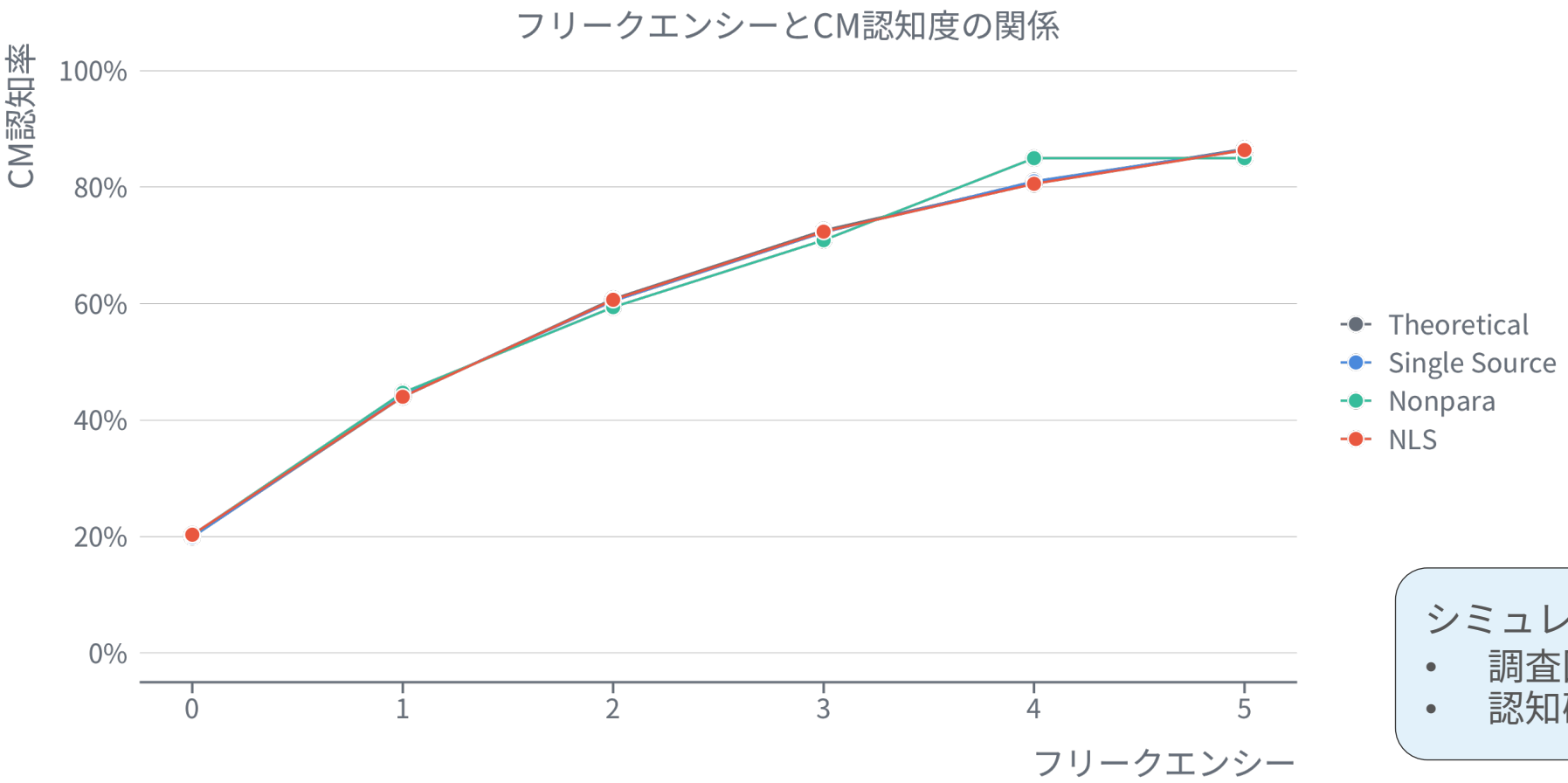
```
awareness_prop ~ 1 - (1 - theta) * (1 - pi)^0 * s_0
                   - (1 - theta) * (1 - pi)^1 * s_1
                   - (1 - theta) * (1 - pi)^2 * s_2
                   - (1 - theta) * (1 - pi)^3 * s_3
                   - (1 - theta) * (1 - pi)^4 * s_4
                   - (1 - theta) * (1 - pi)^5 * s_5
```

```r
model_nls ← nls(
  formula = get_nls_formula(max_fq),
  start = c(theta = 0.1, pi = 0.1),
  lower = c(0, 0),
  upper = c(1, 1),
  algorithm = "port",
  control = list(maxiter = 100, tol = 1e-10),
  data = df_aggregated
)
```

$$a_t = 1 - (1 - \theta) \sum_{f=0}^{\bar{F}} (1 - \pi)^f s_{f,t} + \epsilon_t$$

結果の比較：関数形が正しく特定化できている場合

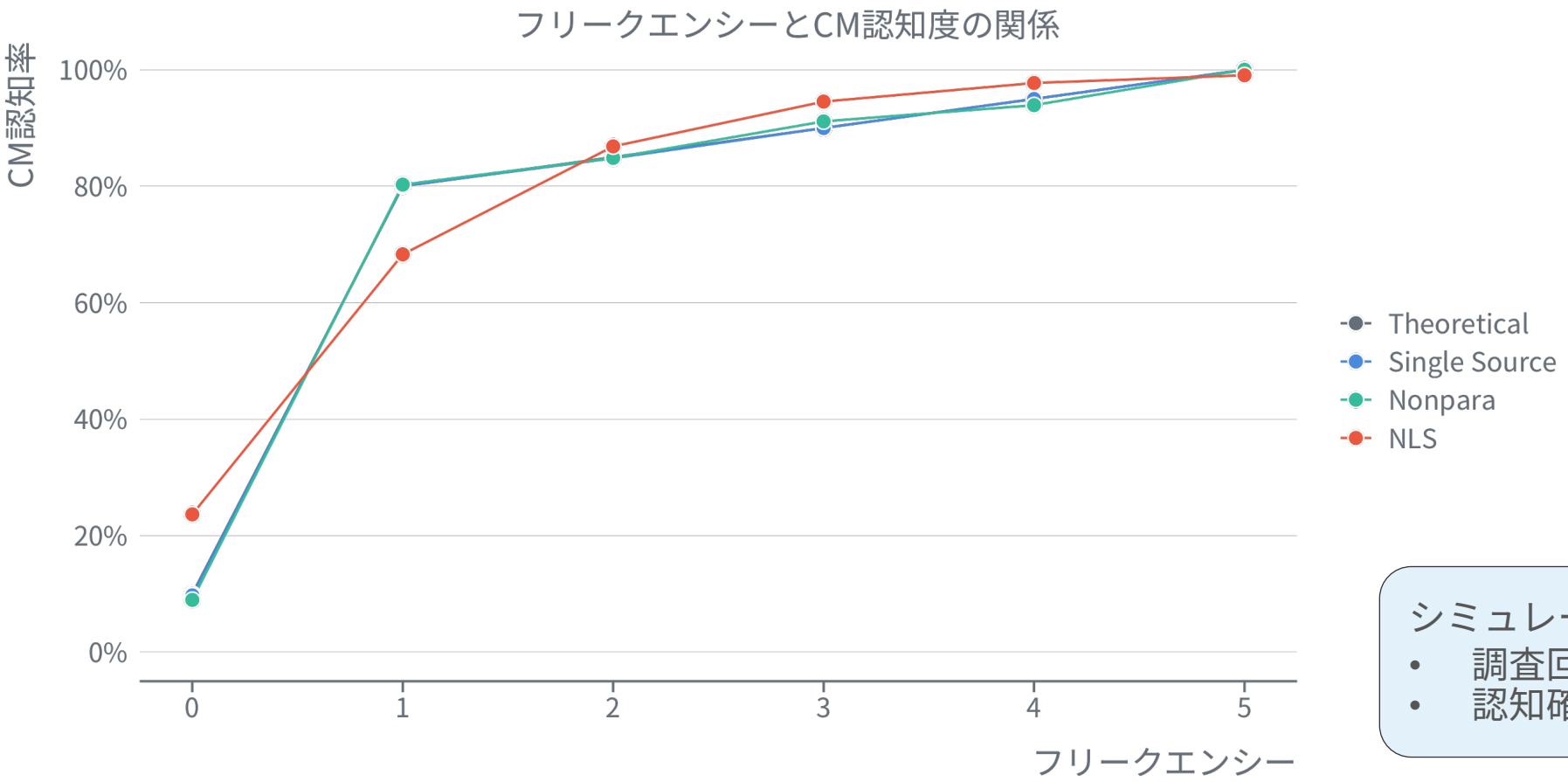
- 調査回数が少なくても、関数形の特定化に成功したNLSはうまく推定できている
- ノンパラの場合は推定が若干安定しない傾向



シミュレーションの設定
- 調査回数：5回
- 認知確率：$1-(1-0.2)(1-0.3)^{f}$

結果の比較：関数形の特定に失敗している場合

- 関数形の特定化に失敗するとNLSは推定がうまくいかない
- ノンパラの場合は関数形と特定していないので対応できている



シミュレーションの設定
- 調査回数：20回
- 認知確率：0.1, 0.8, 0.85, 0.9, 0.95, 1

まとめ

まとめ

- フリークエンシーとCM認知率の関係は通常シングルソースデータから推定するが、実務ではシングルソースデータは手に入らないことも多い
- いくつかの仮定のもとで、マルチソースデータからもフリークエンシーとCM認知率の関係を推定することができる
- 関数形を特定することで、パラメータの解釈性を高め、推定結果を安定させることができる。関数形を外すとバイアスがかかるので注意が必要
- 簡単な数理モデルを考えることでビジネス課題を解決できる（かもしれない）

参考：ベイズ推定の場合

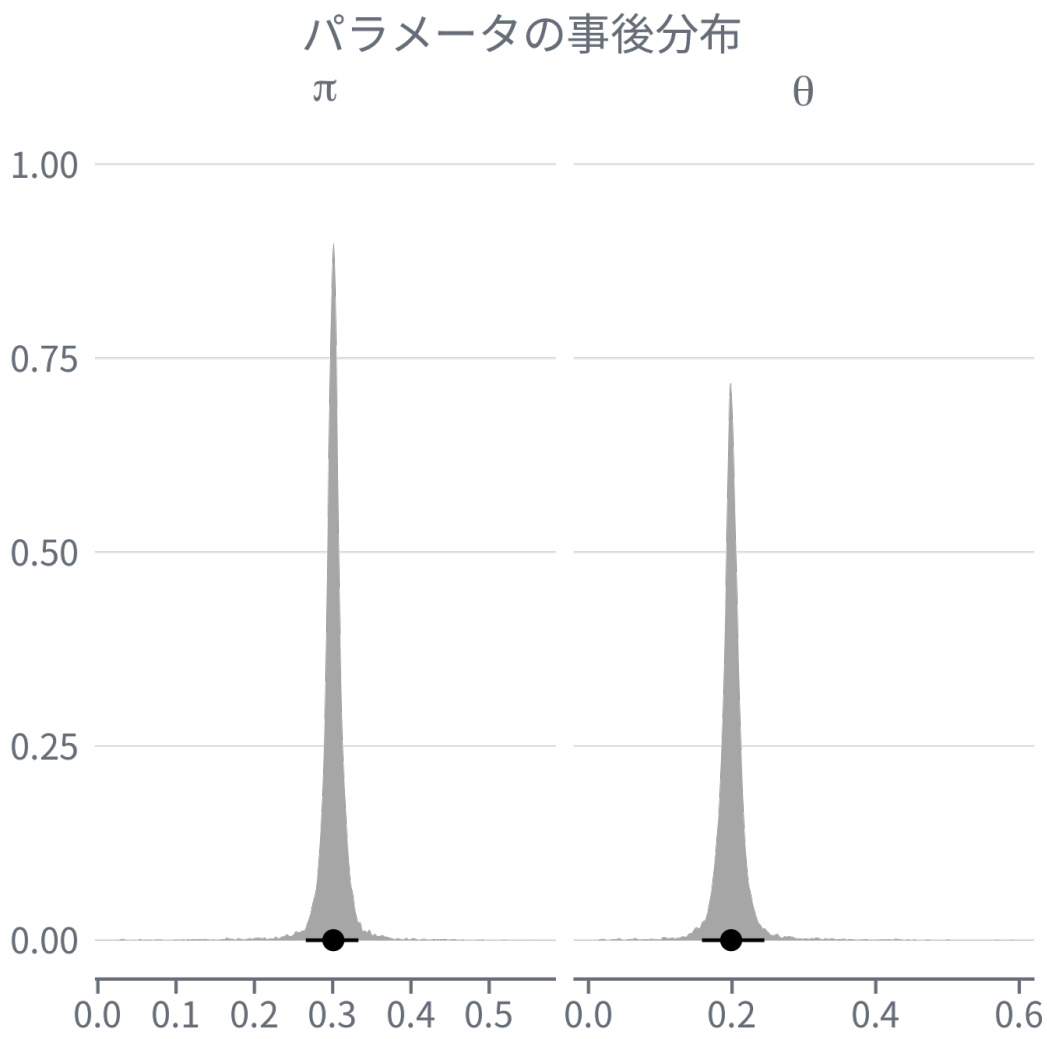
事前分布に一様分布を指定しているが（Beta(1, 1)は一様分布）、
事前知識があるなら事前分布に反映することも可能

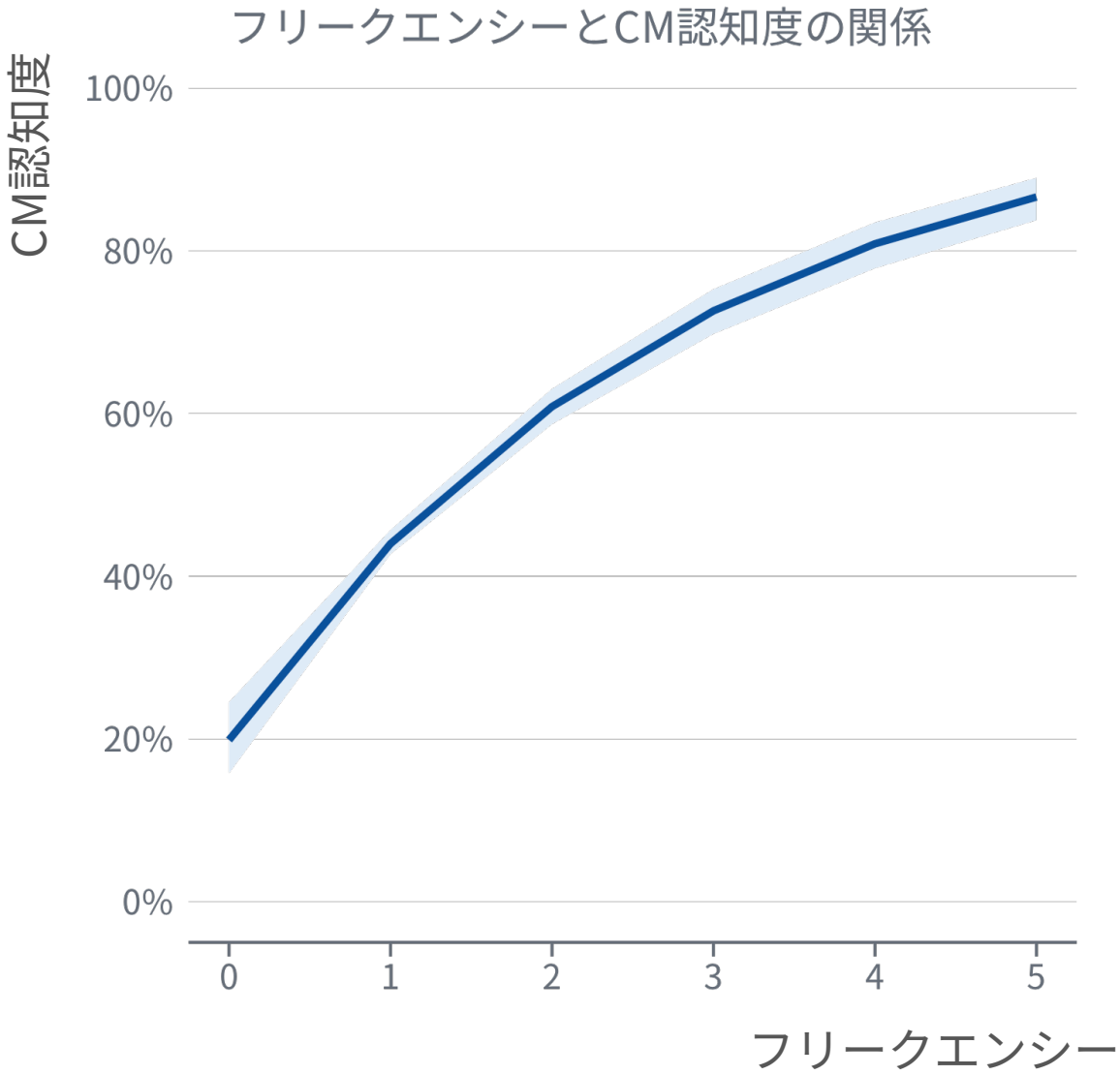
```
brm(
  bf(
    get_nls_formula(max_fq),
    theta ~ 1,
    pi ~ 1,
    nl = TRUE
  ),
  prior = c(
    prior(beta(1, 1), lb = 0, ub = 1, nlpar = "theta"),
    prior(beta(1, 1), lb = 0, ub = 1, nlpar = "pi")
  ),
  data = data,
  iter = 8000,
  cores = 4,
  seed = 42
)
```

$$a_t = 1 - (1-\theta)\sum_{f=0}^{\bar{F}}(1-\pi)^f s_{f,t} + \epsilon_t$$

$$\theta \sim \mathrm{Beta}(\alpha_\theta, \beta_\theta)$$
$$\pi \sim \mathrm{Beta}(\alpha_\pi, \beta_\pi)$$
$$\epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

ベイズ推定を行うことで不確実性を考慮した予測が可能





Rを使えるデータサイエンティスト、エンジニアを大募集