JAPLA 研究会　中野会場
2017 年 5 月 20 日
加筆 2017 年 6 月 17 日
鳥邊錬太郎

忘却を考慮した指数曲線モデル

広告認知率予測モデルの考察と忘却を考慮したプログラムの作成の依頼が来たので、昔の資料をひっくり返して、作業を始めた。これにより、jDllServer を Windows に組み込んだ実験は暫し休養しています。この間、jSoftware.com の MailingList を眺めていたら、どうやらあちらでは jDllServer ではなく、jExeServer を使用するのが主流らしい。さらに、jExeServer の Windows プラットフォームは VisualStudioCSharp(C#.NET)がメインとなっている。

そこで、現在進行中の VB.NET のプログラムは、C#.NET に書き換え中である。と言っても VB.NET と V#.NET では「月とスッポン」ほどの差があり、現在勉強しながらのプログラミング中です。また、これほどの苦労を背負いながら C#.NET に宗旨替えをするには、大きな理由が飛び込んで来たことも、理由の一つです。

それは、スマホ用のアプリ作成プログラムを C#.NET で作成して Android-OS や iOS のスマホ対応のアプリに変換するシステムを開発している Xamarin 社が MS 社に身売りしたことから、この 1 年ほどの間に、Microsoft VS2015 や 2017 に無料で配布され、私はスマホ（特に Android-OS）のアプリ開発に興味があるので、この度、Xamarin が対象

JAPLA5 月の勉強が中途半端であったことと、佐久の小屋のトイレが、今年正月以降の厳冬の環境下、水抜きの不手際から、水洗用タンクと便器基台（何れも陶器製）に亀裂とヒビがが入り、小屋への居住が不可能になり、これまで、修理の手筈や手配を、やっと済ませ 6 月中旬の工事に漕ぎつけました。したがって、勉強は 5 月分と今月分を合併した次第です。したがって、**JAPLA6 月（17 日中野）は佐久の山小屋で、生い茂る草刈りとトイレ修理の最中で、お休みさせて頂きます。**

JAPLA 5 月では、「非線形・指数曲線モデル」から実験に入りましたが、このひと月、勉強を進めていたうちに、指数曲線でも変形の仕方で、線形の形で応用できるので、そのように書き換えました。

1．線形モデル

【重回帰分析】

例題：「中古住宅の価格と住宅の置かれた設定条件等をもとに価格の要因を探る」という問題を考えてみる。

物件名	目的変数(y_i)	説明変数(x_{ij})		
	中古住宅価格	広さ(x_{i1})	駅迄徒歩(x_{i2})	築年数(x_{i3})
A	950	51	10	30
B	990	48	7	28
C	1580	63	8	23
D	1780	79	8	30
E	3350	80	11	4
F	3390	72	8	3
G	3800	120	6	16

表１

表１のデータを用いて、

目的変数を $Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix}$

説明変数を $X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1m} \\ 1 & x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2m} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 1 & x_{i1} & x_{i2} & \cdots & \boldsymbol{x_{ij}} & \cdots & x_{im} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nj} & \cdots & x_{nm} \end{bmatrix}$

と、定義する。

一般形式で眺めると、目的変数 y は、

$$y_i = b_0 + b_j x_{ij}\ \{(j = 1,2,\cdots,m), i = 1,2,\cdots,n\} \quad \cdots\cdots\cdots\cdots (1)$$

のように表せる。

注） b_0 は常数項です。回帰平面がゼロ以外の切片を持つように考慮したもの。
以下、添え字は$\{(j = 1,2,\cdots,m), i = 1,2,\cdots,n\}$を省略します。

【最小二乗法】

回帰分析は、このQを使って予測モデルを推定する。

$$Q = \sum (y_i - \hat{y}_i)^2 = \sum \left(y_i - b_0 - b_j x_{ij}\right)^2 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots（2）$$

Qはxに関する一次方程式の関係になっている。残差平方和Qを最少にするには、式（２）をb_0およびb_jで偏微分して、それぞれをゼロとおいて、b_0およびb_jを解けばよい。この方法を**最小二乗法**という。

Qは、

$$Q = \sum y^2 - 2b_0 \sum y + n b_j^2 + 2 b_0 b_j \sum x + b_j \sum x^2 \quad \cdots\cdots\cdots\cdots\cdots（3）$$

と書けるから、

$$\begin{cases} \dfrac{\partial Q}{\partial b_0} = n b_0 - \sum y + b_j \sum x = 0 \\ \dfrac{\partial Q}{\partial b_j} = b_0 \sum x - \sum yx + b_j \sum x^2 = 0 \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots（4）$$

を解けばよい。未知数が２つなので解はb_0およびb_jに関する連立方程式を解くことに帰着する。

すなわち、

$$\begin{cases} n b_0 - b_j \sum x = \sum y \\ b_0 \sum x + b_j \sum x^2 = \sum xy \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots（5）$$

この式より、未知数、b_0およびb_jの連立方程式を解けばよい。

$$b_0 = \frac{\sum y \sum x^2 - \sum x \sum xy}{n \sum x^2 - (\sum x)^2} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots（6）$$

注）常数項の係数

$$b_j = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots（7）$$

注）説明変数の係数

となり、解は求まる。

しかし、未知数が 3 つ以上になると手計算では面倒で困難である。J 言語で計算するには、行列演算が都合がよい。

式（５）を、行列で表せば以下のようになる。

$$\begin{bmatrix} n & \sum x \\ \sum x & \sum x^2 \end{bmatrix}\begin{bmatrix} b_0 \\ b_j \end{bmatrix} = \begin{bmatrix} \sum y \\ \sum xy \end{bmatrix} \cdots\cdots\cdots\cdots \quad (8)$$

未知数、b_0およびb_jについて書き換えると、

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} n & \sum x \\ \sum x & \sum x^2 \end{bmatrix}^{-1}\begin{bmatrix} \sum y \\ \sum xy \end{bmatrix} \cdots\cdots\cdots\cdots \quad (9)$$

のように書ける。

【J 言語による計算】

モデル式行列表示、(8)式より、

$$\begin{bmatrix} n & \sum x \\ \sum x & \sum x^2 \end{bmatrix}\begin{bmatrix} b_0 \\ b_j \end{bmatrix} = \begin{bmatrix} \sum y \\ \sum xy \end{bmatrix} \quad \left(j = 1, 2, \cdots, m\text{: 説明変数の数, } b_0\text{は常数項}\right)$$

は、

$$\frac{\partial Q}{\partial b.} = -2X'Y + 2X'X = 0 \cdots\cdots\cdots\cdots \quad \text{(J-01)}$$

から導かれた、残差平方和（予測誤差の平方和）をゼロとおいて推定したパラメータ$[b_0,\ b_1]$

$$\begin{bmatrix} b_0 \\ b_i \end{bmatrix} = [X'X]^{-1}X'Y \quad \cdots\cdots\cdots\cdots \quad \text{(J-02)}$$

となる。

【J 言語に渡すデータの整理】

ここで、実際のデータを当てはめてみる。

$$Y = \begin{bmatrix} 950 \\ 990 \\ 1580 \\ 1780 \\ 3350 \\ 3390 \\ 3800 \end{bmatrix}、X = \begin{bmatrix} 1 & 51 & 10 & 30 \\ 1 & 49 & 7 & 28 \\ 1 & 63 & 8 & 23 \\ 1 & 80 & 8 & 30 \\ 1 & 80 & 11 & 4 \\ 1 & 72 & 8 & 3 \\ 1 & 120 & 6 & 16 \end{bmatrix}、(\mathrm{n} = 7)$$

【J 言語でのコーディング】

```
NB.連立方程式の解
NB.目的変数の定義
    ]Y=:7 1$y=:950 990 1580 1780 3350 3390 3800
 950
 990
1580
1780
3350
3390
3800
NB.説明変数・各行のの定義
   x1=:52 10 30
   x2=:49 7 28
   x3=:63 8 23
   x4=:80 8 30
   x5=:80 11 4
   x6=:72 8 3
   x7=:120 6 16
NB.説明変数・各行を結合して行列を作る
  ]x=:7 3$x1,x2,x3,x4,x5,x6,x7
 52 10 30
 49  7 28
 63  8 23
 80  8 30
 80 11  4
 72  8  3
120  6 16
NB.説明変数に常数項を加える
  ]X=:1&,"1 x
 1 52 10 30
 1 49 7 28
 1 63 8 23
 1 80 8 30
 1 80 11 4
 1 72 8 3
 1 120 6 16
NB.式(J-02)をJ言語で計算した結果
NB.予測パラメータはbj (j = 0, 1, ⋯, mの順に表示されている)
  ]b=:((|:X) +/ . * Y)%.(|:X) +/ . * X
 1650.8          ⇐ β0 常数項のパラメータ
 27.9257         ⇐ β1 第 1 説明変数のパラメータ
_19.8146         ⇐ β2 第 2 説明変数のパラメータ
_66.9854         ⇐ β3 第 3 説明変数のパラメータ
```

2．対数モデルの場合

一般に、回帰分析は、直線（単回帰）や重回帰分析がありますが、実際の社会現象を扱う場合には、都合よく行かないのが常です。例えば、テレビ番組や、広告のリーチなどを予測すると、曲線がよく当てはまる。曲線にも色々あり、これらを総称して、非線形な方程式と呼んでいます。

非線形回帰モデルは、一般的には、そのまま最小二乗法を用いることは困難あり、幾つかの解析方法が、先人の知識で開発されています。

ここでは、もっとも簡単な方法を幾つか考えてみます。

下表のデータは、架空の各週に対する累積広告投下量GRP（X）とリーチ（Y）の数表です。リーチは、一般的に広告投下時期に、その広告の累積到達程度（一般的には同一広告に１回以上接触した率）などを視聴率調査などによって収得した数値です。

この数表から、各非線形方程式の当てはめ曲線とそのパラメータを求めることを考えてみる。

DataNo.	Weeks	Reach(X)	累積 GRP(Y)	参考 GRP/W
1	week-01	22.6	26.6	26.6
2	week-02	43.0	66.1	39.5
3	week-03	55.4	164.3	98.2
4	week-04	62.3	221.4	57.1
5	week-05	64.3	278.1	56.7
6	week-06	65.9	323.7	45.6
7	week-07	67.9	418.4	94.7
8	week-08	69.0	479.0	60.6
9	week-09	71.1	534.7	55.7
10	week-10	71.7	606.1	71.4
11	week-11	73.1	673.3	67.2
12	week-12	73.5	718.2	44.9
13	week-13	74.2	791.5	73.3
14	week-14	74.2	795.2	3.7

表２GRP とリーチ

2．対数モデルは、以下の通り、

$$Y = b_0 + b_j \cdot ln(X) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

(1.1)

この式を用いて、表１のリーチを予測してみると、まず、リーチ(観測値) Y_i の推定値 $\hat{Y}_i$ は、

$$\hat{Y}_i = b_0 + b_j \cdot ln(X_i)$$

と書けるから、観測値 Y_i と推定値 $\hat{Y}_i$ の残差平方和 Q は、

$$\mathrm{Q} = \sum\left\{Y_i - b_0 - b_j \cdot ln(x_i)\right\}^2 =$$
$$\sum Y_i^2 + \mathrm{n}b_0^2 + b_j^2 \sum\{\ln(X_i)\}^2 - 2b_0 \sum Y_i$$
$$-2\mathrm{b}_j \sum Y_i \cdot \ln(X_i) + 2b_0 b_j \sum \ln(X_i) \cdots\cdots\cdots\cdots \quad (1.2)$$

のように展開できるので、

(1.2)式を式を、b_0およびb_jでそれぞれ偏微分してゼロとおけば、

$$\begin{cases} \frac{\partial Q}{\partial b_0} = nb_0 + b_j \sum ln(X_i) - \sum Y_i = 0 \\ \frac{\partial Q}{\partial b_j} = b_0 \sum ln(X_i) + b_j \sum\{ln(X_i)\}^2 - \sum Y_i \, ln(X_i) = 0 \end{cases} \cdots\cdots \quad (1.3)$$

したがって、b_0およびb_jを行列表現すれば、

$$\begin{bmatrix} b_0 \\ b_j \end{bmatrix} = \begin{bmatrix} n & \sum ln(x_i) \\ \sum ln(x_i) & \sum\{ln(x_i)\}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum Y_i \\ \sum Y_i \cdot ln(X_i) \end{bmatrix} \cdots\cdots\cdots\cdots\cdots \quad (1.4)$$

となり、b_0およびb_jが求まります。

この解の形式を見ると、線形モデルと全く同じ形式であることがわかる。

実際の計算は、説明変数が複数となるので、一般形はマトリックスの行と列が増えるだけの計算になる。J 言語にとっては、得意のところです。

実際に計算すると、

$\sum\{ln(X)\}$	$\sum\{ln(X^2)\}$	$\sum\{Y \cdot ln(X)\}$	$\sum Y$
80.7185	478.4303	5307.03	888.02

となる。従って、（1.4）式の右辺の左項のマトリックスは

$$\begin{bmatrix} n & \sum ln(X_i) \\ \sum ln(X_i) & \sum\{ln(X_i)\}^2 \end{bmatrix}^{-1} \Rightarrow \begin{bmatrix} 14 & 80.7185 \\ 80.7185 & 478.4303 \end{bmatrix}^{-1} \Rightarrow \begin{bmatrix} 2.6208 & -0.4422 \\ -0.4422 & 0.0767 \end{bmatrix}$$

となります。

（1.4）式の右辺の右項のマトリックスは、

$$\begin{bmatrix} \sum Y_i \\ \sum\{Y_i \cdot ln(X_i)\} \end{bmatrix} \Rightarrow \begin{bmatrix} 888.02 \\ 5307.03 \end{bmatrix}$$

従って、

$$\begin{bmatrix} b_0 \\ b_j \end{bmatrix} = \begin{bmatrix} 2.6208 & -0.4422 \\ -0.4422 & 0.0767 \end{bmatrix} \begin{bmatrix} 888.02 \\ 5307.03 \end{bmatrix} = \begin{bmatrix} -19.2817 \\ 14.34560 \end{bmatrix}$$

となります。

【J 言語による計算】

ここでも、１章の式（J-01）および（J-02）を利用して行列表示で計算する。

$$\begin{bmatrix} b_0 \\ b_i \end{bmatrix} = [X'X]^{-1}X'Y$$

となる。

【J 言語に渡すデータの整理】

ここで、実際のデータを当てはめてみる。

$$Y = \begin{bmatrix} 22.6 \\ 43.0 \\ 55.4 \\ 62.3 \\ 64.3 \\ 65.9 \\ 67.9 \\ 69.0 \\ 71.1 \\ 71.7 \\ 73.1 \\ 73.5 \\ 74.2 \\ 74.2 \end{bmatrix} \quad x = \begin{bmatrix} 1 & 26.6 \\ 1 & 66.1 \\ 1 & 164.3 \\ 1 & 221.4 \\ 1 & 278.1 \\ 1 & 323.7 \\ 1 & 418.4 \\ 1 & 479.0 \\ 1 & 534.7 \\ 1 & 606.1 \\ 1 & 673.3 \\ 1 & 718.2 \\ 1 & 791.5 \\ 1 & 795.2 \end{bmatrix} \text{、} (\mathrm{n} = 14)$$

【J 言語でのコーディング】

```
NB.対数モデルによる、線形化予測計算
NB.目的変数・部分定義
    y1=:22.6 43.0 55.4 62.3 64.3 65.9 67.9
    y2=:69.0 71.1 71.7 73.1 73.5 74.2 74.2
NB.目的変数・部分データを接続
    ]Y=:14 2$y1,y2
22.6
43
55.4
62.3
64.3
65.9
67.9
69
71.1
71.7
73.1
73.5
74.2
74.2
NB.説明変数・各行の定義
   x1=:26.6 66.1 164.3 221.4 278.1 323.7 418.4
   x2=:479.0 534.7 606.1 673.3 718.2 791.5 795.2
NB.説明変数・・部分データを接続
   ]xs=:14 1$x1,x2
 26.6
 66.1
 164.3
 221.4
 278.1
 323.7
 418.4
 479
 534.7
 606.1
 673.3
 718.2
 791.5
 NB.説明変数の自然対数変換 ln(x)を返す。
   ]x=:^. xs
3.28091
4.19117
5.10169
5.39997
5.62798
5.77982
6.03644
6.1717
6.28171
```

```
6.40704
6.51219
6.57675
6.67393
6.67859
NB.説明変数に常数項を加える
   ]X=:1&,"1 x
1 3.28091
1 4.19117
1 5.10169
1 5.39997
1 5.62798
1 5.77982
1 6.03644
1 6.1717
1 6.28171
1 6.40704
1 6.51219
1 6.57675
1 6.67393
1 6.67859
NB.式(J-02)をJ言語で計算した結果
```

NB.予測パラメータは$b_j(j=0,1,\cdots,m$ の順に表示されている)

```
   ]b=:((|:X) +/ . * Y)%.(|:X) +/ . * X
_19.2709
14.3458
```

_19.2709 ⇐ β_0 常数項のパラメータ

14.3458 ⇐ β_1 第1説明変数のパラメータ

```
NB.
NB.Reach 予測値
   Yhat=:X +/ . * b
27.7964
40.8548
53.9171
58.1961
61.4671
63.6453
67.3267
69.2672
70.8453
72.6434
74.1518
75.0779
76.4721
76.539
NB.
NB.説明変数、目的変数(実測)、予測値、予測誤差表示
   'Result=';xs;Y;Yhat;Y-Yhat
```

説明変数 累積 GRP	目的変数 実績 Reach	予測 Reach	実測-予測 Reach 残差

Result	26.6	22.6	27.7964	_5.19645
	66.1	43.0	40.8548	2.14517
	164.3	55.4	53.9171	1.48294
	221.4	62.3	58.1961	4.10392
	278.1	64.3	61.4671	2.83293
	323.7	65.9	63.6453	2.25472
	418.4	67.9	67.3267	0.57328
	479.0	69.0	69.2672	_0.26716
	534.7	71.1	70.8453	0.25471
	606.1	71.7	72.6434	_0.94337
	673.3	73.1	74.1518	_1.05178
	718.2	73.5	75.0779	_1.57790
	791.5	74.2	76.4721	_2.27205
	795.2	74.2	76.5390	_2.33896

３．忘却を考慮した非線形曲線モデルの場合

広告会社現役のころ、盛んに用いた拡張指数曲線モデルのプログラム化を、友人から発注を受けたので、今回は J 言語で挑戦することになった。モデル式は、以下の通りであるが、厄介なのはリカーシブモデルであるため、未だに解析的に解けていないので、今回も「挟み撃ち法」で解くことになったが、これまでは、VB で解いていたが、今回はＪ言語で挑戦したいと思う。

この問題は、たしか、2005 年以前に鈴木義一郎先生と挑戦したことがあったが、勿論先生はＪで解いたが、残念なことに、その Script は残っていないようです。

さて、そのモデルとは以下のようであったが、正しい呼び名は不明です。

その特徴は、例えば広告の効果としての、広告認知度や知名度が、時間とともに忘れられるという、忘却というメカニズムを考えているということにあります。

つまり、いくら宣伝費を投下しても忘却されてしまえば、元の木阿弥となって仕舞うわけで、広告接触率は増加しても、真の広告効果は増加しないことになります。

このモデルは、忘却というパラメータδを設けて、その忘却係数を 1 期前の広告効果に作用させている。また、予測広告効果が、0 から 100%の範囲を超えない工夫もしてある。モデル内の 100 の部分は、効果が比率で与えられているときは「１」を設定して、百分率で与えられているときは「100」を設定すればよい。

しかし、このモデルは解析的に解けなかったため。そこで、ここでは挟み撃ち法で解くことにした。

$$Y_i = (\delta \cdot Y_{i-1} - 100) \cdot e^{-\beta x} + 100 \cdots\cdots\cdots\cdots \quad (14)$$

x_i ：第 i 期の広告刺激

Y_i ：第 i 期の刺激による反応

Y_{i-1} ：1 期前の刺激による反応

δ ：広告刺激の忘却率

β ：広告刺激のパラメータ

挟み撃ち法のアルゴリズムは、おおよそ、以下のとおりである。

（ただし、このルーチンはβのみの予測。実際には、δの予測も付加する。）

Program は VisualBasic

```
Private Sub Hsamiuchi()
    'βの挟み撃ち法開始
        Erase yTemp
        Erase yTempMin
        Erase yTempMax
```

```
ReDim yGoal(n) As Double
ReDim yTemp(n) As Double
ReDim yTempMin(n) As Double
ReDim yTempMax(n) As Double
Dim D As Double = 0.95
yDevMax = 9E+30
yDevMin = 0
Bmax = 0.01
Bmin = 0.0000001
While Abs(yDevMax - yDevMin) > 0.0000001
    yTemp(0) = y(0)
    yTempMax(0) = y(0)
    yTempMin(0) = y(0)
    ySum = 0
    ySumMax = 0
    ySumMin = 0
    ySumSquare = 0
    ySumSquareMax = 0
    ySumSquareMin = 0
   ‘
    B = (Bmax + Bmin) / 2#
    Delta = B / 100
    BdeltaMin = B - Delta
    BdeltaMax = B + Delta
    '修正指数曲線挟み打ち'βの予測
        For i = 1 To n
            yTemp(i) = (D * yTemp(i - 1) - 100) *
                                     Exp(-B * x(i)) + 100
            ySum = ySum + yTemp(i)
            ySumSquare =
                    ySumSquare + (y(i) - yTemp(i)) ^ 2
        Next i
        ySumBar = ySum / n  '平均
        'β-Minの予測
        For i = 1 To n
            yTempMin(i) = (D * yTempMin(i - 1) - 100) *
                              Exp(-BdeltaMin * x(i)) + 100
            ySumMin  = ySumMin + yTempMin(i)
           ySumSquareMin = ySumSquareMin +
                      (y(i) - yTempMin(i)) ^ 2   '平方和
        Next i
        ySumBarMin = ySumMin / n
          '平均
        'β-Maxの予測
        For i = 1 To n
            yTempMax(i) = (D * yTempMax(i - 1) - 100) *
                            Exp(-BdeltaMax * x(i)) + 100
            ySumMax = ySumMax + yTempMax(i)
           ySumSquareMax = ySumSquareMax + _
                            (y(i) - yTempMax(i)) ^ 2
        Next i
        ySumBarMax = ySumMax / n    '平均
    'βの近傍で
```

```
            Select Case (ySumSquareMin - ySumSquareMax)
                Case Is > 0
                    '－近傍から＋近傍へ下って改良される。
                    'yDevMax を採用
                    yDevMin = ySumSquare
                    Bmin = B
                Case Is < 0
                    '＋近傍から－近傍へ下って改良される。
                    'yDevMin を採用
                    yDevMax = ySumSquare
                    Bmax = B
                Case Is = 0
                    GoTo WendEnd
            End Select
    Wend
WendEnd:
End Sub
```

ここで使用するデータ

期	X(刺激)	Y(認知)	$\hat{Y}$(認知)
-1		18.0	
1	300	30.2	
2	0	28.9	
3	300	37.8	
4	0	36.2	
5	0	34.7	
6	100	36.3	
7	100	37.7	
8	200	41.3	
9	300	46.2	
10	0	44.2	
11	0	42.3	
12	0	40.5	
13	0	38.8	
14	0	37.1	
15	0	35.5	
16	0	34.0	
17	0	32.5	
17	0	31.1	
19	0	29.8	
20	0	28.5	

表３：広告刺激量と認知量