平成 21 年度茨城大学工学部情報工学科卒業論文

ロジスティック回帰モデルを用いた ネタバレ記事の検出

平成 21 年 2 月 10 日

工学部情報工学科

執筆者：斉藤章久(04T4027T)

指導教官：新納浩幸　准教授

平成２１年度　茨城大学工学部情報工学科　卒業論文

ロジスティック回帰モデルを用いたネタバレ記事の検出

著者：斉藤章久(04T4027T)
指導教員：新納浩幸　准教授

論文要旨

本研究では、ネタバレ記事の特徴を考察し、ネタバレ記事の自動検出を試みた。

インターネットの普及に伴い、インターネット上の掲示板の利用者の数も増えている。利用者は自分の興味がある分野の掲示板にアクセスし、知りたい情報を手に入れることができ、趣味の合う人同士で交流ができる。しかし、この掲示板で知りたくない情報を見てしまうことがある。そういう情報のひとつにネタバレがある。

ネタバレとは、映画やゲームや漫画なのでまだ先の内容を知らない人にその内容をバラしてしまう書き込みのことである。

ここでは、ネタバレ記事に多く見られる６つの特徴を設定し、それを用いて検出を行う。具体的にはロジスティック回帰を用いてネタバレ記事かどうかを判別する。ロジスティック回帰は、ある事象が発生する確率を直接推定する手法である。従属変数が「あり」「なし」のように２値を扱う場合に利用できる。先の特徴も２値であるため、ロジスティック回帰で用いる説明変数を６つの特徴に設定した。

ネット上の掲示板から得た３つのデータセットを用いて実験を行った。実験の結果、検出のＦ値が 0.417、0.469、0.222 となった。比較的に精度よく検出できた。

今後の課題としては、より信頼性の高いネタバレの特徴を発見し、更に検出の精度を高めることである。

目次

図目次

表目次

第一章　はじめに

1.1 本論文の要旨

インターネットの普及に伴い、インターネット上の掲示板の利用者の数も増えている。利用者は自分の興味がある分野の掲示板にアクセスし、知りたい情報を手に入れることができ、趣味の合う人同士で交流をしたりすることができる。

しかし、この掲示板で知りたくない情報を見てしまうことがある。そういう情報のひとつにネタバレというものがある。

ネタバレとは、映画やゲームや漫画なのでまだ先の内容を知らない人にその内容をバラしてしまう書き込みのことである。このような書き込みは、先の内容を楽しみにしていた人にとっては見たくない情報である。

本研究の目的は、このネタバレの特徴を見つけ出し、ネタバレを自動検出することである。

ここでは、ネタバレの文章に多く見られる６つの特徴を設定し、検出を行う。具体的にはロジスティック回帰を用いて記事をネタバレかどうか判別する。ロジスティック回帰で用いる説明変数を先の６つの特徴に設定した。

実験は３つのデータで行い、それぞれＦ値が 0.417、0.469、0.222 となった。この実験を行って、ネタバレの特徴の一つである【「バレ」という文字が入っている】という特徴が、逆にネタバレの文章以外で使われていることが多いことがわかった。この特徴を省き実験を行うと、Ｆ値が 0.723、0.744、0.160 と一部のＦ値は上がったものの、一つのデータは下がってしまった。

本論文はこの研究を細かく書き残したものである。

1.2 本論文の構成

第２章では、ネタバレの判別を行う際の手法であるロジスティック回帰分析を紹介する。

第３章では、ネタバレの特徴である６つを紹介する。また、ネタバレの特徴をプログラムで判別する方法を紹介する。

第４章では、統計解析プログラムであるＲについて紹介する。まずは簡単な例でＲの使い方を示してある。Ｒにおいてのロジスティック回帰分析のやり方もこの章で紹介する。

第５章では、実験の流れと結果を書き記している。実験の流れは第３章と第４章で説明したことを用いながら紹介する。

第６章では、今回の実験の考察を述べている。実験の結果を確認した上での考えを書き記してある。また、実験で気づいたことに基づき、再実験を行っている。

第二章 ネタバレ判別

2.1 ロジスティック回帰

ロジスティック回帰分析は発生確率を予測する手法である。予測結果はすべて０から１の間の数字で表される。

従属変数に２値の質的変数を用いる

（例）Yes or No

この従属変数の値を実績値として用い、説明変数を用いて発生確率を説明する。

説明変数をax+bという形にし、分析結果をyとすると

$$y = ax + b \qquad (2.1)$$

という式にすることができる。

ここで、yの値は0から1という範囲が決められているので、式(2.1)では成り立たない場合が出てくる。

これのために式を

$$\log e\{y/(1-y)\} = ax + b \qquad (2.2)$$

という式を用意する。

2.2 オッズ

オッズ比とは、ある事象が起こる確率と起こらない確率の比である。
例えば、Aの商品を買う確率をyとし、買わない確率を1-yとする。
その場合の y/1-y の値のことをオッズ比と呼ぶ。

2.3 ロジスティック回帰のグラフ化

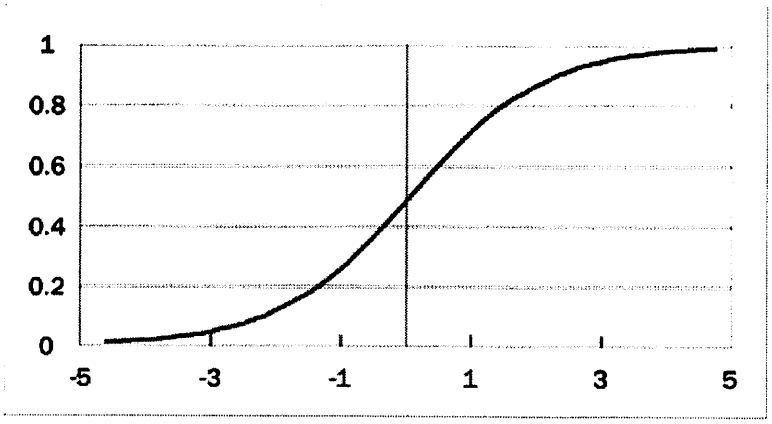

図2.1： ロジスティック回帰

図2.1の横軸はax+bでありlogである。
また、縦軸は発生確率であるyとなっている。

この図はax+bがどんな値をとろうとも、yが1から0の間の値になる。
また、このようなS字を崩したようなデータの形はいろいろな例で使える。

（例）・細菌の繁殖の発生から繁殖による隆盛
・人間の学習線

第三章 ネタバレの特徴

3.1 ６つの特徴

（その１） 「」や『』の数

（例）
緑間 「くっ…」
かがみんに抜かれる。
宮地 「一体なんなんだよコイツは…!?」
木村 「ついこの間まで中坊だった奴に ウチが押されてるってのかよ!?」
大坪 「全国でも見たことがない… なんだこの…常軌を逸した眺躍は…!?」
大坪木村を越えてダンクを決めるかがみん。
「うわぁあ高えぇー!!一人で秀徳を圧倒してるぞ!?」
小金井 「スゲーなナイス火神!!」
かがみん 「もっとガンガンボールくんねーですか」

このように書き込む人が物語を小説のようにネタバレを書き込む場合が多い。
よって今回の実験では、「　や　『　が１記事に４つ以上ある場合は、ネタバレ要素として捉えることにした。

（その２）　*「来週」「次回」「次週」というキーワード*

（例１）

来週号は火神と青峰が初対面。

1on1で勝負するけど、火神はあっさりやられる。

（例２）

次回で試合は終わっちゃうよ。５５点差で負けるよ。

このように次回の内容をネタバレするときに、先頭に「来週」や「次回」などをつけることが多く見られる。

「来週」「次回」「次週」というワードが１つでもある時、ネタバレの要素となるとしている。

（その３）　*「らしい」というキーワード*

（例）

最終的に試合には負けて、新キャラが登場するらしいよ

推測の時などによく用いられる「らしい」もネタバレの文章の中によく見られることがわかった。

ただの推測の場合もあるが、例のように他人から聞いた情報を報告的に形で使われることが多い。

「らしい」というワードが１つでもある時、ネタバレの要素となるとしている。

（その４）　「だった」というキーワード

> *（例）*
> *ジャンプ立ち読みしてきたよ。*
> *黒子のバスケは黒子のパスが青峰にまったく通用せず、試合に負けてしまうっていう展開だったよ*

このように内容を自分で見てきて、それを人に語るような場合によく見られるパターン。「だった」というワードが１つでもある時、ネタバレの要素となるとしている。

（その５）　「バレ」というキーワード

> *（例）*
> *ネタバレ*
> *カラーはセンターに背中合わせの火神青峰　バックに黒子桃井　髪の色は名前そのままみたいだけど薄く赤みがかかってよく分からん*
> *煽りは『ついに対峙する二人・・・まさに猛犬！！』大人気御礼センターカラー*

ネタバレする前に「ネタバレ」や「バレ」などと宣言してからネタバレする投稿も何個か見ることができた。

「バレ」というワードが１つでもある時、ネタバレの要素となるとしている。

（その６）　*「みたい」というキーワード*

（例）
いや、今週の展開は熱いよ

黒子はキレると敬語から標準語に変わるみたいだし

（その３）と（その４）の両方の場合で使われるワード。推測だったり、自分で見た情報を伝える時などに使われるワードである。

「みたい」というワードが１つでもある時、ネタバレの要素となるとしている。

この６つを、今研究のネタバレの特徴として使用し、ネタバレを検出した。

3.2 perlによる特徴判別

3.1で説明した特徴を、perlを使って判別した。

学習データの811個の記事があるスレッドを、プログラムで一つの記事ごとにファイル分けした。そのプログラムソースは付録のA.1に書き記した。

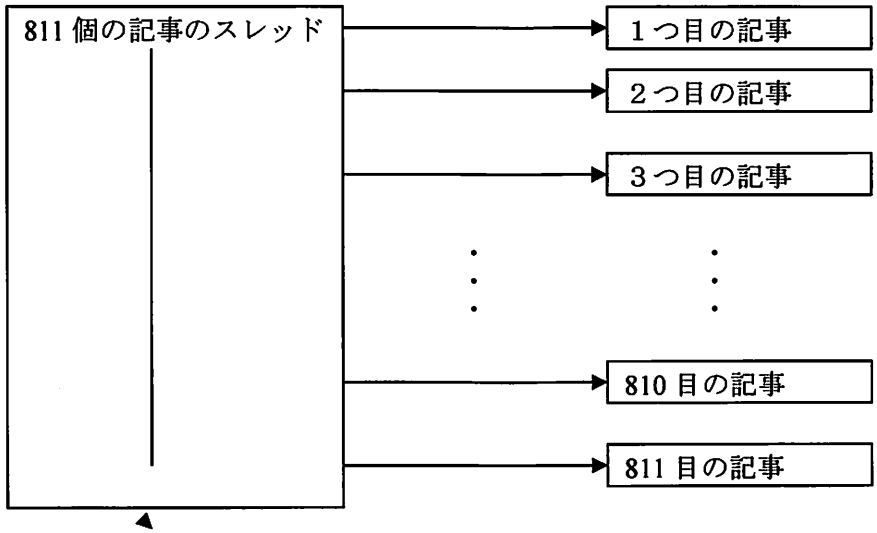

図3.1 スレッドのファイル化

方法

「2ちゃんねる」の記事は、記事番号が投稿の一番先頭に必ず来る。
さらにそこから「:」と空白、そして「名無し」と大抵来るので

```
if($_ =~ /^[0-9]*  :名/)
```

このような条件で記事の分別を行った。
また、その記事を「記事番号.txt」の形でファイル化した。

ネタバレ分別

811個に分けたファイル一つ一つに、第三章で記した６つの特徴があるかどうか調べる。
それぞれの特徴を配列で６個用意し、その特徴がある場合は１、ない場合は０を代入するプログラムになっている。
このプログラムソースは付録のA.2となっている。

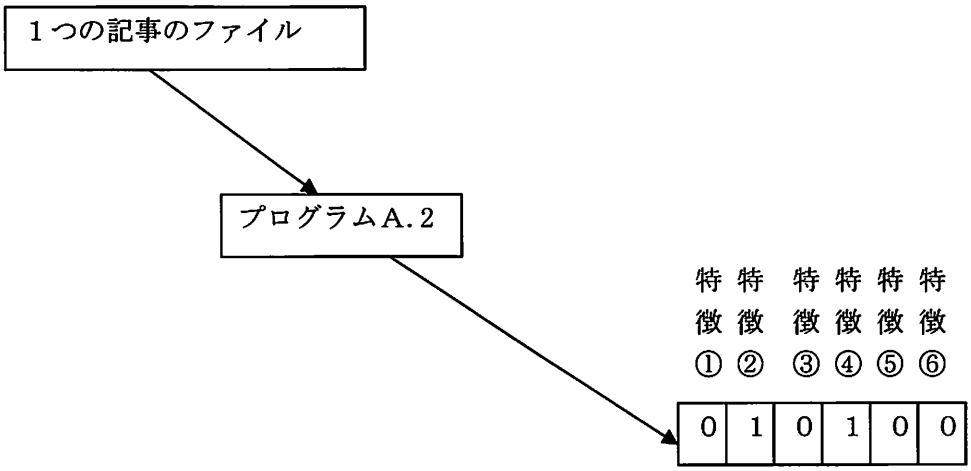

図3.2　１ファイルに対して６つの特徴を判別

最初の学習データで行う場合、プログラムソースのA.2.1を追加して、７番目の値も付け加える。

この値は、手作業でネタバレかネタバレじゃないかを区別した値である。１の場合はネタバレであり、０の場合はネタバレではない記事である。

このプログラムは付録のA.2のプログラムにA.2-Aのプログラムを付け加えて行った。

第四章 R

4.1 Rの使い方

Rとは統計解析のプログラムである。RはR言語、R環境とも呼ばれている。フリーソフトなので、誰でも簡単に使うことができる。

Rはニュージーランドのオークランド大学統計学科 Ross Ihaka とアメリカのハーバード大学生物統計学科 Robert Gentleman により開発が始められた。

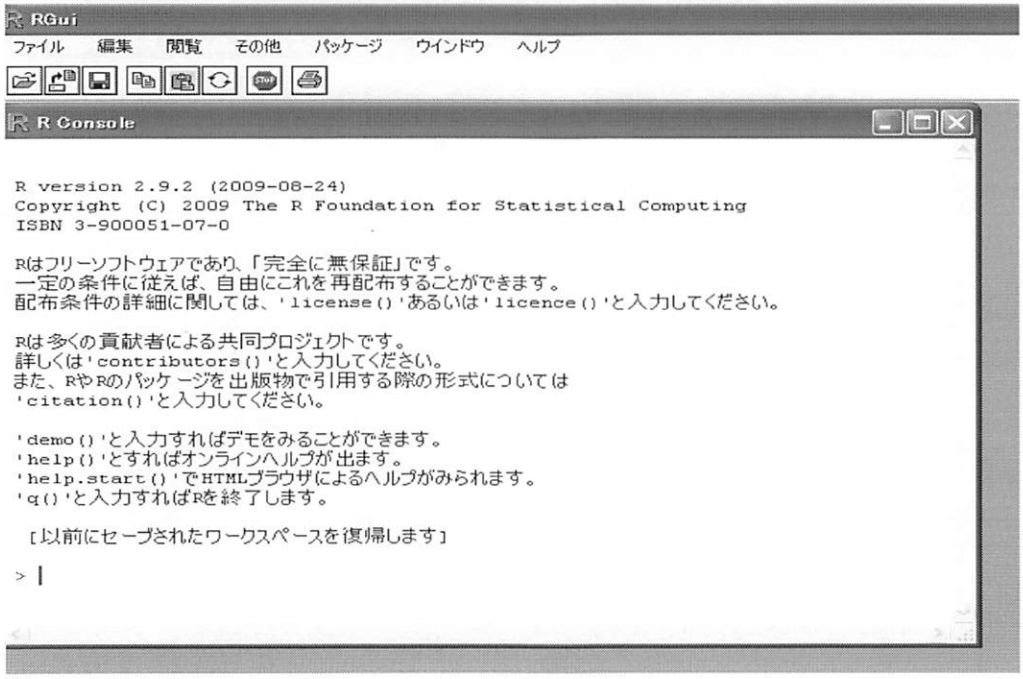

図 4.1: R起動画面

一番簡単な例としては、身体・体重・血液型をまとめた身体データというものがある。

```
Height Weight BloodType
159 45 B
168 67 A
155 49 AB
190 85 O
171 65 A
166 70 O
150 42 A
```

図 4.2: 身体データ

このデータをＲで表示させるには

```
データフレーム名 <- data.table("データフレーム名")
```

を入力する。そしてデータフレーム名を入力すれば表示される。

```
> df <- read.table("test.txt")
> df
       V1      V2         V3
1  Height  Weight BloodType
2     159      45          B
3     168      67          A
4     155      49         AB
5     190      85          O
6     171      65          A
7     166      70          O
8     150      42          A
>
```

図 4.3: Ｒで身体データを表示

また、身体データで身長を横軸、体重を縦軸にしたグラフを表示したい場合は

```
plot(データフレーム名$横軸,データフレーム名$縦軸)
```

今回だと

```
plot(df$Height,df$Weight)
```

で表示できる。プロット図を下に示す。

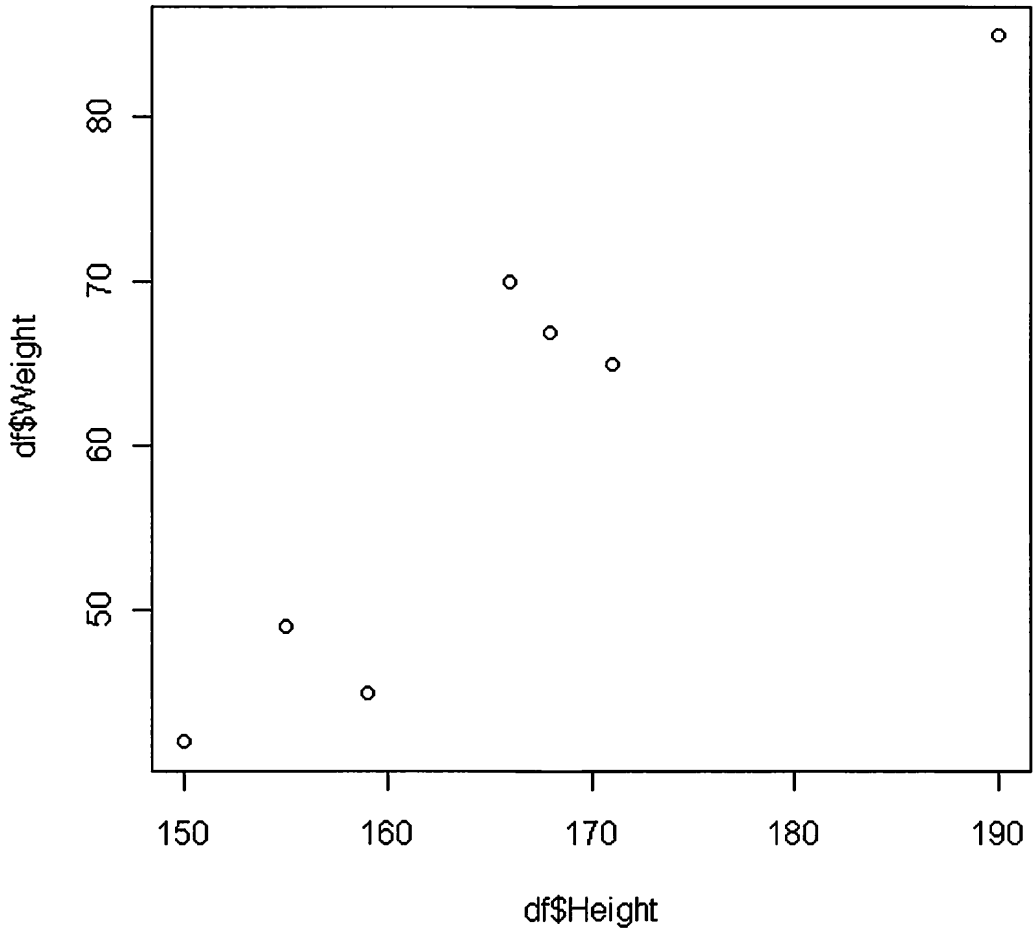

図 4.4: Ｒでの身体データのプロット図

4.2 Rでのロジスティック回帰分析

Rではglm関数を用いて

変数 <- glm(従属変数 ~ 独立変数1+独立変数2+・・・+独立変数n, family=binomial(link=" logit"),データフレーム)

のようにしてロジスティック回帰分析を行うことができる。

実際の実験と共に紹介する。

```
> b <- read.table("kuro3.txt")
> ans <- glm(V7 ~ V1 + V2 + V3 + V4 + V5 +V6, family=binomial(link="logit"),data=b)
Warning message:
In glm.fit(x = X, y = Y, weights = weights, start = start, etastart = etastart,  :
   数値的に 0 か 1 である確率が生じました
> summary(ans)

Call:
glm(formula = V7 ~ V1 + V2 + V3 + V4 + V5 + V6, family = binomial(link = "logit"),
    data = b)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8804  -0.1040  -0.1040  -0.1040   3.2320

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.2174     0.5560  -9.385  < 2e-16 ***
V1            39.5049  6012.6182   0.007  0.99476
V2             1.5611     1.1633   1.342  0.17960
V3             2.1549     1.2219   1.764  0.07780 .
V4             2.3145     0.7398   3.128  0.00176 **
V5           -16.8818  1961.3826  -0.009  0.99313
V6             1.1611     1.1261   1.031  0.30250
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 209.229  on 810  degrees of freedom
Residual deviance:  71.874  on 804  degrees of freedom
AIC: 85.874
```

図4.5 Rによるglm関数

この図4.1の1行目では、bにkuro3.txtのデータを読み込む。(kuro3.txtは学習データを6つの特徴分けしたデータである)

その後、glm関数を使ってロジスティック回帰を行う。

ここで使われている独立変数V1～V6は、それぞれ6つのネタバレの特徴の有無の値である。また、従属変数V7は事前に手作業で調べておいたネタバレかネタバレじゃないかの値である。

ロジスティック回帰分析はp個の独立変数xで式(4.1)のように独立変数yを予測する。

$$y=\frac{1}{1+\exp\{-(b_0+b_1x_1+b_2x_2+\cdots+b_px_p)\}} \qquad (4.1)$$

図4.1の結果にあるEstimateという値が、式(4.1)のb_1～b_6の値になる。

それぞれ

V1・・・39.5049
V2・・・1.5611
V3・・・2.1549
V4・・・2.3145
V5・・・-16.8818
V6・・・1.1611

という値を得た。

また、linear.predictors と fitted.values というものがある。

linear.predictors は$-b_0+b_1x_1+b_2x_2+\cdots+b_px_p$の値になっている。
fitted.values は独立変数 y の予測値となっている。

どちらもRで ans$linear.predictors と ans$fitted.values を入力すれば簡単に確認することができる。
下にプロットしたものを図示する。

```
> ans$linear.predictors
         1          2          3          4          5          6
-22.0992264 -22.0992264  -5.2174227  -5.2174227  -5.2174227  -5.2174227
         7          8          9         10         11         12
 -5.2174227  -5.2174227  -5.2174227  38.7569609 -22.0992264  -1.7417334
        13         14         15         16         17         18
 -5.2174227  -5.2174227  -2.9028765 -22.0992264  -5.2174227  -2.9028765
        19         20         21         22         23         24
 -5.2174227  -5.2174227  -5.2174227  -4.0562796  -5.2174227  -4.0562796
        25         26         27         28         29         30
 -5.2174227  -5.2174227 -22.0992264  -5.2174227  -3.0624763 -22.0992264
        31         32         33         34         35         36
 -5.2174227  -3.0624763  -5.2174227 -22.0992264  -5.2174227  -5.2174227
        37         38         39         40         41         42
 -5.2174227  -5.2174227  -2.9028765 -22.0992264  -5.2174227 -22.0992264
        43         44         45         46         47         48
 -5.2174227  -5.2174227  -5.2174227  -4.0562796 -22.0992264  -4.0562796
        49         50         51         52         53         54
 -5.2174227 -22.0992264 -22.0992264 -22.0992264 -22.0992264  -3.6562944
        55         56         57         58         59         60
 35.8485967  38.7569609 -22.0992264  -5.2174227  -5.2174227 -22.0992264
        61         62         63         64         65         66
 -5.2174227  -5.2174227 -20.9380833  -5.2174227 -22.0992264  -5.2174227
        67         68         69         70         71         72
 -5.2174227  -5.2174227  -5.2174227  -3.0624763 -22.0992264 -22.0992264
        73         74         75         76         77         78
```

図 4.6: linear.predictors の表示結果

```
> ans$fitted.values
          1           2           3           4           5           6           7           8           9
2.525968e-10 2.525968e-10 5.392051e-03 5.392051e-03 5.392051e-03 5.392051e-03 5.392051e-03 5.392051e-03 5.392051e-03
         10          11          12          13          14          15          16          17          18
1.000000e+00 2.525968e-10 1.490929e-01 5.392051e-03 5.392051e-03 5.201155e-02 2.525968e-10 5.392051e-03 5.201155e-02
         19          20          21          22          23          24          25          26          27
5.392051e-03 5.392051e-03 5.392051e-03 1.701866e-02 5.392051e-03 1.701866e-02 5.392051e-03 5.392051e-03 2.525968e-10
         28          29          30          31          32          33          34          35          36
5.392051e-03 4.468188e-02 2.525968e-10 5.392051e-03 4.468188e-02 5.392051e-03 2.525968e-10 5.392051e-03 5.392051e-03
         37          38          39          40          41          42          43          44          45
5.392051e-03 5.392051e-03 5.201155e-02 2.525968e-10 5.392051e-03 2.525968e-10 5.392051e-03 5.392051e-03 5.392051e-03
         46          47          48          49          50          51          52          53          54
1.701866e-02 2.525968e-10 1.701866e-02 5.392051e-03 2.525968e-10 2.525968e-10 2.525968e-10 2.525968e-10 2.517775e-02
         55          56          57          58          59          60          61          62          63
1.000000e+00 1.000000e+00 2.525968e-10 5.392051e-03 5.392051e-03 2.525968e-10 5.392051e-03 5.392051e-03 8.066887e-10
         64          65          66          67          68          69          70          71          72
5.392051e-03 2.525968e-10 5.392051e-03 5.392051e-03 5.392051e-03 5.392051e-03 4.468188e-02 2.525968e-10 2.525968e-10
         73          74          75          76          77          78          79          80          81
5.392051e-03 2.525968e-10 5.392051e-03 2.525968e-10 5.392051e-03 5.392051e-03 5.392051e-03 2.525968e-10 2.179261e-09
```

図 4.7:　fitted.values の表示結果

このときの ans$fitted.values と ans$linear.predictor をそれぞれ縦軸と横軸にしたプロットした図を表示する。

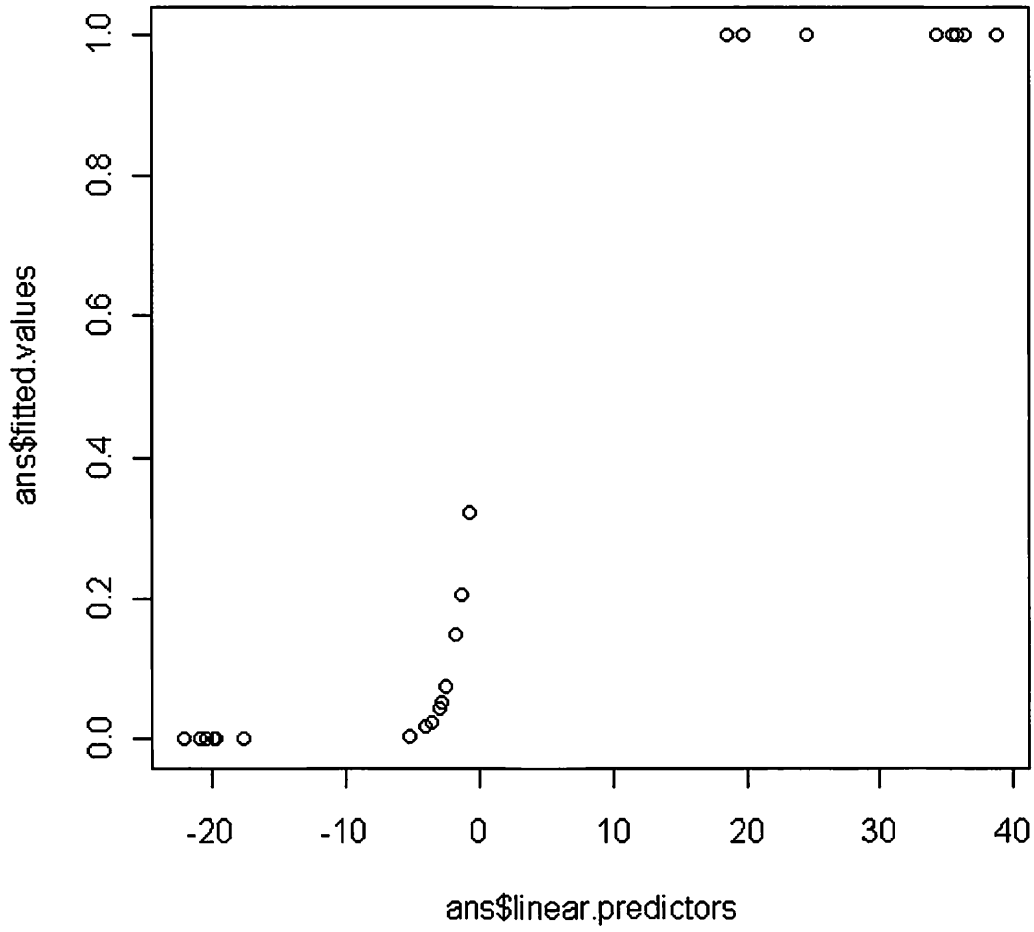

図 4.8: fitted.values と linear.Predictor のプロット図

このプロットした図を見ると、ロジスティック回帰分析のグラフ図と似ていることがわかる。

第五章 実験

5.1 使用するデータ

巨大掲示板「２ちゃんねる」で漫画「黒子のバスケ」に関するスレッドから、８１１個の記事を学習データとして使用した。

またテストデータとしては、同漫画の別のスレッドから７６０個の記事と、漫画「ゼロセン」に関するスレッドから８０５個の記事を使用した。それぞれテスト１・テスト２と呼ぶことにする。

なお、学習データ・テスト１はネタバレを書いても良いというルールのスレッド、テスト２はネタバレを書いてはいけないというルールのスレッドである。

5.2 実験概要

■手順１

この学習データである８１１個の記事のスレッドを３章の3.2で説明したように１記事ごとに８１１個のファイルに分け、一つ一つに６つの特徴判別を行う。

最初の学習データは７つ目の手作業で調べたネタバレかネタバレじゃないかの値も振り分ける

■手順２

７つの特徴分けが終わったら、４章の4.2で説明したようにＲでロジスティック回帰分析を行う。

b_1～b_6の値を導き出す。

■手順３

式(4.1)に手順２で出たb_1～b_6の値を代入してプログラムで計算した。
プログラムの方は、付録のＡ.2にＡ.2-Bを付け加えて行った。

その結果

```
1  4.65937210367599e-008
2  4.65937210367599e-008
3  0.5
4  0.5
5  0.5
6  0.5
7  0.5
8  0.5
9  0.5
10  1
11  4.65937210367599e-008
12  0.969985484723943
13  0.5
14  0.5
15  0.910070824175839
                .
                .
```

図5.1: 式(4.1)の計算結果の一部＜学習データ＞

左の番号は記事番号である。右の数値が独立変数yの値である。

この図で言うと、１０番目の記事が１となっており、ネタバレの可能性が高いということになる。

また、１，２，１１番目の記事の値にe-008とあるが、これは10の-8乗という意味なので、この３つの記事は非常に小さい値となる。よって、ネタバレの記事の可能性がとても低いことになる。

またテスト１とテスト２でも同じように計算してみた

```
1 4.65937210367599e-008
2 4.65937210367599e-008
3 0.5
4 0.5
5 0.5
6 0.5
7 0.5
8 0.5
9 0.5
10 0.5
11 0.5
12 0.5
13 0.5
14 0.5
15 0.826511139159529
          ・
          ・
```

図5.2: 式(4.1)の計算結果の一部＜テスト１＞

```
1 0.5
2 0.5
3 0.910070824175839
4 0.5
5 0.5
6 0.5
7 0.5
8 0.5
9 0.5
10 0.5
11 0.5
12 0.5
13 0.5
14 0.5
15 0.5
        .
        .
        .
```

図5.3: 式(4.1)の計算結果の一部<テスト２>

■手順３

ここで、独立変数yの値が0.9以上の時をネタバレとした。
0.9以上の時の記事番号を表示するプログラムで、ネタバレと判定された記事を確認した。プログラムは付録のA.2にA.2-Cを付け加え行った。

5.3 結果

実験において独立変数ｙが0.9以上、つまりネタバレと判断された記事と、その中での実際のネタバレ数を表5.1に示した。

表5.1:　実験結果①

データセット	ネタバレと判断された記事数	その中でネタバレの記事数
学習データ	68	19
テスト１	63	23
テスト２	73	10

手作業で検出したネタバレの記事数と、その中で実験でネタバレと判断された記事数を表5.2に示した。

表5.2:　実験結果②

データセット	実際のネタバレ記事数	その中でネタバレと判断された記事数
学習データ	23	19
テスト１	35	23
テスト２	17	10

これらを元に正解率、再現率、Ｆ値を求めた

表5.3:　実験結果③

データセット	正解率	再現率	Ｆ値
学習データ	0.279	0.826	0.417
テスト１	0.365	0.657	0.469
テスト２	0.137	0.588	0.222

第六章　考察

この実験の結果を見ると、再現率は0.826・0.657・0.588と全体的に非常に高いのに対し、正解率は0.279・0.365・0.137と全体的に低いことがわかる。

これを改善するために実験結果を一通り目を通したところ、第４章で行ったロジスティック回帰分析においてＶ５のEstimateの値がマイナスになっていることがわかった。ロジスティック回帰分析においてＶ５、つまり【『バレ』というワードが入っている】という５番目の特徴は、ネタバレの特徴にならないということがわかる。

これはおそらく、ネタバレ以外の記事でも『バレ』というワードが出すぎていたため、『バレ』というワードの重要性が薄れてしまったと思われる。

このことをふまえ、再実験を行った。

再実験

５番目の特徴を削り、６番目の【『みたい』というワードが入っている】という特徴を５番目の特徴とした。そして今までの実験と同じような流れで実験を行った。

６番目に手作業で調べたネタバレかネタバレじゃないかの値を入れ、第４章のようにＲでロジスティック回帰分析を行った。

```
> ans <- glm(V6 ~ V1 + V2 + V3 + V4 + V5, family=binomial(link="logit"),data$
> summary(ans)

Call:
glm(formula = V6 ~ V1 + V2 + V3 + V4 + V5, family = binomial(link = "logit")$
    data = c)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0178  -0.0921  -0.0921  -0.0921   3.3061

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.4608     0.5504  -9.921  < 2e-16 ***
V1            22.9223  1512.0321   0.015  0.98790
V2             1.7738     1.1577   1.532  0.12547
V3             2.7748     1.3274   2.090  0.03659 *
V4             2.2984     0.7532   3.051  0.00228 **
V5             1.2898     1.1194   1.152  0.24922
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

図6.1：Ｒによるglm関数（再実験）

図6.1のEstimateの値を見てわかるとおり、マイナスの値がなくなったのでＶ１～Ｖ５はすべてネタバレの特徴と成りうることがわかる。

このときのans$fitted.valuesとans$linear.predictorをそれぞれ縦軸と横軸にしたプロットした図を表示する。

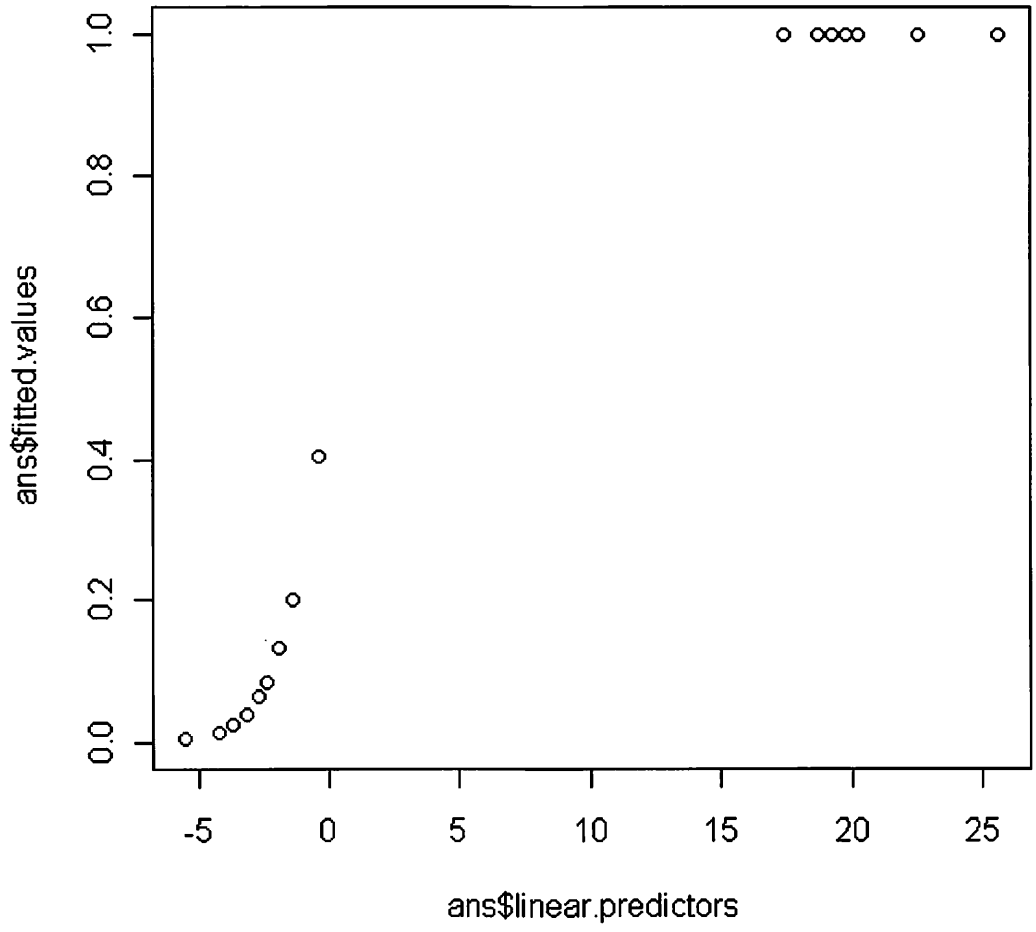

図6.2: fitted.valuesとlinear.Predictorのプロット図(再実験)

Estimateの値を用い、独立変数ｙを計算した。今回の場合、独立変数ｙが0.9以上の記事が多く出てしまっているので、0.95以上をネタバレとした。

５章と同じように、独立変数ｙが0.95以上、つまりネタバレと判断された記事と、その中での実際のネタバレ数を表6.1に示した。

表6.1:　実験結果①(再実験)

データセット	ネタバレと判断された記事数	その中でネタバレの記事数
学習データ	24	17
テスト１	27	23
テスト２	8	2

こちらも同じように、手作業で検出したネタバレの記事数と、その中で実験でネタバレと判断された記事数を表6.2に示した。

表6.2:　実験結果②（再実験）

データセット	実際のネタバレ記事数	その中でネタバレと判断された記事数
学習データ	23	17
テスト１	35	23
テスト２	17	2

これらを元に正解率、再現率、Ｆ値を求めた。

表6.3:　実験結果③（再実験）

データセット	正解率	再現率	Ｆ値
学習データ	0.708	0.739	0.723
テスト１	0.859	0.657	0.744
テスト２	0.25	0.118	0.160

このような結果が得られた。

学習データの正解率が0.365から0.708へ、テスト１の正解率が0.365から0.859へ
学習データの再現率が0.826から0.739へ、テスト１の再現率は変わらず
学習データのＦ値が0.417から0.723へ、テスト１のＦ値が0.469から0.744へ

学習データ、テスト１に関しては大幅な精度の向上が見られた。

しかし、テスト２に関してはネタバレと判断される記事が大幅に減少し、再現率が低下している。また、Ｆ値も下がってしまっている。

これは、テスト１とテスト２の違いが関係していると思われる。

テスト１の記事は『ネタバレ』を書いても良いというスレッドの記事であるのに対して、テスト２は『ネタバレ』を書いてはいけないというスレッドの記事なのである。テスト２のようなスレッドでもネタバレの投稿があるのだが、テスト１に比べるとネタバレを表立って書いているような記事が少ない。それにより独立変数ｙが基本的に少なくなってしまうのではないかと思われる。

また、独立変数ｙが0.95以上の中で、ネタバレのものとネタバレじゃないものの５つの特徴判別の値を表6.4と表6.5と表6.6に記した

表6.4:　学習データの特徴判別

ネタバレである	ネタバレでない
10110	00011
11000	01010
10010	00110
10000	01010
10001	01001
01010	00011
10000	00011
10010	00110
10000	
10001	
11000	
11111	
10100	
10110	
10000	
10000	

表6.5: テスト１の特徴判別

ネタバレである	ネタバレでない
10000	00011
10000	00011
11010	00011
10000	10000
10010	
10010	
10000	
10000	
10000	
10010	
10000	
10110	
11000	
11000	
10000	
10010	
10001	
10000	
10110	
10000	
10001	
10000	
01010	

表6.6:　テスト２の特徴判別

ネタバレである	ネタバレでない
10000	00101
10000	01011
	00011
	10000
	00011
	00011

この３つの表を見て、やはりネタバレの判別の大本になっているのは１番目の特徴ということがわかる。ネタバレのほとんどが１番目の特徴を持っている。

逆にネタバレでないのにネタバレと判別されてしまった記事の特徴を見てみると、主に４番目・６番目（新５番目）の特徴が多いことがわかる。これはデータリストを読み返すと、『だった』というワードはネタバレ記事でもよく使用されているが、過去の話題を出すときなどにも使用されることが多くあり、その記事をネタバレと判定してしまうことがある。『みたい』というのもネタバレ記事では何個か見かけるが、日常会話的に使われる場面も多いので、こちらもネタバレじゃないのにネタバレと判別されてしまうことが多々見うけられた。

（例）

藍本さんは帝光ではベンチだったんだよね？

青峰とはアゴンと雲水みたいな関係かな？

このことは、テスト２のデータを学習データとして実験を行った時にも判明したことである。

学習データに対し、テスト２はネタバレをしてはいけないスレッドであり、ネタバレの数もそれほど多くない。このテスト２を学習データとして、Ｒでロジスティック回帰分析を行うと

```
> ans <- glm(V7 ~ V1 + V2 + V3 + V4 + V5 +V6, family=binomial(link="logit"),$
> summary(ans)

Call:
glm(formula = V7 ~ V1 + V2 + V3 + V4 + V5 + V6, family = binomial(link = "lo$
    data = r)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.34746  -0.08243  -0.08243  -0.08243   3.37240

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.6831     0.6260  -9.078  < 2e-16 ***
V1             6.0744     1.4403   4.218 2.47e-05 ***
V2             5.6909     0.8185   6.952 3.59e-12 ***
V3             1.0380     1.8459   0.562    0.574
V4             1.3205     1.2466   1.059    0.289
V5             4.2725     0.8809   4.850 1.23e-06 ***
V6           -16.9500  1467.6086  -0.012    0.991
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

図6.3:　テスト２でロジスティック回帰分析

図6.3の結果を見るとＶ６、つまり【『みたい』というワードが入っている】という特徴がマイナスの結果となっている。これはやはり、ネタバレの記事以外でも多く使われていることが多いからだということがわかる。

全体的な考察としては、１番目の特徴はネタバレの特徴としてはかなり信頼性の高いものだということがわかった。１番目の特徴がある場合、ネタバレと判断されるものが多数あり、ネタバレでなかったのはこの実験では２個ほどしかなかった。

しかし、その分１番目の特徴だけ精度が良すぎ、他の特徴があまり影響されていないのも見てとれる。今回の実験では１番目の特徴がないネタバレで、正確にネタバレと判定できた記事は２個しかない。

これは他の特徴を考え直す必要がある。

第七章　終わりに

本研究ではロジスティック回帰分析を使い、ネタバレの文章を検出する実験を行った。しかし、ネタバレに特出した特徴というものがなく、検出は困難であった。

その原因としては、普通の過去の会話とネタバレの会話が似ている性質を持っていることである。過去の話などで『だった』や『みたい』のワードを多く使うからである。

そして、ネタバレかネタバレじゃないかの境界線が曖昧なことも原因の一つである。例えば先週の内容の感想の投稿も、先週の内容を見ていない人にとってはそれもネタバレとなってしまう。このような曖昧性によってもネタバレ検出の精度を狂わす。

またネタバレする時、最小限の文字数でネタバレしてくる人もいる。このような場合には、今回のネタバレの特徴では検出することは困難になる。

今後の課題としては、より精度を上げるためのネタバレの特徴の発見である。1番目のような特出した特徴が他にもあれば、精度は増すと考えられる。

謝辞

本研究の遂行及び論文の作成に多大な御助言及び指導を賜った　新納 浩幸 教官に深く感謝します。

また、本研究に助言や協力を頂きました、同研究室の 林 華氏、史 敏氏、 江口 晃氏にも深く感謝します。

参考文献

[1]青木繁伸： "Rによる統計解析 ", オーム社, pp180-181(2009)

[2]金明哲： "Rによるデータサイエンス ", オーム社, pp148-158(2007)

[3]" ロジスティック回帰入門"
<http://www.ibaraki-kodomo.com/toukei/logis.html>
(2009/10/13アクセス)

[4]平山るみ：" ロジスティック回帰分析", 京都大学教育学研究科
<http://www.educ.kyoto-
u.ac.jp/cogpsy/personal/Kusumi/datasem03/hirayama.files/frame.htm>
(2009/10/13アクセス)

[5]" perl入門"
< http://www.kent-web.com/perl/index.html>
(2009/11/10アクセス)

付録

プログラムソースリスト (perl)

スレッドの多くの記事を、それぞれ１つごとにファイル化するプログラムが必要である。

図A.1　１記事ごとにファイル化するプログラムソース

```
1.  #!/usr/bin/perl
2.
3.    while(<>){
4.      chomp;
5.
6.      if($_ =~ /^[0-9]* ：名/){
7.          push(@z,$x);
8.          $x = "";
9.      }else{
10.         $x.=$_;
11.     }
12.   }
13.
14. $filenumber = 1;
15.
16. for($i = 1;$i <= 811;$i++) {
17.     open(F, "> $filenumber.txt");
18.     print F @z[$i],"¥n";
19.     close(F);
20.     $filenumber++;
21. }
```

１つの記事のファイルに対して、６つのネタバレの特徴があるかないか検出するプログラムが必要である。

図A.2　ネタバレの６つの特徴を探すプログラム

```
1.  #!/usr/bin/perl
2.
3.  $space = " ";
4.  $find=0;
5.  $a=0;
6.  $b=0;
7.
8.  open(F, "> $filenumber000.txt");
9.      print F @z[$i],"¥n";
10.
11. for($g=1; $g<=811; $g++){
12.
13.         open(IN, "$g.txt");
14.
15.         while(defined($line = <IN>)){
16.
17.                     while ( ( $find = index ($line, "「", $find )
18.  ) >= 0 ) {
19.                                 $find++;
20.                                 $a++;
21.                     }
22.                     $find=0;
23.
24.                     while ( ( $find = index ($line, "『", $find )
25.  ) >= 0 ) {
26.                                 $find++;
27.                                 $a++;
28.                     }
```

```
29.
30.
31.                 if($a>=4){
32.                                 @x[0]=1;
33.                 }else{
34.                                 @x[0]=0;
35.                 }
36.         $a=0;
37.
38.         $find2 = index($line,"来週");
39.
40.         if($find2 >= 0){
41.                 $b++;
42.         }
43.
44.         $find2 = index($line,"次回");
45.
46.         if($find2 >= 0){
47.                 $b++;
48.         }
49.
50.         $find2 = index($line,"次週");
51.
52.         if($find2 >= 0){
53.                 $b++;
54.         }
55.
56.         if($b>0){
57.                 @x[1]=1;
58.         }else{
59.                 @x[1]=0;
60.         }
61.
62.         $b=0;
63.
```

```
64.         $find3 = index($line,"らしい");
65.
66.         if($find3 >= 0){
67.                 @x[2]=1;
68.         }else{
69.                 @x[2]=0;
70.         }
71.
72.         $find4 = index($line,"だった");
73.
74.         if($find4 >= 0){
75.                 @x[3]=1;
76.         }else{
77.                 @x[3]=0;
78.         }
79.
80.         $find5 = index($line,"バレ");
81.
82.         if($find5 >= 0){
83.                 @x[4]=1;
84.         }else{
85.                 @x[4]=0;
86.         }
87.
88.         $find6 = index($line,"みたい");
89.
90.         if($find6 >= 0){
91.                 @x[5]=1;
92.         }else{
93.                 @x[5]=0;
94.         }
95.
96.         for($i=0; $i<=5; $i++){
97.                 print F @x[$i];
98.                 print F " ";
```

```
99.             }
100.
101.            print F "¥n";
102.            }
103.    close(IN);
104.    }

        close(F);
```

また、最初の学習データでは手作業で調べたネタバレ記事番号がある。
その記事番号が来たときに、７番目の特徴を１に、それ以外には０が入るようにする。
これは次のプログラムをプログラムソースＡ.2の79行目に挿入することでできる。

図Ａ.2-A　7番目特徴を加えるプログラム

```
1.  if($g==10 || $g==55 || $g==56 || $g==83 || $g==264 || $g==324
2.  || $g==353 || $g==354 || $g==384 || $g==388 || $g==389 ||
3.  $g==403 || $g==420 || $g==444 || $g==474 || $g==511 || $g==613
4.  || $g==614 || $g==648 || $g==728 || $g==772 || $g==773 ||
5.  $g==783){
6.          @x[6]=1;
7.  } else {
8.          @x[6]=0;
9.  }
10.
11. for($i=0; $i<=6; $i++){
12.          print F @x[$i];
13.          print F " ";
14. }
15. print F "¥n";
```

式(4.1)に求めたb_1～b_6の値を代入し、独立変数yを算出するプログラムが必要である。

これは次のプログラムを、プログラムA.2の79行目に代入し、80～83行目を消すと実行することができる。

図A.2-B　独立変数yを算出するプログラム

```
1.  $p=(39.5049)* @x[0] + (1.5611)* @x[1] + (2.1549)* @x[2] +
2.  (2.3145)* @x[3] - (16.8818)* @x[4] + (1.1611)* @x[5];
3.
4.  $s =1 + exp(-$p);
5.  $ans =1 / $s;
6.
7.  print F $g;
8.  print F " ";
9.  print F $ans;
10. print F "¥n";
```

また、ネタバレは独立変数yが0.9以上としているので、0.9以上の時の記事番号を表示する必要がある。

これはさきほどと同じように、次のプログラムをプログラムA.2の79行目に付け加える。そして80～83行目を削除すると実行することができる。

図A.2-C　独立変数yのちが0.9以上の記事番号を表示するプログラム

```
1.  $p=(39.5049)* @x[0] + (1.5611)* @x[1] + (2.1549)* @x[2] +
2.  (2.3145)* @x[3] - (16.8818)* @x[4] + (1.1611)* @x[5];
3.
4.  $s =1 + exp(-$p);
5.  $ans =1 / $s;
6.
7.  if($ans>0.9){
8.          print F $g;
9.          print F "¥n";
10. }
```