Webサイトにおけるアクセス者のサイト離脱とサイト内遷移

金　子　武　久

1．はじめに

1－1．研究意図

今日，インターネットは人々の生活の中に普及し，インターネット上のショッピングサイト（以下，Webショッピングサイト）では書籍や雑誌を筆頭に，宿泊・航空・鉄道チケット，衣料・アクセサリー，食料品，アルコール類，コンピュータソフトウェア，家具，CD・ビデオ・DVDなど実に様々な商品が販売されるようになってきている。中には，自動車やバイク，一戸建て住宅，土地なども扱われている。インターネットにはWebショッピングサイトだけではなく，企業の広報活動を担った企業Webサイトも多数存在する。そこでは取扱商品の情報提供，各種イベントの情報提供，資料請求や問合わせの窓口，など様々な活動が展開されている。

こういったWebサイトには商品の販売や情報提供などの何らかの目的が存在し，その目的達成のためにアクセス者に対して様々な誘導が行われている。例えば，商品販売を目的とした場合，アクセス者が特定の商品の注文を確定した段階で一つの目的が達成されることになるが，Webサイトはそこに到達するために，アクセス者に理解しやすい情報提供を行い，安心して快適な買物ができるように様々な工夫を凝らす。特定の色配列をとることによってクリックボタンを目立たせたり，アクセス者の注意を引くためにインパクトのある絵柄を組み入れたりしてアクセス者を導いていく。また，アクセス者がWebサイト内のどこにいるのか不明になったり，次に何をすべきなのか困惑したりしているとき，適切なアドバイスを提供することでアクセス者をナビゲートしていく。こういったナビゲートがうまくいけばアクセス者はページ遷移を繰り返していくであろうが，ナビゲートがうまくいかなければ，アクセス者はそのWebサイトを離脱してし

まうであろう。

本論文ではWebサイト内でアクセス者を導いていく様々なWebページ属性を検討し，それらがサイト内のページ遷移とサイト離脱とどう関係しているかを分析することが目的である。

1－2．研究の位置づけ

本研究はWebマイニングという研究分野に位置づけられる。Webマイニングには(1)Web内容マイニング，(2)Web利用マイニング，(3)Web構造マイニングの主要な研究分野が存在すると指摘されている。Web内容マイニングとは，Webページに含まれる文章の内容を自動的に発見することを意図している。Web利用マイニングとは，Webサイトへのアクセス者によるアクセスパターンを自動的に発見することを狙っている。また，Web構造マイニングとはWebページのリンク構造に関するパターンを自動的に発見することを目指している（詳細は武田等訳(2004，pp102-114)，Srivastava（2000）を参照)。本研究は，Webページに含まれる文章からキーワードを抜き出してデータ化しているのでWeb内容マイニングと関連がある。また，アクセスログデータの解析を行っているのでWeb利用マイニングとも関連が深い。さらに，Webページ間のリンクについてもデータ化して扱っているので微視的な視点でのWeb構造マイニングにも関連がある。

インターネット上にはWebマイニングに関する数多くの論文が公開されている。これらを集積して論文リストを作成・公開しているサイトがある（例えばCYBERmetricsを参照)。およそ70篇の論文を入手して先行研究を調べたがそこには本研究と類似したものは存在しなかった。

2．分析用のWebページ属性

アクセス者を導いていくWebページ属性には様々なものが考えられる。Webページを構成している多種多様な素材（画像，文字列など）には，その形，その色，そのサイズ，その配置などの特徴がある。こういったものがすべてアクセス者を導いていくことに利用されるわけであるが，これらをすべてデータとして把握することには困難が伴う。また，本研究では統計的なデータ解析を意図しているので，Webページを可能な限り数字として把握したいという思惑もある。そこで，今回の研究で取り扱うWebページ属性としては図表2－1のようなものを取り上げた。

●当該ファイルサイズ

ここでの当該ファイルとはアクセスされるWebページそのものを指す。通常はHTMLで記述されているため，その各種タグが多数埋め込まれている。凝ったページであるほど埋め込まれているHTMLタグが多くなる。従って当該ファイル内での記述内容が多くなり，自然とファイルサイズが大きくなる傾向にある。また，当該ファイルにはコンテンツとしての文字列も同時に埋め込まれている。豊富なコンテンツであるほど通常は文章が長くなる傾向にあるため，これによっても当該ファイルサイズは大きくなる。

図表2－1　分析に用いる変数とその意味

変数名	意味
当該ファイルサイズ	当該Webページのファイルサイズをバイト単位で表したもの
リンクhtmlファイル数	当該Webページから次ページや他サイトへのリンク数
リンク画像ファイル数	当該Webページに貼り付けられている画像ファイルの数
リンク画像ファイルサイズ	当該Webページに貼り付けられている画像のバイト単位のファイルサイズ
その他リンクファイル数	当該Webページにリンクづけられているその他（例えば音声）のファイル数
その他リンクファイルサイズ	当該Webページにリンクづけられているその他ファイルのバイト単位のファイルサイズ
画像領域面積	当該Webページに含まれる画像領域の面積を平方ミリメートルで表したもの
文字領域面積	当該Webページに含まれる文字領域の面積を平方ミリメートルで表したもの
文字領域面積構成比	文字領域の面積の構成比を百分率で表したもの
単語種類数	当該Webページに含まれる文字列から単語を抜き出し，その単語種類の数

この研究が行われた2001年当時はまだブロードバンド時代が到来する以前であり，通常の電話回線を使ったダイヤルアップ方式によるインターネットアクセスが一般的であった。その頃は回線速度が遅く，画像のようなファイルサイズの大きなデータを受信するのにかなりの時間がかかったものである。

当該ファイルサイズが大きくなることによって，アクセス者側のブラウザがサーバーから送らてくる各種情報を処理するのに時間がかかり，Webページを表示させるボタンやリンクをクリックしてから実際にブラウザによって表示されるまでに時間がかかり，アクセス者にストレスをかけてしまう可能性がある。また，文章が大量に埋め込まれている場合は，アクセス者がその膨大な情報に困惑してしまい，ストレスを感じてしまう可能性もある。

以上のような検討により，当該ファイルサイズが大きいほどアクセス者にストレスをかけることになり，これによってアクセス者がサイトを離脱してしまうことが考えられる。

●リンクhtmlファイル数

これは当該Webページに埋め込まれている他ページへのリンクの数である。リンク先は当該Webサイト内の他ページであるかもしれないし，当該Webサイト外の他ページであるかもしれない。しかしながら，現段階では両者の区別を行っていない。これは今後の課題である。暗黙的に，リンク先は当該Webサイト内の他ページになっていることを想定している。

他ページへのリンクがゼロである場合，そのページに一度アクセスすると抜け出せなくなってしまう。ブラウザの「戻る」ボタンを使って強制的に前ページに戻らなければならない。また，他ページへのリンクが少数の場合，そのリンクを見つけることが困難となり，実質的にはそのページから抜け出せなくなったとアクセス者に誤認される可能性がある。行き場を失ったアクセス者はそこで当該サイトを離脱してしまうことが考えられる。

●リンク画像ファイル数，リンク画像ファイルサイズ

これらは当該Webページに貼り付けられている画像ファイルの数とそのバイト単位のサイズである。大きな画像はアクセス者にストレスを課し，これによってサイト離脱をもたらしてしまう可能性が考えられる。

●その他リンクファイル数，その他リンクファイルサイズ

ここでのその他リンクとは，主として音声データへのリンクを指す。前者は音声データへのリンクがいくつあるかを表したもの，後者はリンク先の音声データのファイルサイズである。先の画像と同様に，これらは一般的に大きなファイルになりがちである。その受信には時間がかかることとなり，アクセス者に過大なストレスを課すことにもなりかねない。音声データの受信中にストレスを感じたアクセス者が当該サイトを離脱してしまうことが考えられる。

●画像領域面積，文字領域面積，文字領域面積構成比，単語種類数

これらは文字通りの意味をもった変数である。画像も文字もアクセス者に対して何らかの意味ある情報を伝えるものであるが，これが多くなるとアクセス者に過大な負荷を与えることになる。アクセス者の処理能力を超える膨大な情報量の文字や画像が提示されると，アクセス者はストレスを感じ，これによって当該サイトを離脱してしまうかもしれない。情報量の大きさを画像領域の面積や文字領域の面積で代替的に把握しようというわけである。文字領域面積構成比や単語種類数はこれと同様の趣旨で取り上げられた。

次章では，Webページ属性を自動的に抽出するための準備，上の項目の順番に従いつつ，どのようにWebページ属性を抽出しているのかを説明する。

3．Webページ属性の自動抽出

Webサイトを構成するページは数百から数千ページに及ぶこともある。このような多数のWebページについてその属性を調べ，それを人手によってデータ化していく作業はあまりに非現実的である。そのため，本研究ではWebページ属性を抽出するにあたり，それを自動的に行うプログラムを開発した。本節では，この自動抽出プログラムについて概説する。htmlファイル等からページ属性を自動的に測定するために今回開発したプログラムはPerlというコンピュータ言語を用いて作成した。説明をより具体的にするために，必要に応じてPerlの文法を用いている部分がある。

3－1．Webページ属性抽出のための準備

3－1－1．必要なファイル

ページ属性を抽出するWebサイトを構成しているすべてのWebページファイルとすべての

画像ファイル，及び，すべての音声ファイルなどをしかるべきディレクトリに事前に保存しておく。

Webショップなどを運営しているサイトでは，そのWebサイトの顧客情報や取引情報を格納したデータを持っていることが多い。そのようなサイトでは，アクセス者からの問い合わせに応じてデータベースにアクセスし，そのアクセス者の氏名，住所などの顧客情報や過去の購入商品リストなどの該当するデータを参照・処理してWebページに表示することがよくある。また，アクセス者の過去の購入商品分野から適当な推奨商品リストを構成し，それをWebページに表示することもある。

アクセス者に固有のこのようなWebページはWebサーバーによるCGIやASPの機能を利用したプログラムによってリアルタイムに構成されているものであるが，現在のページ属性自動抽出プログラムではそのようなWebページにはまだうまく対応できてはおらず，アクセス者固有の情報を無視するという対応になっている。

3－1－2．画像ファイルからの文字列抽出

現在のところ，画像ファイルからの文字抽出は自動的には行なわれない。事前に人間が画像データを1つひとつ目で確認し，画像ファイル名とそこに含まれる文字列との対応表を作成しておく必要がある。その対応表はimages.txtというファイル名でしかるべきディレクトリ（後述）に保存しておかなければならない。

ページ属性の自動抽出プログラムでは，文字領域の面積を計算したり画像領域の面積を計算したりする際に，このimages.txtを参照し，それらの数値の微調整をしている。

例えば，あるWebサイトのxxxx/imagesディレクトリに含まれる画像ファイルに対して，images.txtというファイル名で次の図表3－1のようなテキストファイルを事前に作成しておく。なお，現在の自動抽出プログラムではimages.txtという名前は固定しておく必要があり，変更することはできない（もちろん，ページ属性抽出プログラムを書き直せばこのファイル名を変更

図表3－1　画像ファイルに含まれる文字列のテキストファイルへの書き出し

```
# カンマ区切で3つのフィールドに分かれる
# 第1フィールド：画像ファイル名，
# 第2フィールド：文字列だけフラグ，
#                    0＝はい（文字列だけ）　1＝いいえ（絵や写真も含まれる）
# 第3フィールド：ファイルに含まれる文字列
#（注意）文字列や絵や写真もなく空白だけの場合は第2フィールドに0を入れること
#（注意）含まれる文字列が何もない場合は第3フィールドには改行のみを入れること
goods1.gif, 0, 雑貨
goods2.gif, 0, 食品
bottan.gif, 0, 商品リスト
buy.gif, 0, ご注文
syou.gif, 0, 詳細
menu1.gif, 1, メニュー
```

することは可能)。また，画像ファイルが格納されているディレクトリには必ず1つのimages.txtを作成しておかなければならない。

ファイルの前半部分に＃記号で始まる行が複数存在しているが，このような＃で始まる行はコメントとして自動抽出プログラムからは無視される。もちろん，このようなコメントが必要なければ何も記入しなければよい。

このファイルは，カンマ区切りでフィールドが分かれており，第1フィールドはそのディレクトリに含まれる画像ファイル名が入る。第2フィールドには，その画像ファイルが文字だけを含んでいるか，あるいは絵や写真が含まれているかを表すフラグが入る。画像ファイルを表示させた場合，それが文字だけの場合は0，絵や写真が含まれている場合は1を入れる。第3フィールドには，画像ファイルに含まれる文字列部分を入れる。

上の例では，＃行以外の最初の行に，

goods1.gif, 0, 雑貨

とある。これは，第1フィールドがgoods1.gif，第2フィールドが0，第3フィールドが「雑貨」，となっていることを表している。つまり，goods1.gifという画像ファイルには文字だけで絵や写真が含まれておらず，その文字列は「雑貨」であることを示している。

また，最下行には

menu1.gif, 1, メニュー

とある。この第1フィールドにはmenu1.gif，第2フィールドは1，第3フィールドは「メニュー」となっている。これらは，menu1.gifという画像ファイルには絵や写真が含まれており，加えて文字列として「メニュー」が含まれていることを表している。

以上までのimages.txtを画像ファイルが存在するディレクトリ毎に用意しておけばよい。従って，画像ファイルを格納しているディレクトリが複数存在している場合には，そのディレクトリ毎に1つのimages.txtファイルが必要となる。

3－2．Webページ属性自動抽出の流れ

3－2－1．Webサイトを構成するファイルリストの作成

●ファイル群の定義

Webサイトを構成する各種ファイル群を定義しておく。ファイルの種類としては，htmlファイル類，画像ファイル類，その他ファイル，としている。htmlファイル類には拡張子が*.html, *.htm, *.aspとなっているものが含まれる。画像ファイル類としては，拡張子が*.gif, *.jpgとなっているものが含まれる。その他ファイルとしては，拡張子が*.wavとなっているものが含まれる。これらの定義は必要に応じて変更することができる。

●ファイル群別のファイルリストの作成

Webサイトのトップディレクトリからのパス名付きで各種ファイルリストを作成する。UNIX

系のファイルリスト表示コマンドlsを利用してファイルリストを取得する。その際，lsコマンドに再帰的なファイルスキャンを行わせるために-Rオプションをつけて起動する。すると，パス名とそこに含まれるファイル名の一覧が取得できるようになる。これをPerlの内部から呼び出し，lsコマンドの結果をPerlの変数に格納する。その格納はPerlのハッシュ配列（連想配列）に行ない，ファイル名をキーとし，そのキーに対応する値として1を代入しておく。

例えば，$existhpagesize{'/index.html'}=1のようにしてキーとそれに対応する値を格納する。この場合のハッシュ配列の名前は%existhpagesizeとなる。なお，ハッシュ配列名の先頭の%記号はexisthpagesizeがハッシュ配列であることを示すPerlの言語上の仕様である。また，そのハッシュ配列にキーを通じて参照する場合にはハッシュ配列名の先頭記号が$印になる（例えば，$existhpagesize{'/index.html'}）。ハッシュ配列を作成した後で，%existhpagesizeというハッシュ配列に対して'/index.html'というキーを与えると1という値が帰ってくることになる。

初めはこのようにキーに対して1という値を保持しているが，後でこのハッシュ配列に対して任意のファイル名をキーとして与えてそのファイルサイズを値として再代入している。

ハッシュ配列は文字列をキーとしてそれに対応する値を検索するのが高速なので，様々なプログラムで頻繁に利用されている。

こういったハッシュ配列をhtmlファイル群，画像ファイル群，その他ファイル群のそれぞれで作成しておく。それぞれ，%existhpagesize，%existpictsize，%existetcsizeという名前でハッシュ配列を使っている。

3－2－2．各種ファイルのファイルサイズの抽出

直上で説明したhtmlファイル群，画像ファイル群，その他ファイル群のそれぞれのハッシュ配列に関して，次のような手続きでファイルサイズの抽出と格納を行なう。

例えば，htmlファイル群のハッシュ配列%existhpagesizeに対して，まず，その配列に含まれるファイル名（パス名付き）を取得する。続いて，そのパス名付きファイル名を引数としてPerlのstat関数を呼び出す。Perlのstat関数は引数として指定されたファイルについての様々な指標を出力してくれる。その中のひとつにバイト単位で測定されたファイルサイズがあるので，これを取得する。

パス名付きファイル名と取得されたファイルサイズとを格納するために同じハッシュ配列に対してパス名付きファイル名をキーとして指定し，そのキーに対応する値としてファイルサイズを格納する。例えば，

$existhpagesize{'/index.html'} = (stat('/index.html'))[7]

とすると，%existhpagesizeというハッシュ配列のキー/index.htmlというWebページに対してそのファイルサイズが格納されることになる。なお，この式の右辺はPerlによるファイルサイズ取得の書式例である。

以上と同様の処理を画像ファイル群とその他ファイル群のそれぞれに適応することで，パス名

付きファイル名とそのファイルサイズとの対応を示すハッシュ配列が得られることになる。このような手続きによって，先程までは値が1であったハッシュ配列に値としてファイルサイズが再格納されることになる。

3－2－3．各種ファイルのリンク数抽出

Webページを構成するhtmlファイルには様々なファイルがリンクづけられている。リンクされているものが<a href="xxxxx">のようになっている場合は，他ページ（あるいは他サイト）のxxxxxへのリンクとして表示されることになる。

リンクされているものが<img src="yyyyy.gif">のようになっている場合は，画像yyyyy.gifがそのページに貼り付けられて表示されることになる。

現在のプログラムでは，<a>タグの属性がhrefとなっている場合はそのhrefの値を他ページへのリンク先として取得している。しかしながら，実際のところこれだけのルールではWebページに直接埋め込まれたjavascriptなどの他のプログラム部分を拾ってきてしまうことがあるので，href属性の値を取得した後に若干のゴミ掃除をする必要がある。

また，当該ページに貼り付けられている画像ファイル名として<img>タグのsrc属性で指定されているyyyyy.gifというような値を取得している。

<a>タグ，<img>タグ以外で貼り付けられているものはその他のリンクファイルとしてそのファイル名を取得している。

ところで，このようにしてリンクファイル名を取得する際には，htmlファイルに含まれるタグを解析しなければならない。HTMLはテキスト中に何らかの機能をもったタグを埋め込むことでページを記述するコンピュータ言語であり，それは改行を無視し，アルファベットの大文字小文字を無視し，1個以上のスペースは100個でも1,000個でも1個と同じに扱ったりするなど，かなり自由度の高い書き方が可能である。そこで，HTMLのタグを識別し，その属性と値を取得するためにPerlなどで利用可能な正規表現が威力を発揮する。しかしながら，HTMLのタグ取得にぴったり合った正規表現を使わないと思わぬゴミを取得してしまうことになるため，その利用にはかなりの注意が必要となる。そこで，今回のWebページ属性自動抽出プログラムでは，Perlの拡張モジュールとして定評のあるHTML::LinkExtorを利用した。このモジュールにはhtmlファイルの中かから各種タグ，その属性，その値を抽出する関数が含まれている。なお，このHTML::LinkExtorはCPAN（Comprehensive Perl Archive Network，http://www.perl.com/CPAN/）から入手可能である。

3－2－4．文字列の抽出

● htmlファイルからの文字列抽出

htmlファイルは，文字列としてのHTML自身と，そのページのコンテンツを記述した文字列で構成されている。htmlファイルの中からHTMLのタグ関連の部分とそれ以外のブラウザに表

示される有意味な文字列とを区別して把握しなければならない。htmlファイルの中からブラウザに表示される文字列を取得するには，Perl拡張モジュールのHTML::TreeBuilderが利用可能なようであるが，これはまだ機能限定版で表やフレームには対応していないとのことなので，今回はこれを利用していない。代わりに，Lynxというテキストブラウザを利用し，これによってHTMLのタグ関連部分とそれ以外の文字列との区別をしている。Perlの内部からLynxを呼び出し，その際に引数としてパス名付きhtmlファイル名を渡し，更に，起動オプションとして「-dump-force_html」をつけて起動させる。これによって，htmlファイルの中のブラウザで表示されるコンテンツとしての文字列部分が取得できるようになる。しかしながら，Lynx固有のテキストメッセージなども含まれるため，ページ属性自動抽出プログラムでは，後で不必要な文字列を掃除している。

なお，LynxはWWWの初期の頃に利用されていたブラウザであり，テキストのみを表示し，画像はすべて無視する仕様になっている。現在ではこれをメインに利用するアクセス者は極少数であろうが，画像が必要でなく，キーボードで素早いページ遷移をしたい場合には現在でも利用されることがある。

以上のようにしてLynxを利用して抽出された文字列を何らかの変数に格納しておく。

●画像ファイルからの文字列抽出

画像ファイルの中にも文字列が含まれていることがよくある。続いて画像ファイルに含まれる文字列を取得する。そのためには，当該ページに貼り付けられている画像ファイルについて，パス名付き画像ファイル名のリストを作成しておく。そのリストに含まれる画像ファイルが保存されているディレクトリに用意されているはずのimages.txt（3－1－2節を参照）を調べ，貼り付けられている画像ファイルに含まれている文字列を取得する。

3－2－5．当該ページの文字領域面積の抽出

htmlファイルから取得された文字列は，17インチのモニターで，解像度1024×768ドットのもとでのデフォールトサイズの半角1文字を横幅2.07mm，高さ3.75mmとし，全角は横幅を4.14mm，高さ3.75mmとして扱っている。

取得された文字列の長さを半角基準で計算し（これはPerlのlength関数で取得する），取得された半角文字数×2.07×3.75で文字列部分の面積を計算している。

また，画像ファイルから取得された文字列についても同じモニター解像度のもとで半角1文字の横幅を1.54mm，その高さ3.00mmとし，画像ファイルに含まれる文字列部分の面積を，取得された半角文字数×1.54×3.00で計算している。

以上のような半角1文字の横幅や高さは実際にモニターに表示された全角文字20文字5行分程度をノギスで実測し，その平均値によって全角1文字分の横幅と高さを決定し，半角は横幅を全角の半分とした。Webページは，プロポーショナルな文字表示が一般的であり，全角文字の横

幅半分が半角文字の横幅と等しくはない。さらに，半角文字はその文字種によって横幅が個々に異なる。従って，現在の文字領域面積の計算された値は概算的なものであることをここに強調しておく。

html に含まれる文字列部分の面積に，そのページに貼り付けられている画像ファイルに含まれる文字列部分の面積を加算して，当該ページの文字領域面積としている。

当該ページに貼り付けられている画像ファイルのリストからパス名付きの画像ファイル名を取得し，その画像ファイルが保存されているディレクトリに用意してあるはずの images.txt ファイルを参照する。images.txt ファイルの第2フィールドが0の場合はその第1フィールドに記載されている画像ファイルには文字列だけしか含まれていないことを表しており，またそのフラグが1の場合にはその画像ファイルには文字列の他に絵や写真も含まれることを表している。この第2フィールドの0/1フラグを利用して，画像領域処理を分岐させている。

第2フィールドが0となっている画像ファイルは絵や写真を含んでいないことを表しているので，その場合は，画像とはみなさない。一方，第2フィールドが1となっている場合には少なくとも絵や写真が含まれているのでそのファイルを画像ファイルとみなすことにしている。

ところで，第2フィールドが1であるからといって，そのファイルには絵や写真だけがふくまれていることを意味しているわけではない。その場合でも文字列を含んでいる場合がある。あるファイルに絵や写真とともに文字列が含まれている場合は，現在のプログラムでは，そこに含まれる文字列が画像の一部として扱われると同時に文字列としても扱われるように作成してある。

現在のページ属性自動抽出プログラムでは文字列単位の文字サイズを考慮していない。一般的に文字列はその役割に応じて様々な大きさで表示される。見出し，小見出し，本文，などではほとんどの Web ページで異なる文字サイズが用いられている。文字列部分の大きさは <Hn> タグや <font> タグなどによって指定されているので，その値を自動的に抽出することは可能である。自動抽出プログラムの中で様々な文字列にそれぞれの文字サイズを関連づけておくことも可能である。今後は，文字列単位でその大きさを取得し，適当な解像度とフォントを仮定した上で，文字領域面積や文字領域面積構成比の値を計算するようにプログラムを改良する必要があろう。

3－2－6．当該ページの画像領域面積の抽出

ある html ファイルの中に <img src="yyyyy.gif" width=200 height=100> のような記述があった場合，そのページには yyyyy.gif という画像ファイルが貼り付けられており，その表示上の大きさは横幅200ピクセル，高さ100ピクセルであることをこの記述は示している。

この記述の中から width の数値と height の数値を抜き出し，表示上の画像面積を算出している。1ピクセル0.3mm で計算している。なお，この1ピクセルの大きさは，17インチモニターで解像度1024×768ドットのもとで，実際にいくつかの画像をブラウザ上に表示させ，その横幅と高さをノギスで実測し，その画像の width と height のピクセル数でそれぞれ割り算し，平均的な値として1ピクセルを0.3mm とした。

以上のようにして取得されたwidthとheightの数値にもとづいて，画像面積＝width×height×0.3×0.3として計算している。

htmlファイルには複数の画像ファイルが貼り付けてあることが多いので，そのページに貼り付けてあるすべての画像ファイルについて上と同様の計算をし，すべての画像の面積を合計することで当該ページの画像領域面積としている。

ところで，現在のページ属性自動抽出プログラムはこのような画像サイズの記述が1行でなされていることを前提にして作られている。htmlファイルの中で，

```
<img src＝"yyyyy.gif" 改行
 width＝200 改行
 height＝100 改行
>
```

のように複数行にわたる形式で記述してある場合は画像面積の取得に失敗することになる。今回，事例研究で利用したWebサイトではこのような複数行にまたがる記述はなかったが，他のサイトではあり得る記述形式なので，今後はWebページ属性の自動プログラムでもきちんと対処する必要がある。

3－2－7．当該ページに含まれる単語種類数の抽出

●日本語文章の形態素への分割

西洋系の言語，例えば英語では，文章を構成する単語はスペースによって区切られているため，英単語を切り出すことは比較的容易である。しかしながら，日本語の文章から日本語の単語を抜き出すことはかなりの困難が伴う。

日本語の文章から，それを構成する形態素に切り分けることを「分かち書き」というが，これを行なうソフトウェアとして「茶筌chasen」と「kakasi」が有名である。後者はWebページなどでも利用されることの多い日本語全文検索システムNamazuの前処理を行なうソフトウェアとしてデフォールトで利用されている。分かち書きには日本語辞書が大きな役割を果たすが，辞書が比較的小さく，処理が速く，十分実用的な結果を得ることができるということで，今回の研究ではkakasiを利用している。

例えば，「Ｗｅｂページのページ属性を自動的に抽出するプログラムを開発しました。」という文章をkakasiにかけて処理すると，次のような分かち書き結果を得る。

「Ｗｅｂ＿ページ＿の＿ページ属性＿を＿自動的＿に＿抽出＿する＿プログラム＿を＿開発しました＿。」

一繋がりの日本語文章がスペース区切りで形態素に分けられるが，ここでは見やすいようにスペースの代わりに下線＿で表示してある。

●日本語としての単語の生成

形態素へと分かち書きしたままでは語句としての意味が失われてしまい，Webページのコンテンツを表すものとしては都合が悪い。そこで，形態素の中から文章の意味を表す重要な単語を抜き出す必要がある。上の文章例では，次のようになろう。「Ｗｅｂページ，ページ属性，自動的，抽出，プログラム，開発」。ここでは，文章中の助詞などのひらがな部分を削り，「Ｗｅｂ」と「ページ」という語句を繋げて「Ｗｅｂページ」という単語を作ったり，「ページ」と「属性」という語句を繋げて「ページ属性」という単語を作ったりしている。

日本語文章をkakasiによって形態素へと分かち書きした後，一定のルールに従って語句を再結合させて有意味な単語を構成している。現在のところ，完全ではなく簡便なルールとして次のように処理している。

➢形態素の3語句続きについての例

中黒「・」で挟まれたカタカナ語句を結合してひとつの単語にする

「マーケティング」「・」「マネジメント」→「マーケティング・マネジメント」

➢形態素の2語句続きについての例

カタカナ語句の後に漢字語句が続く場合は結合してひとつの単語にする。

「マーケティング」「戦略」→「マーケティング戦略」

漢字語句の後にカタカナ語句が続く場合は結合してひとつの単語にする。

「戦略的」「マーケティング」→「戦略的マーケティング」

全角アルファベット語句の後にカタカナ語句が続く場合は結合してひとつの単語にする。

「Ｗｅｂ」「マーケティング」→「Ｗｅｂマーケティング」

➢形態素1語句の例

「Ｗｅｂ」→「Ｗｅｂ」（前後に漢字語句やカタカナ語句が無い場合，そのまま）

「ページ」→「ページ」（前後に漢字語句やカタカナ語句が無い場合，そのまま）

「わたし」→「」（ひらがなをすべて無視）

以上のような簡便ルールによって日本語の単語を生成し，当該Webページを構成する単語の種類がいくつあるかをカウントしている。これが当該Webページの単語種類数となる。

ところで，現在のところ利用していないが，Webページのコンテンツを表すものとして単語そのものを値とする変数も用意してある。当該Webページにどのような単語が掲載されているかがパス名付きページファイル名をキーにして参照することができる。例えば，トップページ'/index.html'にどのような単語が含まれているかが，$keywoard{'/index.html'}の値としてカンマ区切りで単語リストが得られる。ページ別のこのような単語リストは，ページ毎にリストの長さが異なり（単語の数が異なり），しかも数値属性ではないので，分析に利用することがなかなか困難である。そのため，将来的な研究のための準備としてプログラム内に用意はしてあるが，現

在のところWebページ属性として利用することはしていない。

3－2－8．文字領域面積構成比

これまでに説明した文字領域面積と画像領域面積を用い，次のように文字領域面積の構成比を計算している。

$$\text{文字領域面積構成比}=\frac{\text{文字領域面積}}{\text{文字領域面積}+\text{画像領域面積}}\times 100$$

Webページは，表面上，文字領域と画像領域と背景によって構成されている。従って，より厳密には上の式の分母には，見えている背景部分の面積をも加算すべきである。しかしながら，背景部分の面積はブラウザをどの程度の大きさで開いているかによって大きく異なるものになってしまい，背景部分の面積を特定することが困難である。今回の研究では背景部分の面積を無視して文字領域面積の構成比を算出している。

4．離脱型Webページと遷移型Webページの差異分析

4－1．離脱率の高低差を検定するシステム

なんらかのビジネス目的でWebサイトを運営する側にとっては，サイトに訪れたアクセス者がどのWebページを閲覧し，どのようなページ遷移をしたのかを把握することは今後のWebサイトの設計改良へ向けて大きな基礎を与えてくれるものと思われる。同様に，アクセス者がページ遷移を繰り返し，サイトを離脱していった場合，その離脱がどのような要因によるものなのかを把握することもWebサイトの改良にとって重要な情報をもたらしてくれるものと思われる。まずは離脱要因を取り除くことがアクセス者志向のWebサイトを構築するための第一歩と考えられるからである。そこで，ここでは離脱要因の候補としてこれまで扱ってきた10個のWebページ属性変数を取り上げ，これらがサイトの離脱とどのような関係になっているのかを調べることにする。分析の流れは図表4－1にまとめてある。なお，これ以降，離脱型Webページは離脱率の多いWebページとして捉え，遷移型Webページは離脱率の少ないWebページと捉えることにする。

以上の分析ステップをスムーズに処理するため，分析システムを構築した。次項でそれを例示する。なお，ステップ2からステップ4は反復過程であるが，構築したシステムでは一括処理して結果を表示するようにしている。

4－2．事例研究

4－2－1．データの選択と離脱率高低の基準設定

以下では実在したWebショッピングサイトを事例とした分析結果を示したい。そのWebサイトはアルコール飲料を専門に輸入・製造販売するところであったが2002年頃に閉鎖され，現在では別形態で運営されている。ここでのデータは2001年を中心とした2年間ほどのアクセスログ

図表4－1　分析の流れ

ステップ1	サイトを構成するWebページを離脱率の多いWebページ（A群，離脱型Webページ）と離脱率の少ないWebページ（B群，遷移型Webページ）とに2分割
ステップ2	あるページ属性変数に関して，離脱の多いWebページ（A群）の平均値と少ないWebページ（B群）の平均値を計算
ステップ3	2群の平均値の差の検定
ステップ4	ステップ2に戻り，他のページ属性についても同様に検定
ステップ5	すべての属性について検定を行った後，離脱の多いWebページと少ないWebページとでどのページ属性が関係していそうかを検討

データとその頃のWebサイトを構成する各種ファイル群である。

このような分析を行った先行研究は見当たらなかったため，これから行う分析は試行錯誤による探索的なものになる。そのため，探索的な分析が行いやすいように専用のシステムを構築した。その分析システムはWebサーバー上で動作するCGIとしてデータ選択，各種設定，差異分析を行うもので，クライアントマシンのブラウザを通して分析ができる仕組みとなっている。

分析にあたって，まずは，対象となるログデータを指定する。「整備後のそのままのログデータ」，「初セッションのみ」，「ゴール到達後削除」，「ゴール到達セッション削除」，など，いくつかの視点でデータを削ったものを用意した。

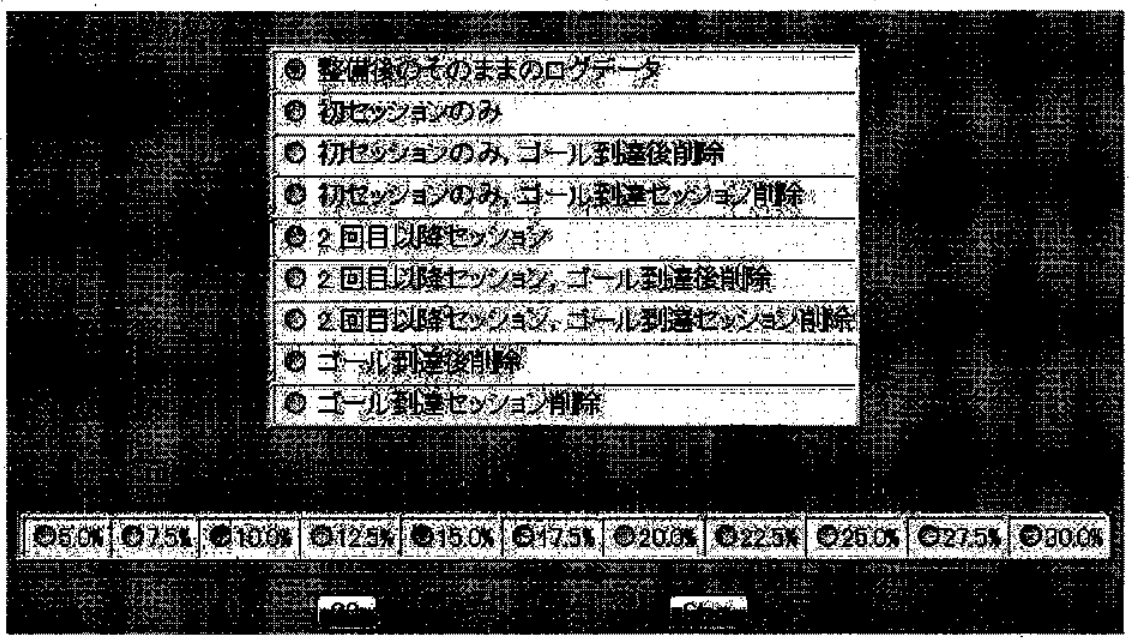
また，離脱率の高低によって各ページをグループ分けするのであるが，その離脱率の高低を決める基準を事前に設定しておかなければならない。その設定は5％から35％まで，2.5％キザミで選ぶことができる。何パーセントの基準を設定するかは一概に決定することができない。基準値を変更して以下の分析を何回か行い，適当な基準値を見つけるしかない。ここでは，何回かの試行錯誤の結果，15％に設定して分析を行った。

以上のような設定は図表4－2に示したように簡単に行うことができる。ここで図表4－2に提示されているログデータのタイプについてそのポイントを解説しておく。

●整備後のそのままのログデータ

通常，アクセスログデータにはWebサイトにアクセスしてきた者の様々なページ履歴が記録されている。その中にはアクセス者のクリックシークエンスによるページ履歴記録の他に，(1)各種検索エンジンが飛ばす情報収集ロボットがアクセスしてきた記録，(2)アクセス者がブラウザのURL指定窓で直接入力したWebページのURL名の記録，(3)アクセス者が同じページで閲覧更新を何度も行った場合の記録，なども残されている。(1)の情報収集ロボットが残したアクセスログデータは今回の分析では無意味データとみなし，削除した。また，(2)のうち，直接入力されたURLに実在しないページ（アクセス者による入力ミスなど）が含まれていてもそれがアクセスログデータに残っているので，実在しないWebページをデータから削除した。(3)については，同じWebページを更新によって閲覧していることが遷移とみなされ，これによって離脱率をい

図表4-2　ページ属性に関する差異分析



たずらに引き下げてしまうので都合が悪い。そこで連続して同一Webページを複数回閲覧した場合はそれを1回のアクセスとみなし，引き続く連続した同一Webページのアクセスログデータを削除した。以上のような整備を行ったものがここでの「整備後のそのままのログデータ」を指す。

●初セッションのみ

ここでセッションとは，アクセス開始となったスタートページからサイト離脱時の最終の閲覧ページまでの一連のページ履歴を指す。初セッションのみのデータは，上記「整備後のそのままのログデータ」の中から，特定のアクセス者が当該Webサイトに初めてアクセスしてきたときの，アクセス開始ページから離脱するまでの一連のページ遷移をデータとして扱ったものを指す。初セッションとはいうものの，入手されたアクセスログデータの記録開始日時以前にアクセスしていたかどうかが不明なため，ここでの初セッションという意味は「入手されたデータの中で」という制限つきのものである。

●ゴール到達後

ここでのゴールとは，商品注文，資料請求，特定情報の提供，アンケート回答など，Webサイト運営者が設定した目標ページを指す。実際のところ今回はデータ入手の制約上，「見積もり」ページにアクセスした段階で便宜的にゴールとみなした。

「ゴール到達後削除」とは，そのアクセス者によるセッションでゴール到達後のページ遷移のアクセスログを削除したものを指す。ゴール到達前の離脱とゴール到達後の離脱ではその意味するところが大きく異なる。前者はWebサイトの運営側に対してサイトの構造や各Webページ

の問題を投げかけるものであり，後者はあまり問題のない離脱であると考えられる。従って，サイト構造やWebページの問題点をより鮮明に浮き上がらせるためにゴール到達後のデータを削除することが考えられる。これによってゴールに到達したセッションには離脱が存在しないことになる。するとこれがまた1つの問題を提起する。ゴールに到達したセッションには離脱が存在しないことになるので，そのようなセッションにおけるページ遷移はすべて離脱率を引き下げることにつながるため都合が悪い。そこで「ゴール到達セッション削除」データを用意した。これは，ゴールに到達したセッションの履歴すべてを削除したものを指す。

● 2回目以降セッション

初セッションではどこにどのような情報がWebページに掲載されているかをアクセス者が知らないため，探索的なページ遷移になりがちである。アクセス者は初セッションでWebサイトの仕組みを学習しているようなものである。一方，2回目以降のセッションはある程度の学習を済ませた後のセッションなので目的のWebページへと比較的スムーズに遷移していくことができる。このように探索的な初セッションと目的志向的な2回目以降のセッションとでは遷移の仕方に大きな違いがありそうである。それらを分析する際にこのデータを利用する。

4－2－2．グループ分けの微調整

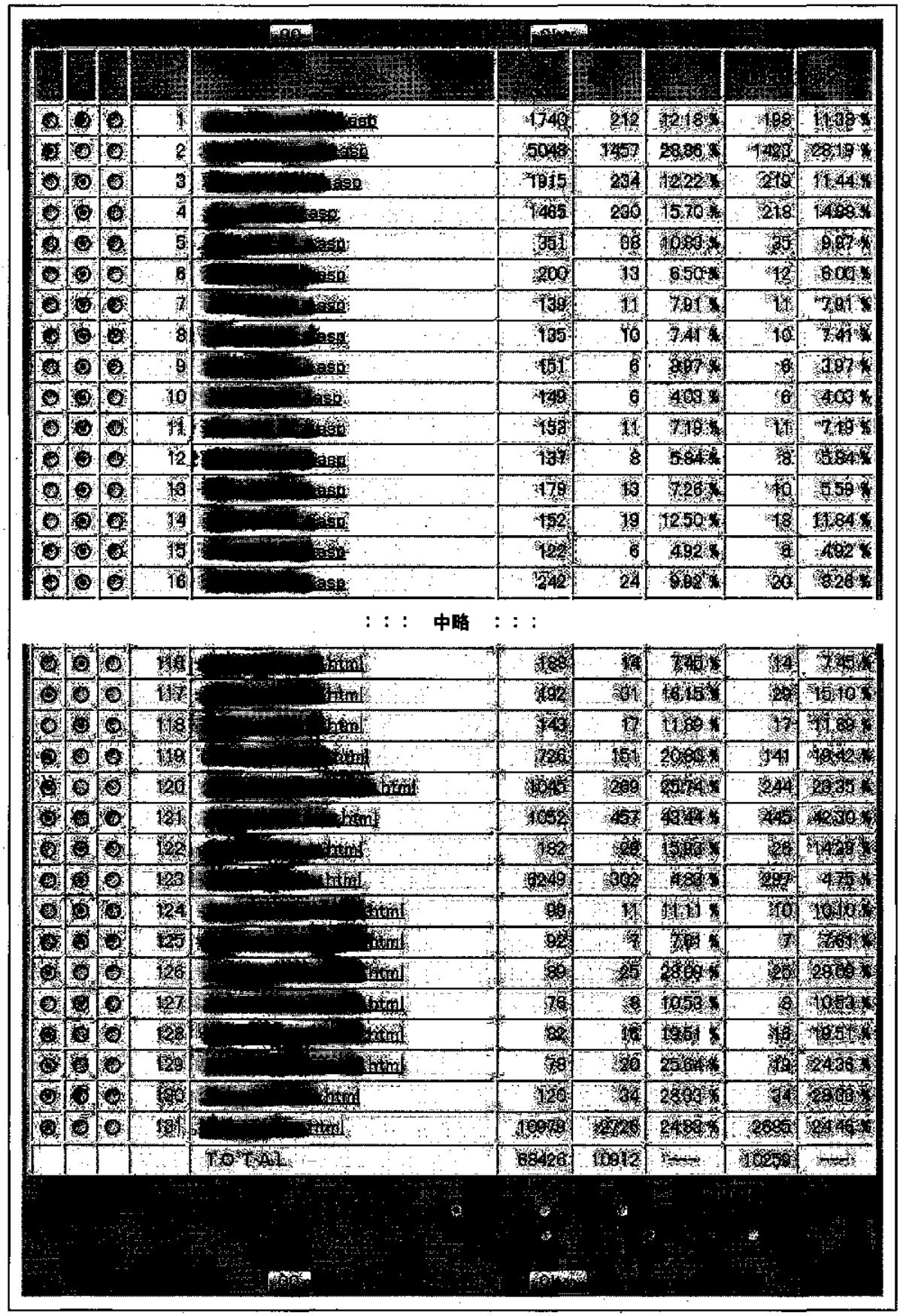
上のような設定で図表4-2に示されているGOボタンをクリックすると，サイトを構成しているWebページ毎の離脱率を表示したものが現れる。例えば，図表4-3がそれである。その図表の左側から順に，「A群」，「B群」，グループ分け対象から外すことを意味する「無視」からなるラジオボタンが表示されているのがわかる。なお，これらは実在したWebショップのWebページ群であるが，分析対象となったWebショップが特定できないようにHTMLファイル名などをワザと判別できないようにしてある。

ここでは「A群」が離脱率の大きなグループを指し，「B群」が離脱率の小さなグループを指している。「無視」はグループ分けの対象から外して分析しないことを意味している。これらのラジオボタンの設定は先ほどの離脱率の高低基準を設定したところから自動的に行われている。

ここでの例は離脱率高低の基準として15%を採用している。従って，あるWebページの離脱率が14.9%であった場合は15%未満ということで離脱率の低い方にグループ分けされることになる。この辺りの微調整を分析者が目で見て判断できるように分析システムが作ってあるので，必要に応じてA群とB群へのグループ分けの微調整を行う。

また，図表4-3に表示されているWebページリストは，アクセスログデータを分析用に調整する際に，*.htmlや*.aspなどの拡張子をもっていてオリジナルのアクセスログデータに記録が残っているWebページファイルから機械的に作成されたものであるため，中にはWebページとして相応しくないものが含まれている可能性がある。例えば，一部のaspファイルはWebブラウザには表示されず，バックグランドで動作する一種のプログラムであるということがある。

図表4-3 Webページ別アクセス統計と離脱率にもとづくグループ分け



			1	[redacted].asp	1740	212	12.18 %	198	11.38 %
			2	[redacted].asp	5048	1457	28.86 %	1423	28.19 %
			3	[redacted].asp	1915	234	12.22 %	219	11.44 %
			4	[redacted].asp	1485	230	15.70 %	218	14.88 %
			5	[redacted].asp	351	38	10.83 %	35	9.97 %
			6	[redacted].asp	200	13	6.50 %	12	6.00 %
			7	[redacted].asp	139	11	7.91 %	11	7.91 %
			8	[redacted].asp	135	10	7.41 %	10	7.41 %
			9	[redacted].asp	151	6	3.97 %	6	3.97 %
			10	[redacted].asp	149	6	4.03 %	6	4.03 %
			11	[redacted].asp	153	11	7.19 %	11	7.19 %
			12	[redacted].asp	137	8	5.84 %	8	5.84 %
			13	[redacted].asp	179	13	7.26 %	10	5.59 %
			14	[redacted].asp	152	19	12.50 %	18	11.84 %
			15	[redacted].asp	122	6	4.92 %	6	4.92 %
			16	[redacted].asp	242	24	9.92 %	20	8.26 %

：：： 中略 ：：：

			116	[redacted].html	188	14	7.45 %	14	7.45 %
			117	[redacted].html	192	31	16.15 %	29	15.10 %
			118	[redacted].html	143	17	11.89 %	17	11.89 %
			119	[redacted].html	726	151	20.80 %	141	19.42 %
			120	[redacted].html	1045	269	25.74 %	244	23.35 %
			121	[redacted].html	1052	457	43.44 %	445	42.30 %
			122	[redacted].html	182	28	15.38 %	26	14.29 %
			123	[redacted].html	6249	302	4.83 %	297	4.75 %
			124	[redacted].html	99	11	11.11 %	10	10.10 %
			125	[redacted].html	92	7	7.61 %	7	7.61 %
			126	[redacted].html	89	25	28.09 %	25	28.09 %
			127	[redacted].html	76	8	10.53 %	8	10.53 %
			128	[redacted].html	82	16	19.51 %	16	19.51 %
			129	[redacted].html	78	20	25.64 %	19	24.36 %
			130	[redacted].html	128	34	28.93 %	34	28.93 %
			131	[redacted].html	19979	4725	24.88 %	2895	24.46 %
				TOTAL	88428	10912	----	10258	----

そのようなaspファイルはWebページとして扱わない方が賢明であると思われるので，これらを分析から取り除いた方がよい。そこで，上記のA群やB群の他に「無視」を表すラジオボタンが用意してある。Webページとして扱えないようなものがリストにある場合は，無視のラジオボタンをクリックして分析から外すことができる。

4－2－3．Webページ別アクセス統計

分析対象となるWebページとそれに関連するアクセス統計としてのアクセス数，離脱数，離脱率が図表4－3に提示してある。ところで，図表4－3には「離脱1」，「離脱率1」，「離脱4」，「離脱率4」，という類似した列が存在する。「離脱1」というのは調整されていない生の離脱フラグにもとづいて集計された離脱であり，それぞれのページで何回の離脱があったかを示している。「離脱率1」はそれぞれのページの総アクセス数で「離脱1」を割って100をかけた離脱率を示している。一方，「離脱4」というのは，ゴールを「見積もり」ページとした上で，そのページに到達した場合は離脱1のフラグを消し，離脱していないものとみなした調整済みの離脱を表している。従って，ゴールページに到達した分だけ離脱数が少なくなる。「離脱率4」は総アクセス数で「離脱4」を割って100をかけた離脱率を表している。なお，「離脱1」や「離脱4」に含まれる数字は上記の意味を区別するための単なる記号で，数字自体に意味はない。

4－2－4．等分散の検定についての設定

2群の平均値の差の検定を実施する前に，2群の分散が等しいかどうかを検定しておく必要がある。2群の分散が等しいという帰無仮説が統計的に棄却できるかどうかを調べる等分散の検定がそれである。等分散の帰無仮説が棄却できなかった場合の2群の平均値の差の検定と等分散の帰無仮説が棄却された場合の2群の平均値の差の検定では，検定に用いる計算式が異なるのである。

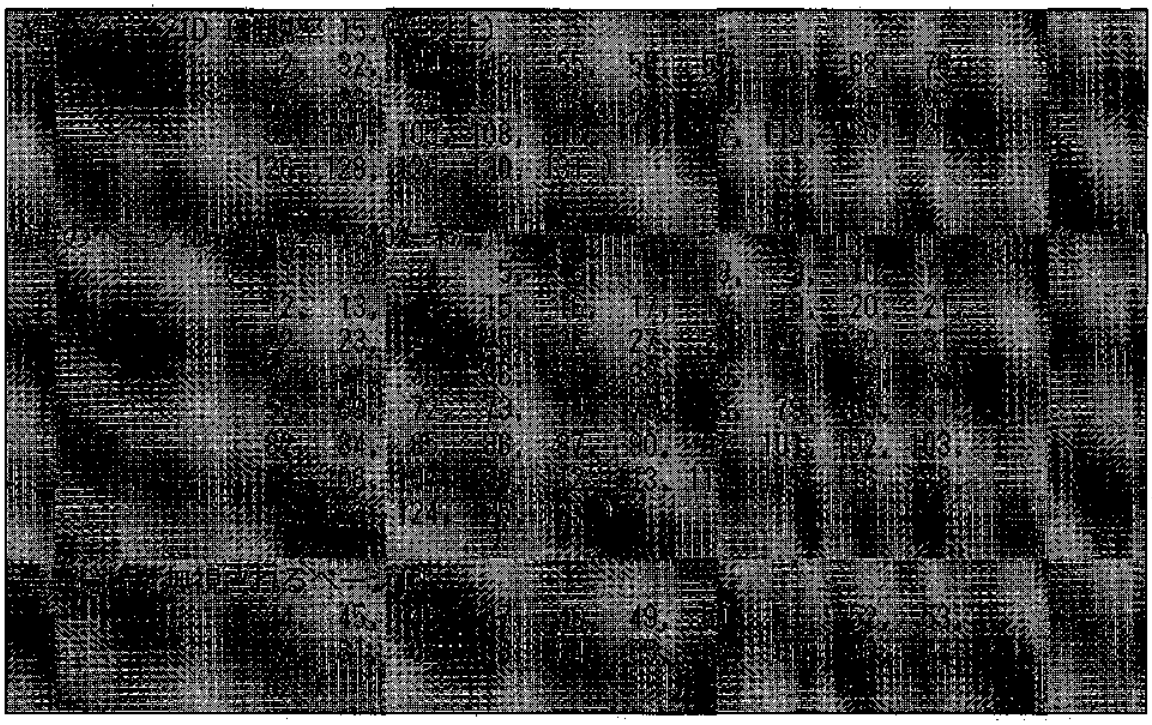
ここでは等分散の検定での有意水準を変更することができる。デフォールトでは5％の有意水準が設定してあるが，必要に応じて1％と10％の有意水準が選べる。有意水準の値を1％と小さく設定すれば，等分散という帰無仮説が棄却されにくくなり，等分散のもとでの2群の平均値の差の検定が行われやすくなる。一方，10％のように有意水準を大きく設定すると，等分散の帰無仮説が棄却されやすくなり，不等分散のもとでの2群の平均値の差の検定が行われやすくなる。通常は5％のままでよいと思われる。

必要に応じてデフォールトの設定を変更し，あるいは特に変更の必要がなければ，図表4－3の上部や下部にあるGOボタンをクリックして2群の平均値の差の検定を実施する。

4－2－5．平均値の差の検定結果

最終的に図表4－4から図表4－10のような結果が表示される。図表4－4には整備後のそのままのログデータを用いた場合のA群（離脱率の高いWebページグループ），B群（離脱率の低

図表4－4　整備後のそのままのログデータによる離脱率高低のグループ分け



いWebページグループ），分析で無視されるページ群，がそれぞれページ識別番号で提示してある。図表4－5から図表4－10までは6種類の2群の平均値の差の検定結果が提示してある。なお，これらは両側検定の結果である。それぞれは図表4－2の初期設定で分析に使用するデータタイプを変えて分析を行ったものである。なお，図表4－5から図表4－10には図表4－4と微妙に異なるページID表がそれぞれ存在したが，ここではその提示を割愛した。

表の右端に「*」印が表示されているが，これらは何%水準で統計的に有意であったかを示している記号である。「*****」は両側1%水準で有意，「****」は両側5%水準で有意，「***」は両側10%水準で有意，「**」は両側15%水準で有意，「*」は両側20%水準で有意ということを表している。なお，一般的な統計的仮説検定では，1%水準と5%水準を用いることが多く，15%や20%水準が用いられることはほとんどない。

●初セッションと2回目以降セッションの比較

まず図表4－5で整備後のそのままのログデータ（全データ）による離脱率高低による差の検定結果を見てみる。文字領域面積，単語種類数といった2つのページ属性が5%水準で統計的に有意となった。すなわち，離脱率15%を基準として，これ以上を離脱率の大きいWebページのグループ，これ未満を離脱率の小さいWebページのグループとした場合，この2つのグループの文字領域面積の平均値は統計的に異なるものと主張でき，同様に，単語種類数の平均値も統計的に異なるものと主張できることになる。文字領域面積も単語種類数もいずれも文字に関係するWebページ属性変数である。従って，Webページに含まれる文字が多い場合にはそのページへのアクセスを最後に当該サイトから離脱してしまう可能性があることをこの分析は示唆している。

図表4－5　整備後のそのままのログデータ（全データ）による離脱率高低の差の検定（両側）

ページ属性名	A群(mean)	B群(mean)	t値	d.f.	p値	差の検定
当該ファイルサイズ（byte）	8,777.63	6,277	1.387	37.3	0.1735	*
リンクHTMLファイル数	10.34	5.21	1.498	36.5	0.1428	**
リンク画像ファイル数	27.63	18.13	1.551	39.3	0.1288	**
リンク画像ファイルサイズ（byte）	41,609.9	31,671.6	1.634	39.5	0.1101	**
その他リンクファイル数	0.03	0	0	0	1	
その他リンクファイルサイズ（byte）	2,044.69	0	0	0	1	
画像領域面積（平方ミリ）	6,866.66	4,890.07	1.562	43.6	0.1256	**
文字領域面積（平方ミリ）	17,601.1	6,656.25	2.404	34.7	0.0217	****
文字領域面積構成比	59.17	57.4	0.36	108	0.7195	
単語種類数	105.91	55.32	2.192	35.4	0.0351	****

*****：1％水準，****：5％水準，***：10％水準，**：15％水準，*：20％水準，で有意。

図表4－6　初セッションデータによる離脱率高低の差の検定（両側）

ページ属性名	A群(mean)	B群(mean)	t値	d.f.	p値	差の検定
当該ファイルサイズ（byte）	8,859.24	6,113.74	1.78	45.4	0.0818	***
リンクHTMLファイル数	9.95	5.08	1.656	44	0.1049	**
リンク画像ファイル数	28.76	16.95	2.148	42.9	0.0374	****
リンク画像ファイルサイズ（byte）	40,109.9	32,531.9	1.451	50.6	0.153	*
その他リンクファイル数	0.02	0	0	0	1	
その他リンクファイルサイズ（byte）	1,703.9	0	0	0	1	
画像領域面積（平方ミリ）	7,065.43	4,702.12	2.007	47.2	0.0505	***
文字領域面積（平方ミリ）	16,372	6,468.29	2.584	41.9	0.0133	****
文字領域面積構成比	59.23	57.4	0.374	69	0.7093	
単語種類数	101.31	54.44	2.406	42.9	0.0205	****

*****：1％水準，****：5％水準，***：10％水準，**：15％水準，*：20％水準，で有意。

図表4－7　2回目以降のセッションデータによる離脱率高低の差の検定（両側）

ページ属性名	A群(mean)	B群(mean)	t値	d.f.	p値	差の検定
当該ファイルサイズ（byte）	9,055.96	6,395.43	1.333	30.8	0.1922	*
リンクHTMLファイル数	11.04	5.41	1.357	28.5	0.1855	*
リンク画像ファイル数	29.68	18.24	1.561	29.7	0.1292	**
リンク画像ファイルサイズ（byte）	39,165.4	33,354.7	0.83	31.2	0.4128	
その他リンクファイル数	0	0.01	0	0	1	
その他リンクファイルサイズ（byte）	0	872.73	0	0	1	
画像領域面積（平方ミリ）	6,942.39	5,032.94	1.279	32	0.2101	
文字領域面積（平方ミリ）	17,589.6	7,594.5	1.8	27.9	0.0827	***
文字領域面積構成比	58.89	57.65	0.238	108	0.8121	
単語種類数	106.64	59.39	1.684	28.2	0.1032	**

*****：1％水準，****：5％水準，***：10％水準，**：15％水準，*：20％水準，で有意。

以上までは有効なWebアクセスログデータをすべて用い，そこに現れるWebページを分析対象とした。しかしながら，分析はこれに限らない。例えば，アクセス者が初めてそのサイトに訪れた場合のページ遷移パターンと2度目や3度目にそのサイトに訪れた場合のページ遷移パターンは異なることが予想される。前者はあちこちページを渡り歩きながら目的とするページを探すようなアクセスをするであろうし，後者は目的のページまで一直線にアクセスをするであろう。従って，2回目以降のセッションでのアクセスログを削除し，初回セッションのみのアクセスログを作り，そこに現れるWebページでの離脱率で2群のグループ分けをしてページ属性変数についての平均値の差の検定をすることが考えられる。

続いて図表4-6を見てみよう。これはアクセス者による初セッション時のアクセスログデータのみを分析対象としたものである。差の検定結果によると，5％水準で統計的に有意となった変数はリンク画像ファイル数，文字領域面積，単語種類数であった。全データの分析では見えてこなかったリンク画像ファイル数の違いが明らかとなった。これは当該Webページに貼り付けられている画像ファイルの数を指し，離脱率が高いWebページでは貼り付けられている画像ファイル数が平均的に多いと統計的に主張できる。このことは，初セッション時において，画像が多かったり文字が多かったりなどしてアクセスに時間がかかり，アクセス者にストレスを課すような場合にはアクセス者のサイト離脱を発生させることを示唆するものと理解することができよう。

一方，図表4-7を見てみる。これはアクセス者による2回目以降のセッション時のアクセスログデータのみを分析対象としたものである。なお，初セッションデータと2回目以降のセッションデータを結合すると全データとしての整備後のそのままのログデータとなる。差の検定結果をみると，5％水準で統計的に有意となった変数はひとつも存在しなかった。ここで取り上げた10個の変数で離脱率の高低を5％水準で統計的に有意に識別できなかったということは，画像の多さや文字の多さは2回目以降のセッションでは離脱につながらないことを示唆している。2回以上のセッション行うアクセス者はある程度強い目的意識をもっていると考えられるから，その場合には多少のストレスがあっても離脱にはつながらないことを示唆しているように理解できる。

●ゴール到達後削除データによる初セッションと2回目以降セッションの比較

直上で行った分析をもう少し突っ込んで検討してみたい。Webサイトの運営側にとっては，アクセス者が何らかのゴールに到達したか否かは非常に重要な問題である。ここでのゴールとはWebサイト運営側が設定したゴールであり，例えば商品の購入注文であったり，資料請求であったり，特定の情報提供であったりと多様であり得る。そのようなゴールに到達したアクセス者がその後に離脱してもそれは必然的な離脱であり，大きな問題とはみなされない。ところが，ゴールに到達していないアクセス者が途中で離脱してしまうことはWebサイト運営側にとっては何らかの問題を提起しているものと考えられる。Webページのコンテンツが貧弱であったの

図表4-8　ゴール到達セッション削除データによる離脱率高低の差の検定（両側）

ページ属性名	A群(mean)	B群(mean)	t値	d.f.	p値	差の検定
当該ファイルサイズ（byte）	8,690.91	5,993.82	1.817	48.4	0.0754	***
リンクHTMLファイル数	9.86	4.83	1.783	46.2	0.0812	***
リンク画像ファイル数	28.34	16.36	2.272	45.6	0.0279	****
リンク画像ファイルサイズ（byte）	39,507.6	31,717.9	1.528	54.8	0.1323	**
その他リンクファイル数	0.02	0	0	0	1	
その他リンクファイルサイズ（byte）	1,626.45	0	0	0	1	
画像領域面積（平方ミリ）	6,942.95	4,569.67	2.075	49.4	0.0432	****
文字領域面積（平方ミリ）	16,050	6,197.88	2.686	44	0.0102	****
文字領域面積構成比	59.12	57.19	0.416	108	0.6782	
単語種類数	99.82	52.48	2.534	45.4	0.0148	****

*****：1％水準，****：5％水準，***：10％水準，**：15％水準，*：20％水準，で有意。

図表4-9　初セッション，ゴール到達セッション削除データによる離脱率高低の差の検定（両側）

ページ属性名	A群(mean)	B群(mean)	t値	d.f.	p値	差の検定
当該ファイルサイズ（byte）	8,551.91	6,164.63	1.67	51	0.101	**
リンクHTMLファイル数	9.37	5.19	1.537	49.3	0.1308	**
リンク画像ファイル数	27.78	16.92	2.148	47.8	0.0368	****
リンク画像ファイルサイズ（byte）	39,397.8	32,571.3	1.4	58.6	0.1667	*
その他リンクファイル数	0.02	0	0	0	1	
その他リンクファイルサイズ（byte）	1,555.74	0	0	0	1	
画像領域面積（平方ミリ）	6,822.3	4,730.03	1.905	52.9	0.0622	***
文字領域面積（平方ミリ）	15,465.5	6,501.87	2.539	46.2	0.0145	****
文字領域面積構成比	59.42	57.14	0.487	79.5	0.6275	
単語種類数	97.15	54.5	2.375	47.8	0.0216	****

*****：1％水準，****：5％水準，***：10％水準，**：15％水準，*：20％水準，で有意。

図表4-10　2回目以降，ゴール到達セッション削除データによる離脱率高低の差の検定（両側）

ページ属性名	A群(mean)	B群(mean)	t値	d.f.	p値	差の検定
当該ファイルサイズ（byte）	8,434.4	6,198.7	1.481	48.5	0.1451	**
リンクHTMLファイル数	9.42	5.19	1.483	47.5	0.1446	**
リンク画像ファイル数	27.51	17.07	1.945	46.1	0.0579	***
リンク画像ファイルサイズ（byte）	38,100.4	32,737.3	1.036	54.7	0.3047	
その他リンクファイル数	0.02	0	0	0	1	
その他リンクファイルサイズ（byte）	1,664.28	0	0	0	1	
画像領域面積（平方ミリ）	6,601.56	4,824.19	1.52	49.5	0.1348	**
文字領域面積（平方ミリ）	15,042.9	6,991.27	2.138	44.6	0.038	****
文字領域面積構成比	58.43	57.66	0.164	108	0.8699	
単語種類数	92.95	57.6	1.849	46.5	0.0709	***

*****　1％水準，****：5％水準，***：10％水準，**：15％水準，*：20％水準，で有意。

か，素材の配置がまずかったのか，リンクの提示位置がわかりにくかったのか，素材の情報量が多すぎてアクセス者に多大なストレスを課してしまったのか。このように考えると，ゴールに到達したアクセス者の離脱はWebサイトの問題を示唆する離脱とはみなされないので，ゴールに到達したセッションのアクセスログデータを削除することで分析がより明確になると思われる。そこで，ゴール到達セッションのデータを削除した上で，先と同様に，初セッションと2回目以降セッションの比較分析を行った。

図表4-8は初セッションと2回目以降の区別をしないもので，単純にゴール到達セッションのアクセスログを削除したデータで差の検定を行ったものである。これによると，リンク画像ファイル数，画像領域面積，文字領域面積，単語種類数の4つの変数が5％水準で統計的に有意であった。

図表4-9は初セッションで，そのうちゴール到達したセッションのアクセスログを削除したデータで差の検定を行ったものである。これによると，リンク画像ファイル数，文字領域面積，単語種類数の3変数が5％水準で統計的に有意であった。離脱率が高いWebページではこれらの変数の平均値が離脱率の低いWebページの平均値と比べて統計的に有意な差があると把握された。これは情報量の多いWebページはそのアクセス・表示に時間がかかり，これによりアクセス者にストレスを課し，離脱を促していることが示唆される。あるいは，情報量の多さが，アクセス者の処理能力を超え，これによるストレスで離脱を促している可能性もある。

図表4-10は2回目以降セッションで，このうちゴールに到達したセッションのアクセスログを削除したデータで差の検定を行ったものである。これによると5％水準で統計的に有意となった変数は文字領域面積の1つだけであった。離脱率の高いWebページと離脱率の低いWebページでは，文字領域面積の平均値に有意な差のあることが統計的に支持された。

5．まとめと今後の課題

5-1．ま と め

ブロードバンド時代の到来以前の分析ではあるものの，離脱率の高いWebページと離脱率の低いWebページとでは，それらが提示する情報の量（リンク画像ファイル，文字領域面積，単語種類数）に統計的な差異のあることが把握された。さらに，統計的な差の検定を経ていないものの，アクセス者が初セッションであるか2回目以降のセッションであるかによっても差異のあることが把握された。これらを踏まえると，Webサイトの運営において，アクセス者が初セッションであるか2回目以降のセッションであるかを識別し，それに応じたWebページを提示する必要があるといえる。特に，初セッションのアクセス者には画像や文字を軽めにしてアクセスにかかわるストレスを軽減し，これによって離脱を減らして1ページでも多くのサイト内遷移をしてもらう工夫が必要といえる。

5－2．今後の課題

今回の研究では，当該ファイルサイズ（byte），リンクhtmlファイル数リンク画像ファイル数，リンク画像ファイルサイズ（byte），その他リンクファイル数，その他リンクファイルサイズ（byte），画像領域面積（平方ミリ），文字領域面積（平方ミリ），文字領域面積構成比，単語種類数，をページ属性として取り上げた。それぞれの測定における問題点や課題などはその説明をしている節や項で指摘しておいた。

Webサイトの離脱やWebページの遷移に影響を与えるような要因としては，今回の研究で取り上げた変数の他にも様々なものを考えることができる。

5－2－1．背景色や文字列部分の色

現在のWebページ属性自動抽出プログラムでは，色属性を抽出していない。しかしながら，その技術はページに貼り付けられた画像ファイル名などを自動抽出することと同じなので，今後は色属性を自動抽出することが課題として挙げられる。

色はRGBを数値で表現することが可能である。Redについて256色，Greenで256色，Blueで256色を指定することができるので，16,777,216色を数字で表現することができる。また，Black，White，Midnightblueなど，キーワードで色を指定することもできる。色キーワードからRGBの16進数表記への変換テーブルを用意しておけば，htmlファイルから色属性を自動的に抽出することは容易である。

しかしながら，色属性については取り扱いが困難な側面がいくつか存在する。

- 1ページには多数の色が使われていることが一般的であり，その多数の色を当該Webページの属性としてどのような形でデータとして表現するか。
- 背景色と文字色との組み合せや，隣接する文字の文字色の組み合せによっては，文字が際立って見えたり，霞んで見えたりするため，背景色と文字色との組み合わせをWebページ属性データとしてどのように表現するか。
- 背景色と文字色との組み合せや，隣接する文字の文字色の組み合わせによっては様々な印象をアクセス者に与えそうである。落ち着いた雰囲気，楽しい雰囲気，危険な雰囲気など，色の組み合せによる心理的な影響をどのようにWebページ属性データとして表現するか。

5－2－2．背景画像

背景画像はページの印象にかなり大きな影響を与えそうである。どのような背景画像がWebページに貼り付けてあるのかをデータ化することは意義のあることと言える。

また，文字色との組み合せによっては背景画像が文字の判読を容易にさせたり困難にさせたりする。その組み合わせをWebページ属性としてどのようにデータ化するかも課題である。

5－2－3．文字列や図の配置

配置については，ブラウザをどのような大きさで起動しているかによってずいぶんと違った印象を与える。例えば，ブラウザを小さく開いている場合，外見上は図が中央部分に配置されているように見えても，実は大きなページの左上部分に過ぎないということがある。また，表示がスクロールなしの１ページに収まっているかどうかもブラウザの大きさに依存して異なってくる。

スクロールが必要な場合はアクセス者にそれだけの負担を強いることになり，これがページ遷移や離脱に大きな影響を与えそうである。しかしながら，アクセス者がどの程度の大きさでブラウザを開いているかはサーバー側のログデータからは知ることができないので，文字列や図の配置をデータ化することはほとんど不可能といえそうである。そこで，モニター解像度やブラウザのサイズを無視し，htmlに記載されている配置情報（left，center，right）を読み取って大まかな配置をデータ化することが考えられる。しかしながら，上下の配置はページをスクロールすることによってその都度変化するのでデータ化することは困難である。とは言うものの，最上段と最下段は識別することができると思われる。

5－2－4．アクセス者属性の考慮

以上までの分析は離脱の原因を専らWebページの属性に求めていた。しかしながら，アクセス者の特性も離脱には大きく関係しているように思われる。例えば，目的意識の明確なアクセス者は多少の情報提供過多があってストレスを感じても，目的を達成するために簡単には離脱をしないであろう。一方，通りすがりのアクセス者は興味本位のちょっとしたアクセスなのでわずかなストレスによっても離脱してしまうということが考えられる。今回の研究では便宜的に初セッションと２回目以降セッションという形でデータを分割することでアクセス者属性を間接的に取り扱った。しかしながら今後の研究ではアクセス者の属性を積極的に考慮に入れ，それをどのように変数として扱うか，その上で離脱と遷移に関する分析を行いたい。

5－2－5．離脱・遷移に対する分析変数の非線形性の考慮

今回の研究では個々の分析変数が単独で離脱や遷移と関係しているかどうかを調べてきた。しかしながら，個々の分析変数が単独ではなく，相互に複雑に絡み合いながら離脱や遷移に関係していると考えるほうが現実的である。それらの複雑な絡み合いを分析するために，今後は決定木を分析手法として採用し，分析変数の離脱・遷移に対する非線形的な関係を研究したい。

通常，決定木による分析では統計的な検定作業が伴わないため，分析結果の妥当性を主張することが困難である。そこでブートストラップ法を適用して分析結果の妥当性を検討することを試みたい。また，どのような分析変数が離脱や遷移に影響を与えるかどうかを調べるため，変数探索アルゴリズムを採用したい。

謝 辞

本論文はインターネット・マーケティング・サイエンス研究所の資金援助とデータ提供を受けて2001年度に行われた共同研究を基礎とし，筆者が直接担当した部分について大幅な追加・修正を施したものである。お世話になった当研究所，並びに共同研究チームのメンバーに感謝の意を表明したい。なお，本論文に含まれ得る誤りの責任が著者にあることは言うまでもない。

また，本研究では多数のオープンソースソフトウェアを使用させていただいた。Linux，Perl およびPerl 拡張モジュール，Lynx，kakasi，Emacs，Apache といったソフトウェアの開発関係者に感謝の意を表したい。

参考文献

Chang, Healey, McHugh, Wang 著，武田善行・梅村恭司・藤井敦 訳，『Web マイニング』，共立出版，2004年。

Cooley, Robert, Bamshad Mobasher, Jaideep Srivastava, *Data Preparation for Mining World Wide Web Browsing Patterns*, Journal of Knowledge and Information Systems, 1999.

Srivastava, Jaideep, Robert Cooley, Mukund Deshpande, Pang-Ning Tan, *Web Mining: Pattern Discovery from World Wide Web Transactions*, SIGKDD explorations January 2000, Volume 1, Issue 2, 2000.

CYBERmetrics（http://www.cindoc.csic.es/cybermetrics/）

金子武久，「インターネットCRM におけるデータマイニングの位置づけ」，『創価経営論集』，第29巻第1・2合併号，2004年。

守口剛，阿部誠，金子武久，井上達紀，佐藤栄作 共著，『クリックシークエンスに影響するページ属性の把握と予測モデルの構築』，インターネット・マーケティング・サイエンス研究所報告書，2002年。

Tom Christiansen，Nathan Torkington 著，田和勝 訳，『Perl クックブック』，オライリー・ジャパン，2001年。

馬場 肇，『Namazu システムの構築と活用―日本語全文検索徹底ガイド』，ソフトバンクパブリッシング，2001年。

CPAN（Comprehensive Perl Archive Network，http://www.perl.com/CPAN/）