

ワープロ利用者の思考時間に関する 統計的モデルの精密化

木村 泉・粕川正充
(東京工業大学理学部)

梗概

日本語ワープロソフト「松」(旧版)の利用者が、かな漢字変換の結果を受け入れるか否かを決断するために費やした時間(変換判断時間)の統計的分布を解析する。これは先に報告した解析を、次の2点において精密化したものである。(a) フロッピー・ディスク装置から辞書項目が読み出されてから実際に変換結果が画面に表示されるまでにかかる計算時間(画面更新時間)を実測し、解析データに加味した。(2) 以前には分布を三つの成分から成るものと解釈したが、ほかの実験からの傍証に基づいて、そのうちの二つを一つにまとめた。想定されるメカニズムは以前のものより自然さを増したといえる。

Refining a Statistical Model of the User Think Time on a Japanese Word Processor

Izumi KIMURA, and Masaatsu KASUKAWA

Department of Information Science, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152 Japan

Abstract

A statistical distribution, presented previously, of the decision time a user of the Japanese word-processing software Matsu (old version) spends in deciding to accept or reject the result of phonetic-to-ideographic conversion is reanalyzed. The points are: (a) the time, which elapses after a dictionary entry is accessed and before the result of conversion is actually displayed, is taken account of; and (b) two of the three statistical components identified previously are combined into one, giving a more plausible interpretation.

0. 狙い

[1]では、ワープロソフト「松」を使って実作業をしている利用者が、かな漢字変換の結果を受け入れるか否かを判断するために費やした時間（以下変換判断時間と記す）の頻度分布を解析し、その解析結果をもとに利用者の思考過程に関する一つの模型を与えた。その模型はもとの分布の形状のみならず、他の利用者についての同種の分布の形状も、（模型では予測していなかったある現象——先打ち現象——が生じることもあるという点を別とすれば）よく説明していた[2]。しかしその模型による解釈には、若干気になる点が残っていた。すなわち、

- a. [1]では上記変換判断時間の測定に当たって、ワープロソフトが画面更新に費やすCPU時間（以下画面更新時間と記す）を無視し、フロッピー・ディスクからの辞書アクセスのための割り込みが完了すれば画面は即時に更新されるものと仮定した。その仮定には厳密に言えば疑問の余地があった。実際、複雑な活用語尾を伴う文節の変換では、フロッピー・ディスクの動作音と実際の画面更新の間にずれが、利用者の目にもわずかながら感知される場合があった。
- b. [1]では問題の分布を三つの成分（それぞれ山1、山2、山3と仮称）から成るものとしたが、このうち山3は低い山で、形状から特徴を見定めにくく、また認知心理学的に見ても実体がいまいち明らかでない、というきらいがあった。

本文では、これらの問題点を解決するためにおこなった追加実験、およびそれに基づく[1]のデータの再解析について述べる。具体的には、まず問題点aに対処するため、[1]で使った打鍵タイミングデータをもとに計算機との対話を再現（プレーバック）することによって、各変換判断時間についてそれに伴う「松」の画面更新時間を実測した。そしてその結果を加味した頻度分布について、[1]における解析を繰り返した。ただしその際問題点bに即応して、分布の解釈を一部変更した。すなわち、従来山2および山3（二つの正規分布）にわけていたものを、単一の山（仮に山2Aと名づける）に当てはめることとした。山2および山3を統合することにしたのは、[3]の実験の結果から得たヒントに基づいてのことである。

以下、第1節では[1]の結果を手みじかに要約し、基本的な用語、記法を導入する。第2節では画面更新時間の計測方法、および計測結果について説明する。第3節では[3]の実験からの「ヒント」について具体的に述べ、第4節では[1]の分布に第2節の結果を加味したものを、上記「ヒント」を考慮して解析しなおした結果について説明する。第5節では結果を評価し、今後の課題について述べる。

1. [1]の結果の要約

1.1 分布と当てはめ結果

図1(a)に、[1]で解析した頻度分布のヒストグラムを再掲する。これはある1篇のエッセイの英文和訳作業、「松」（旧版）を使ってした際に生じた、前記の意味での変換判断時間の分布である。ただし辞書検索のためのディスクアクセスが終われば、変換結果はスクリーンに表示されるものと仮定している。すなわち、画面更新時間を0と見なしている。

作業はまず原文を読み、疑問のある部分について辞典を引き、次に原文を見ながら直接訳文を打ち込むという手順によった。訳文を打ち込んだあとは全体を見なおし、スタイル（1行の文字数など）を調整し、最終原稿をプリンタで打ち出した。電話がかかってきたなどによる合計数分のロスタイム、およびワープロソフトの開始終了時間を含むそこまでの総作業時間は約6時間22分である。

ただし冒頭の約50分についての記録は、打鍵データ採取ソフトウェアの操作ミスによって失われた。記録が失われた部分のほとんど（約38分）は原文の読みと辞書引きに費やされ、打鍵データが実際に失われたのは約8分、打ち上がり正味273字（改行符号等を含み、空白行は1字と数える）のみであった。さいわいその273文字は問題が発生したときすべて画面上に残っていたので、ほかの人に画面を見ながら打ちなおしてもらった。

図1の分布は、その部分を除く全作業時間中に生じた、すべての文節変換に関する初回の変換判断時間を含んでいる。ここで初回の変換判断時間とは、かな文字文に対してはじめてなされた変換キーの打鍵に対する変換判断時間という。変換キーを1回叩いたあと、得られた変換候補に満足せず、他の候補を求めるためになされた2回目以降の変換キーの打鍵は考えなかった。また連文節変換や単漢字変

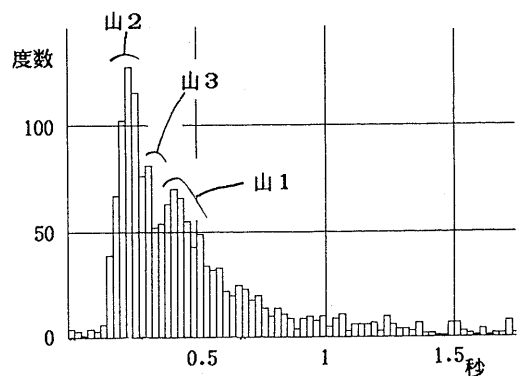


図1 [1]における(a)頻度分布とその成分

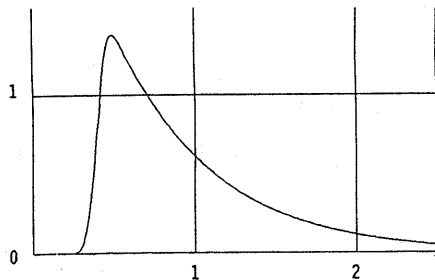


図2 関数 Ψ ($\mu = 0.4$ 、 $\sigma = 0.05$ 、 $\lambda = 0.6$)

換は対象から除外し、文節変換のみ収集した。かな文字文の入力はローマ字モードによった。

くわしくいうと、対象としたのは下記の3から4までの時間であり、3の開始は2にいうディスクアクセスの最後の1回が起こった時点と仮定した。

1. 利用者が文節をローマ字打ちで打鍵すると、対応するかな表記が画面にあらわれる。
2. 利用者は文節の切れ目で変換キー（XFERキーと呼ばれる）を叩く。するとかな漢字変換操作が開始される。一般にはその際、フロッピーディスクのアクセスが何回か起こる。
3. やがて変換結果が計算機のスクリーン上に提示される。利用者はそれを見て変換結果を受け入れるべきか否かを判断する。
4. 利用者はその判断を、文章の先を打ちはじめること（結果を受け入れる場合）またはもう一度変換キーを叩くこと（別の交換候補を要求する場合）によって計算機に伝える。ただしここで先を打ちはじめという中には、確定キーを叩いて一応変換結果を確定させ、次にワープロソフトの編集機能を使って文章の別の場所の修正に掛かる、というようなケースも含まれる。

[1]では図1の分布を、図に示すように三つの山から成っているものと解釈した。このうち山1はさまざまな時定数をもつ指数分布と正規分布が、標本同志が足し合わされているという意味で組み合わせさせたもの、山2および山3は正規分布である。

1.2 Ψ 分布

上で言及した「指数分布と正規分布が、標本同志が足し合わされているという意味で組み合わせさせた分布」は、[1]のみならず本文でも重要な役割を演ずるので、ここでその性質を記しておくことにする。指数分布の時定数を λ 、正規分布の平均を μ 、標準偏差を σ とすると、この分布（以下便宜上 Ψ 分布と呼称する）の確率密度関数は次のと

おりである。

$$\Psi = \frac{1}{\lambda} e^{-\frac{1}{2} \left(\frac{\sigma}{\lambda}\right)^2} \cdot e^{-\frac{x - (\mu + \sigma^2/\lambda)}{\lambda}} \cdot \Phi \left[\frac{x - (\mu + \sigma^2/\lambda)}{\sigma} \right] \quad (1)$$

ただし、

$$\Phi [x] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-y^2/2} dy \quad (2)$$

関数 Ψ の形状は図2に示すとおりであって、一つのピークをもち、長い尾を引く。その尾は指数関数的変化を示す。

[1]における山1は、さまざまな λ と共通の μ 、 σ をもつ Ψ 分布の重ね合わせ（線形結合）になっている。[1]ではこの分布の指数関数的な尾の形状に着目することによって、4種の λ を分離することができた。それらの λ は大きい順にそれぞれ19.89秒、3.55秒、0.643秒、0.214秒であった。

1.3 思考時間模型

この当てはめ結果から[1]では、図1(a)の分布の成因を次のように解釈した。

- a. 利用者は画面に表示された変換結果を見てその当否を判断し、判断の結果いかに応じて適当なキーを打鍵する。以上に要する時間は、基本的にはあるばらつきの小さい正規分布にしたがう。
- b. ただし(a)のプロセスの途中の、当否の判断の段階までできたところで、ある確率で確認等のための「間」が生じる。そうなった場合の所要時間は、(a)とは別の、あるばらつきの小さい正規分布にしたがう。
- c. また(b)に代わって、(上記の、当否の判断の段階まで進んだところで)より深い思考過程が「発火」することがある。発火する思考過程には種々の深さのもの（用語字に関するものか、文章全体の構想に係る戦略的なものか、など）があり、それぞれある確率をもって発火する。
- d. 一旦発火すると思考過程は、思考の深さに応じたある時定数をもつ指数分布にしたがって継続する。その時定数は、深い思考に関わるものほど長い。
- e. 思考が終了すれば何らかの打鍵操作がおこなわれ、プロセスが終了するが、そのとき打鍵されるものは一般に変換開始以前に打鍵しようと思っていたものとは違うので、さらにそこで体勢を立てなおすためのある時間が掛かる（その時間もばらつきの小さい正規分布にしたがう）。

[1]では、これを図3に示すような図式にまとめた。

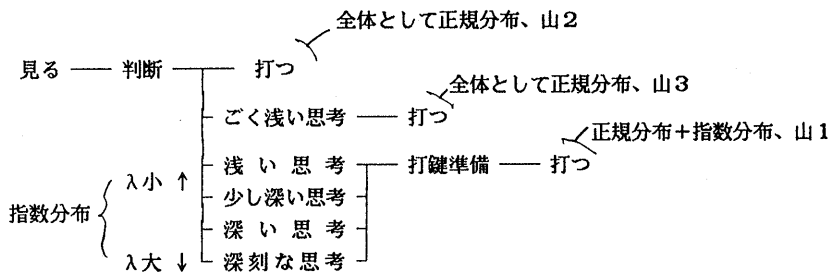


図3 統計的模型(旧版、[1]より再掲)

2 画面更新時間の測定

たまたま[1]の実験で取り上げたワープロソフト「松」(旧版)は、反応が速いことで著名である。実際、「松」の発売以来長い年月を経て種々の意味ですっとすぐれたワープロソフトが多数開発された今日なお、その一点ゆえに「松」を使っている利用者が実在する。

とはいうものの、「松」の画面更新時間が利用者の変換判断時間に比べて無視してよいほど小さいかどうかは具体的検討を要する。実際[3]に記したように、比較的少数のサンプルについてこれを計測してみたところ、その平均値は36msecとなった。これは典型的な変換判断時間の数%に当たる。

特に問題なのは、この画面更新時間がときにきわめて大きくなるらしいことである。実際にそうであることは、あとで示す。

そこで今回、図4に示すように2台のパーソナルコンピュータを用意し、図1のもとになった打鍵タイミング記録をもとに、もとの著作・編集作業の中で起こった事象を第1のパーソナルコンピュータ(たまたまNEC PC-9801を使用)にブレーバックさせ、その結果得られた信号を第2のパーソナルコンピュータ(たまたまNEC PC-9801E)の鍵盤の接続口から送り込むことによってもとの実験の際の「松」の働きを再現し、その際の画面更新時間を計測することを試みた。ほとんどすべてのことはソフトウェア的手段で処理したが、第1のコンピュータを第2のそののキーボードセットに見せかけるための簡単なインタフェースだけは特別に作成した。このインタフェースはキー入力コードを送り出す機能と、ブレーバック側コンピュータからのリセット信号を受け取って然るべき処置をする機能をもっていなければならない。実際、後者は「松」をブレーバックしようと思えば必須の機能である。

周知のように「松」はNEC PC-9800シリーズ上で動作する。PC-9800シリーズでは、画面に書き出されている文字データがどんなものであるかは主記憶上のテキストRAM領域を

見ればわかるようになっていた。そこで画面更新時間の測定は、一定時間ごとにテキストRAMの内容の然るべき部分を走査し、変化があれば(打鍵等と同様にして)時刻を記録するという方法でおこなうことにした。くわしくはカーソル位置の前後各80バイト分のみ走査した。画面が次々に書き換えられてゆくプロセスを全部記録するとデータが膨大になりすぎるので、鍵盤のメーク、ブレイク、ディスクからの割り込みなどのような他の事象で前後を区切られた各区間において、最初と最後の画面更新だけを記録することにした。

ブレーバックのもととなった打鍵タイミングデータの採取はインタバルタイマにより、10msecごとに割り込みを起こすという方法でおこなったが、この10msecという周期は画面の更新周期となじみが悪い。そこで今回は、画面リフレッシュ動作の開始信号による割り込み(17.72msecごと)を利用した。すなわち、画面のリフレッシュ動作の開始と同時にテキストRAMの内容のチェックを開始する。それは7.72msec以内には確実に終了する程度の作業である。そこでテキストRAMの内容チェックと平行してインタバルタイマを走らせ、内容チェック作業が実質的に終了したあとはタイマの内容を絶えず監視しながら7.72msec経過する

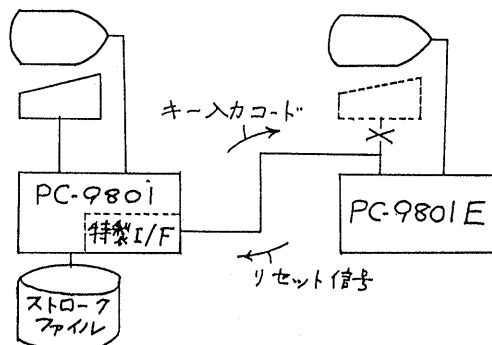


図4 ブレーバックの機構

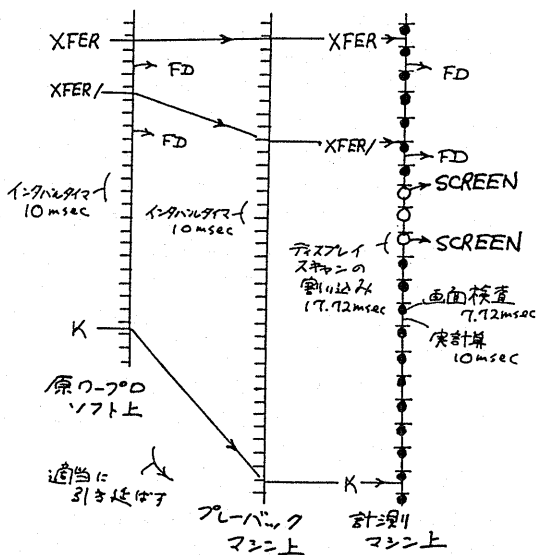


図5 画面更新速度の測定方法

のを待つ。7.72msec経過後は割り込みから戻ってワープロソフトの中を走る。画面リフレッシュ動作の割り込みが次に起こるのは、それからちょうど10msecあとということになる。これはもとのプレーバック用データが10msecきざみで採取されているのともなじみがいい。すなわち、この実験で画面更新にn回の割り込み周期が必要であったとすれば、もとの実験で対応する画面更新に要した時間はn回の

割り込み（10msec周期での）に要した時間であったと見ることができる。

問題の7.72msecの間は、実計算はおこなわれないが、鍵盤（いまの場合正体はプレーバック機構）やフロッピー・ディスクからの割り込みは受けつけられる。それらの割り込みが同じ7.72msecの間に2回以上起こると、割り込みの結果が失われる恐れもあるが、フロッピー・ディスクからの割り込みは1回起こったあとは、実計算の結果として新しいコマンドがCPUからディスクのコントローラに送られない限りもう起こらないから心配はない。問題は鍵盤であるが、これについてはプレーバック側からの送り出し速度を落とすという方法で対処する。以上の方法の概要を、図5にタイムチャートの形で示す。図は、送り出し速度は1/2に落とすように描かれているが、これは実情どおりではない。送り出し速度は実際にテストして決めた。

なお画面更新のタイミングはハードウェア的にとらえる、という方針も可能である。たとえばバス上のアクセス要求を見張っていて、それがテキストRAMへの書き込みアクセスであるときは割り込みを起こす、というような機構を追加することが考えられる。しかしここでは、これまでの行きがかりもあって上記のようなソフトウェア的方法を採用した。

図6(a)はプレーバックのもととなった打鍵データの一部、図6(b)は対応する画面更新時間測定結果を示す。ここに示すのは、まず「きよだいな」と打ち、変換キーを叩き、首

53	up	57.699	1.145	23	up	31.170	1.300
54	k	86.872	29.174	24	k	42.580	11.410
55	y	87.176	0.304	25	SCREEN	42.620	0.040
56	o	87.268	0.091	26	SCREEN	42.650	0.030
57	d	87.450	0.182	27	y	42.960	0.310
58	a	87.551	0.101	28	SCREEN	42.970	0.010
59	i	87.754	0.203	29	o	43.080	0.110
60	n	88.079	0.324	30	SCREEN	43.100	0.020
61	a	88.261	0.182	31	d	43.360	0.260
62	xfer	88.413	0.152	32	SCREEN	43.370	0.010
63	FD	88.474	0.061	33	a	43.500	0.130
64	FD	88.687	0.213	34	SCREEN	43.510	0.010
65	k	89.102	0.416	35	i	43.800	0.290
66	e	89.204	0.101	36	SCREEN	43.810	0.010
67	i	89.325	0.122	37	n	44.200	0.390
68	s	89.832	0.507	38	SCREEN	44.220	0.020
				39	a	44.450	0.230
				40	SCREEN	44.460	0.010
				41	xfer	44.640	0.180
				42	FD	44.680	0.040
				43	FD	44.810	0.130
				44	SCREEN	44.880	0.070
				45	k	45.530	0.650
				46	SCREEN	45.600	0.070
				47	e	45.690	0.090
				48	SCREEN	45.710	0.020
				49	i	45.870	0.160
				50	SCREEN	45.890	0.020
				51	s	46.480	0.590

図6 打鍵タイミングデータの例 (a)原型、
(b)画面更新記録つき

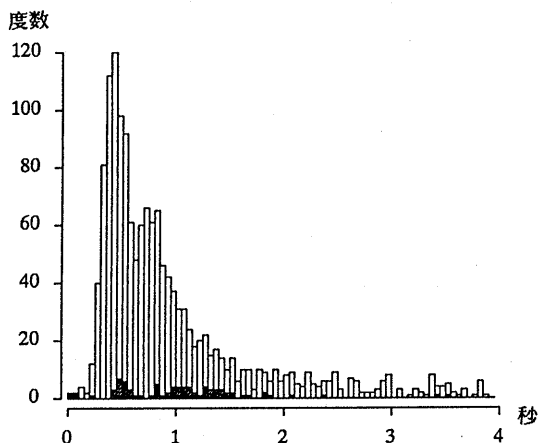


図7 「真」の変換判断時間の頻度分布。黒く塗ってあるのは、変換が起こらなかった場合

尾よく「巨大な」ができたことを確認して先（「けいさんきの」）を打ちかけた、という場面である。各欄の意味は左から順に、事象番号、事象名、通算時間（秒）、一つ前の事象以降の経過時間（秒）である。図6(b)の、事象名がSCREENである行の経過時間欄を合計すれば画面更新時間が得られる。たとえば「巨大な」に対する画面更新時間は、図6(b)の事象番号44にあるとおり70msecかかっている。この場合の画面の書き換えは一気に起こっており、したがってSCREENは1回しか記録されていない。この70msecを、図6(a)の65行目の0.416secから引けば真の変換判断時間（ここでは0.346sec）が得られる。

図7はその「真の変換判断時間」の頻度分布である。ただしかな文字列に対して変換を試みたあと、得られた変換結果が気に入らないためにBSキーを叩いて変換を取り消し、そのあとと思いなおしてまた変換キーを叩いたという場合における変換キーの打鍵に伴う変換判断時間は、再度得られる変換結果が先刻のものと同じであって予測がつく、という点で通常の「初回の変換結果」とは違うと考えてここからは省いた（全3件）。また図8には、画面更新時間の頻度分布を示す。全体としては、画面更新時間はごく短い、中には非常に長い事例がある。たとえば650msecの枠にも1度数あるが、これは「このとしに」から「この年に」を作ろうとしたところ、全体としては変換キーの7回の打鍵が必要であったが、その初回の変換において恐らくは「古野としの」ができ、その際に長い計算時間がかかったというもの、と考えられる。実際「松」の画面をビデオ記録にとって観察してみたところ、そういうことも十分あり得る、ということがわかった。

なお、図7のヒストグラムのの下の方に黒く塗ってあるのは、図7の分布のうち変換を試みたところ（辞書項目が登録されていなかったなどの理由によって）失敗に終わり、何の画面更新も起こらず、その結果画面更新時間（SCREEN行の所要時間欄の合計）が0となった、という事象における変換判断時間の分布である（図8の、0msecの枠の度数がこれに当たる）。この黒い部分の形状は、図7の頻度分布の一般的な形状と比べると多少ずれている。これは画面に何か変化が起こることを期待して待っていた利用者が、何も変化が起こらなかったためにぐっと詰まったところだ、とも解釈できる。このように、起こっている心理学的過程が違っている疑いがあるので、第4節の解析では頻度分布のこの部分を省き、実際に画面の書き換えが起こった場合についてのみ解析をおこなった。

ブレーバックはもとの打鍵ファイルをほとんどそのまま用いたが、ごくわずかの点（具体的には2箇所）で手入れが必要であった。またもとの実作業の際には、のちにブレーバックをすることになるとは予測していなかったため、用いた変換用辞書が保存されておらず、そのため辞書の原型にほぼ等しいと思われるものを再現するために多大の努力を要した。

3 他からの「ヒント」と新しい模型

[3]では、筆者の一人を含む9人の被験者に、一息でいえる程度の長さの例文を、打つべき文章をすっかり頭に入れたあとで一気に打鍵する、という方法で打ってもらった。打鍵は好みのワープロソフト（一太郎の第1版または第2版）でやってもらい、3回間違えずに打てるようになるまで繰り返した。そしてその3回の打鍵から変換判断時間を抽出し、頻度分布を調べた。これによって[1]の意味での山1だけを純粋な形で取り出すことができるのではないかと期待したが、結果は期待に反して正規分布とならず、むしろ Ψ 分布と解釈せざるを得ないようなものがあらわれた。

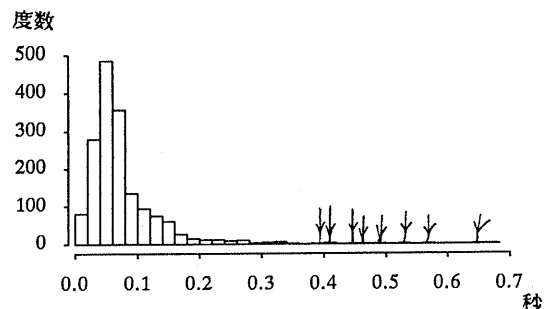


図8 「松」の画面更新時間の頻度分布。よく見ると矢印のところにわずかに度数がある

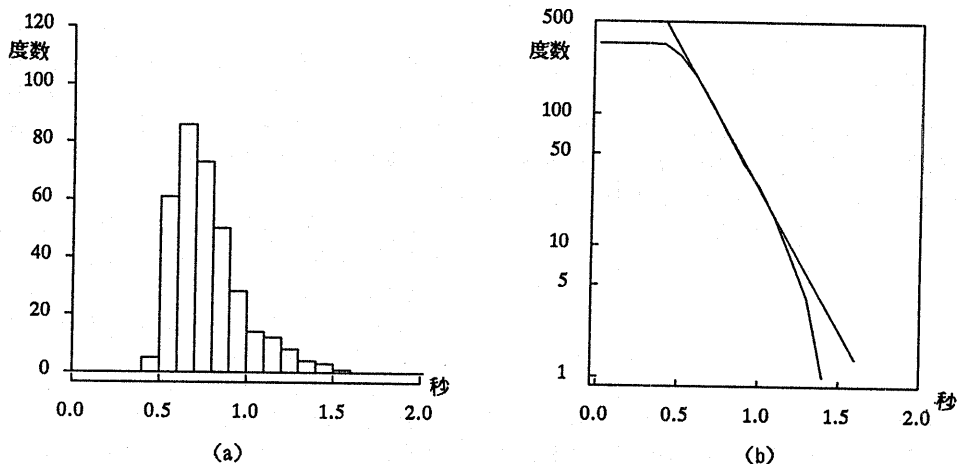


図9 「松」における変換判断時間の純粹成分 (a)頻度分布、(b)累積分布、半対数表示

図9は、同様の実験を「松」についてやってみた結果を示している。被験者は[3]にいう被験者Aであって、本文で解析している分布（図1および7の分布）を採取した際の被験者と同一人物である（実は筆者の一人）。画面更新時間の補正はしていない。図9(a)は変換判断時間のヒストグラム、図9(b)は対応する累積密度分布を半対数目盛りで示したものである。

この分布はいかにも Ψ 分布らしい形をしている。 λ を計算してみると0.202となり、これは[1]において山1を構成している四つの Ψ 分布のうち、一番入の小さいもの（ $\lambda_4 = 0.214$ 、[1]の表1参照）とも、よく照応している。

そこで図7の分布を次のように解釈しなおしてはどうか、とのアイディアが浮かぶ。まず分布の左側の、[1]で山2および山3から成るものと解釈したピークは、実は二つの正規分布から成るのではなく、図9(a)のような Ψ 分布にほかならない、と考える。また[1]にいう山1の最小の λ （上述）は、図9の Ψ 分布の λ と同じものであると考える。出ている値はわずかに違いますが、[1]における $\lambda_4 = 0.214$ は多段のあてはめの最後の段階で得られたものであり、あまり高い精度を期待できるようなものではないことを考慮して、むしろ図9の $\lambda = 0.202$ （これを以下 λ_0 と書く）の方を採用する。そして図3を、図10のように改訂する。または言葉であらわせば次のとおりである。

- 利用者は画面に表示された変換結果を見てその可否を判断し、その結果いかに応じて適当なキーを打鍵する。以上に要する時間は、結果を見るところと打鍵するところは正規分布、判断するところは指数分布に、全体としては Ψ 分布にしたがう。
- ただし(a)における可否の判断の結果として、はじめに思っていたのと違うことを打鍵したくなる場合もある。

その場合、打鍵が本当にはじまる前には運動器官の態勢を整えなおす必要があり、そのためにある一定の打鍵準備時間が必要となる。それは恐らく正規分布にしたがう。

- (b)において違うことを打鍵したいという、その「違うこと」は、ただちに具体的には思い浮かばず、ある程度深い思考過程の結果としてはじめて定まる場合もある。それらの思考過程には種々の深さのもの（用語用字に関するもの、文章全体の構想に係る戦略的なもの、など）があり、それぞれある確率をもって発火する。
- 一旦発火すると思考過程は、思考の深さに応じたある時定数をもつ指数分布にしたがって継続する。その時定数は、深い思考に関わるものほど長い。

このように、「違うこと」を打ちたくなった場合、余分の思考時間がかからないこともあり得るものとする。図10の下から4行目がそれをあらわしている。その場合総所要時間は、図9の Ψ 分布と同じ λ （ $=\lambda_0$ ）を持ち、 μ と σ は打鍵準備の部分からくる不確実性に対応して若干大きいかも知れない、というような Ψ 分布にしたがうことになる。その μ および σ を μ' および σ' と記す。また、そのときの Ψ の確率密度関数を Ψ'_0 と記す。

「違うこと」を打ちたくなったとき、その「違うこと」を考え出すために余分の思考時間が必要となった場合（図10の下3行分）には、上の（ λ_0 、 μ' および σ' で定める） Ψ 分布、つまりに Ψ'_0 の分布に、さらにその余分の思考時間がしたがう指数分布が（標本同志が足し合わされるという意味で）組み合わされることになる。その余分の指数分布の時定数は場合ごとに異なる。それを λ_n とし、 $\lambda = \lambda_n$ 、 $\mu = \mu'$ 、 $\sigma = \sigma'$ であるような Ψ を Ψ'_n とすると、総思考時間のしたがう分布の確率密度関数 Θ_n は、

$$\Theta_n = \frac{1}{\lambda_n - \lambda_0} \cdot [\lambda_n \Psi'_n - \lambda_0 \Psi'_0] \quad (3)$$

となる。実際、時定数 λ_n をもつ指数分布から取り出された標本、時定数 λ_0 をもつ指数分布から取り出された標本、および平均 μ' 、標準偏差 σ' をもつ正規分布から取り出された標本の和の分布の確率密度関数は、

$$\int_0^x \frac{1}{\lambda_n} e^{-\frac{x-y}{\lambda_n}} \cdot \frac{1}{\lambda_0} e^{-\frac{y}{\lambda_0}} dy$$

$$= \frac{1}{\lambda_n - \lambda_0} \cdot [e^{-\frac{1}{\lambda_n}} - e^{-\frac{1}{\lambda_0}}]$$

となり、 Θ_n の分布はこの分布から取り出された標本と、平均 μ' 、標準偏差 σ' の正規分布から取り出された標本の和の分布に当たっているからである。

分布のうちの山2A以外の部分のことを山1Aと呼ぶことにする。上の解釈によれば、これは「違うこと」の打鍵のための山である。

4 再当てはめと解釈

この節では、図7の頻度分布を図10の模型に当てはめてみた経過と結果を述べる。

第2節で述べたように、本節の当てはめは図7の分布から、さらに黒く塗られた部分(変換キーを叩いたのに変換が起らなかったケース)を除外したものについておこなう。それを図11(a)に示す。

[1]での解析では、図1の分布が右の方に向かって引いている長い尾に着目し、 λ の大きい順にそこから Ψ 分布をはがしてくるという方針をとった。当初同様のことを図11についてやってみたが、図9の右端を裁ち落としたような形のものが残った。このように、右から先にはがしたのは削りすぎになってしまう。これは図9の形のものが相当長い尾を引いていることに起因することである。

そこではぎとりは逆に左から、まず図9に相当するもの

をはがす、という手順ですることにした。そうした場合困るのは、[1]でやった長い尾に着目する、という手が利かないことである。話をはつきりさせるため、出発点でのその Ψ 分布の λ は、図9における0.202をそのまま採用し、一応最大点が0.40秒のところで達成されているものと考えて解析を進めた。

そうやって左側の山(山2A)をはがしたあと、残ったものに[1]でおこなったと同じような解析を施す。ただしその際第3節(3)式を考慮して、はがされた Ψ 分布((3)式の Ψ'_n)に見合う Ψ'_0 成分を、解析中の度数分布に足し戻してやらなければならない。

しかるにそのようにしてみたところ、 $\lambda=0.202$ から出発したのではうまく行かないことがわかった。図10の考えが正しいとすれば残ったものの中から同じ λ に対応する成分が出てくるはずである。ところが最後に残ったものは3秒前後の成分で、つじつまが合わなかった。

そこで見当をつけて $\lambda=0.19$ から出発してみたところうまく行き、図11(b)~(e)のような結果が得られた。すなわち、まず図9に対応して図11(b)が切り取られ、残余(c)に対して(d)のような当てはめがなされ、全体として(e)のような当てはめ結果が得られた。結果の数値的なまとめを表1に示す。図11(d)には、下から順に λ_2 、 λ_3 、 λ_4 の寄与を示す境界線が記入してある。 λ_1 の寄与はあまりに小さくて、ここには図示できない。

λ_4 は、図10の考えによれば $\lambda=0.19$ と等しくなければならないはずであるが、実際には0.203になっている。しかしながらこの値は最小自乗法によって決まる値であり、当てはめ区間の選びかたにかなり敏感である。したがってこの程度合えば十分な一致が得られたと見てよい。

この当てはめ結果について、時間軸を総計63区間に区切って χ^2 値を計算してみたところ、値51.79が得られた。区切りかたは先に[1]の当てはめについて谷越[2]が採用したと同じものとした。これを谷越の得た値60.15と比較するとより小さく、当てはめに使う分布の種類が減っておりひい

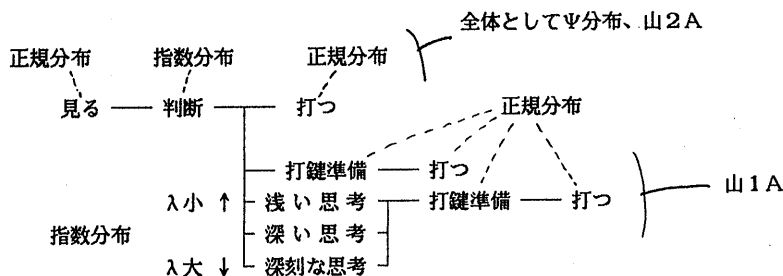


図10 統計的模型(改訂新版)

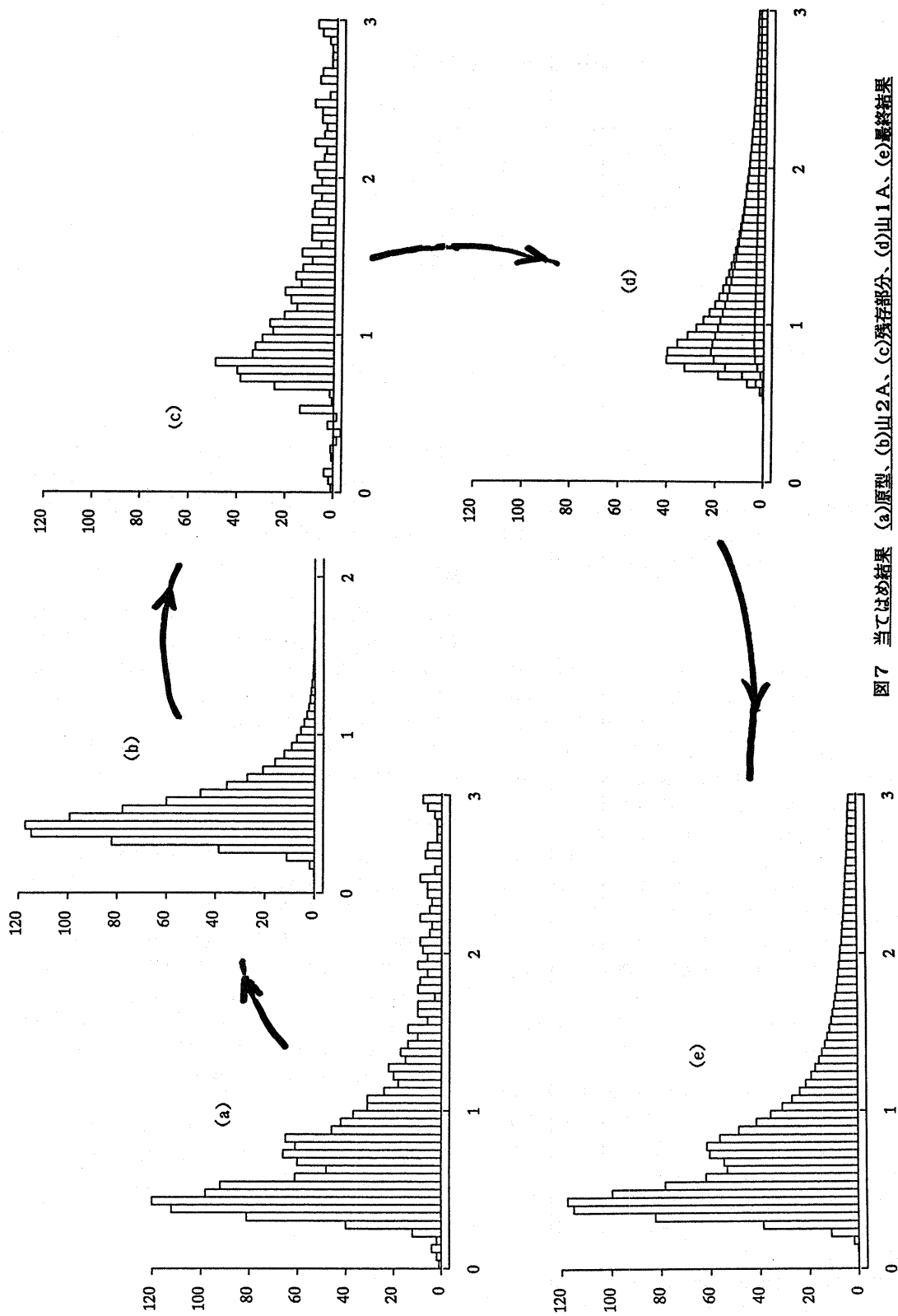


图7 当てはめ結果 (a)原型、(b)山2 A、(c)残存部分、(d)山1 A、(e)最終結果

表1 思考時間分布の当てはめ結果

山1A

n	当てはめ範囲	λ_n	重み	関数
1	20~49.20 ^秒	19.91 ^秒	11.34 ^件	Θ_1
2	4~12	3.67	291.4	Θ_2
3	1.6~2.2	0.754	248.1	Θ_3
4	0.7~1.0	0.203	235.9	Ψ_4

$$\mu' = 0.698 \quad \sigma' = \sigma = 0.063$$

山2A

$$\lambda = 0.190 \quad \mu = 0.325 \quad \sigma = 0.063 \quad \text{重み} = 804(\text{件})$$

ては自由度が増していることとあいまって、一致は一層高まったといつてよい。

なお右側の山の解析の際に使う σ は、第3節にいう σ' であり、したがってそれは山2Aにおける σ より小さくはないはずである。ここでは山2Aの形から自然に得られた σ をそのまま σ' として使用した。

5 討論

以上のように図7のデータには、かなりもってもらしい解釈を与えることができることがわかったが、もちろん残された問題はたくさんある。第1に、第4節の当てはめは部分的に目分量を含んでいる、という点が気になる。事情は[1]におけると同様である。[1]における、これに対応する問題点は谷越[2]の再解析によって解決された。それに相当することをこれからやって行く必要がある。

第2に、今回図9の実験結果をもとに解析を進めたのと同様に、他の実験事実から傍証を集めて話を一層はっきりさせることが望ましい。また第3に、やはり[1]について谷越がしたことであるが、本文の考えを他の被験者、および他のデータについて適用し、テストしてみることが望ましい。

本文の結果はどんな役に立つだろうか。とりあえず次のようなことが考えられる。

1. 本文の結果はCardら[4]が彼らの打鍵レベル模型についていっている「熟練者が誤りなく遂行した作業」の性格づけに役立つ可能性がある。つまり、それは山2Aのことだ、というような性格づけである。
2. 本文の結果は会話型システム上で利用者が貰やす時間を、システム側の時間と利用者側の時間に切りわけける目安を与えるかも知れない。
3. ここで考えた指数関数的な頻度分布は、どうもCardらのモデル人間プロセッサ[4]の枠内で考えていたのではう

まく説明しきれないように見える。こういう分布が実在すること自体は確かのようなので、これを説明しようとしてみるのが頭脳の機構の一層の理解の上で、進歩に向けての有効な刺激のもとになるかも知れない。

謝辞

統計学の利用に関し、貴重な助言を賜った森村英典教授に深謝する。また統計的解析の工具類を整備してくれた谷越浩一郎、第3節の実験データの収集に協力してくれた根岸 康、第2節の実験のためのインターフェースを製作してくれた山口 敏の各氏に感謝する。

なおこの研究に関して昭和61年度科学研究費補助金一般研究(C)第61580020号による補助を受けたので付記する。

参考文献

- [1] 木村、粕川、谷越：ワープロ利用者の思考時間に関する統計的模型、情報処理学会研究報告（日本語文書処理研究会）JDP-8-3（1986.9）
- [2] 谷越、木村：ワープロ利用者の思考時間模型再論 — 他の被験者ではどうか？、情報処理学会研究報告（日本語文書処理研究会）JDP-11-2（1987.3）
- [3] 木村、根岸、谷越、粕川：日本語ワープロ向け新打鍵レベル模型における思考時間パラメータの直接測定、情報処理学会第34回全国大会、5X-3（1987.4）、pp.1337-1338
- [4] S.K.Card, T.P.Moran & A.Newell: The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, New Jersey(1983).