

コマンド予測シェル使用時のユーザのキー入力の負担

西村俊和 美濃導彦 池田克夫

京都大学工学部

本論文では、ユーザのキー入力の負担を数値化することにより、コマンド予測シェルの予測の正答率と候補提示のユーザインタフェースとの関係について考察し、コマンド予測シェルの実現可能性について論ずる。

コマンド予測シェルは、入力コマンドを予測してその候補を提示するコマンドインタプリタであり、ユーザは候補を選択することによりコマンドを実行できる。コマンド予測シェルがユーザのキー入力の負担を減少させる条件は、シェルの予測部と、候補提示を行うユーザインタフェース部との間の条件に依存している。そこで、ユーザが候補を選択する際の負担をユーザインタフェース部のパラメータとし、これを数値化して候補の提示がユーザの負担の減少になるための条件を求めた。その結果、コマンド予測シェルがユーザの負担を減少させるのは、第一候補ではその正答率、第二候補以下の候補では候補が提示された場合における条件付き正答率が、ユーザインタフェース部のパラメータから決定される定数よりも大きい場合であることを示した。さらに、パラメータを情報理論を利用した手法とユーザの応答時間とによって求め、ユーザの負担を減少させるコマンド予測シェルが実現可能であることを示した。

The Cost of Key-typing with the Shell that Predicts Commands

Toshikazu NISHIMURA, Michihiko MINOH and Katsuo IKEDA

Faculty of Engineering, Kyoto University

Kyoto-shi, 606-01 Japan

The shell that predicts users' commands makes it easy to enter commands into UNIX with a character terminal. Command prediction is to choose command candidates by their occurrence possibility in histories. First, we evaluate the cost of typing commands and show that presenting the first candidate will make it easy to enter commands if the user is an expert. Second, we measure the intervals of key typing in using our shell and show that our shell accelerates typing rate if the hitting rate of the first candidate is more than 40 % for an expert user.

1 まえがき

コマンド予測シェルは、ユーザが次に入力すると考えられるいくつかのコマンドを順序づけして候補とし(コマンド予測)、それをユーザに提示するものである[6]。候補が正しければ、ユーザはコマンドの文字すべてを入力することなく、確認の操作だけでコマンドを入力することができる。そのため、ユーザはわかりにくい操作を憶えることなく簡単に利用することが可能であり、また、ユーザのキー入力による負担を減少させることができ、同時にタイプミスも減少できる。

コマンド予測シェルは、大きくコマンド予測部とユーザインタフェース部の二つに分けられる。コマンド予測部は、ユーザの入力するコマンドを予測してその候補を生成する。コマンド予測の候補は必ずしも正しくないため、これをそのまま実行することはできない。そこで、ユーザインタフェース部において、コマンド予測部で生成された予測候補をユーザに提示して選択させ、コマンドを実行する必要がある。

予測候補が正答した場合は、ユーザは確認の操作だけでコマンドを入力ことができ、ユーザのキー入力による負担を減少させることができる。しかし、提示された予測候補が誤っている場合、ユーザは誤った候補を拒絶した上に、より下位の候補を受諾したり、コマンド名の文字すべてを入力したりしなければならず、通常のシェルを使っている場合よりもかえって負担が増加するおそれがある。従って、コマンド予測シェルがユーザの負担を減少させるためには、予測候補が正答した場合に減少する負担と、候補が誤答した場合に増加する負担について十分な注意を払う必要があると考えられる。そこで、本論文ではユーザのキー入力による負担に注目し、コマンド予測シェルがユーザの負担を減少させる条件を示してコマンド予測シェルの評価を行うことを考える。

候補の受諾/拒絶のユーザの負担は、コマンド予測シェルのユーザインタフェース部における候補の提示法、あるいはそれを用いるユーザの資質等に左右される。また、正答/誤答の割合は予測の正答率である。コマンド予測シェルの実現のためには、より正答率の高い予測法とユーザの負担

ができるだけ軽い候補提示を行うユーザインタフェースが必要である。

両者は一般に互いに独立であると考えられるので、本研究では、コマンド予測シェル使用時の基本となるユーザの負担を等価なキー入力の負担を与える文字数として評価し、これをユーザインタフェース部のパラメータとすることにより、コマンド予測シェルがユーザの負担を減少させる条件をユーザインタフェース部のパラメータと予測部の正答率との関係として明らかにすることを試みた。これにより、コマンド予測シェルのある実現を単に評価するだけではなく、コマンド予測シェルの設計指針を与えることが可能である。さらに、ユーザインタフェース部の個々の実現に依存することなくパラメータの概略値を求め、その値が条件を満たしていることを明らかにし、コマンド予測シェルの実現可能性を示した。

以下、2章ではユーザのキー入力の負担の数値化を行い、3章では双方のパラメータ間の関係について考察し、コマンド予測シェルがユーザの負担を減少させる条件を明らかにする。続いて、コマンド予測シェルの実現可能性の検証を行うため、4章ではコマンド予測部の正答率について述べ、5章ではユーザインタフェース部のパラメータの推定について述べる。6章はむすびである。

2 予測候補を提示した場合のキー入力負担

2.1 候補提示

ユーザは、自分の心理的世界において行いたいこと(目的)を意識する。一方、システムの状態は、物理的世界において物理的に記述される。Normanは、双方の世界の隔たりを淵(Gulf)と呼んだ[1]。ユーザがシステムに働きかけるには心理的世界から物理的世界への「実行の淵(Gulf of Execution)」を越えねばならず、またシステムの状態を理解するためには、物理的世界から心理的世界への「評価の淵(Gulf of Evaluation)」を越えねばならないとした。

ユーザインタフェース部では、これをどのように実現した場合においても、提示された候補が何であるかを確認する動作と確認した結果すなわち

候補の受諾 / 拒絶をシェルに伝える動作が必要である。これらはそれぞれ、ユーザが自分の望む候補を選択するというタスクにおける「評価の淵」および「実行の淵」に相当する。そこで、これらがコマンド予測シェル使用時のユーザのキー入力の負担の基本単位と考え、同程度の負担となるキー入力の文字数で示す。すなわち、正の実数 S, K について、

1. 提示された候補を読みとって受諾 / 拒絶を判断する動作
= S 文字分のキー入力の負担と等価
2. 判断した結果をシェルに伝える動作
= K 文字分のキー入力の負担と等価

とする。

これらの値は、ユーザインタフェース部の個々の実現法やユーザによって異なる。ここではこれらをユーザインタフェース部のパラメータと呼ぶ。また、以下で単に負担と言った場合にはキー入力の負担を指すことにする。

2.2 予測候補

コマンド予測により、ユーザが次に入力するコマンドの予測候補が得られる。予測候補はユーザに提示され、適切な候補が選択される。通常、全ての予測候補を示すことはできないため、予測候補を確からしさの順に第一候補、第二候補、第三候補、…と並べ、表示装置の許すかぎり上位の予測候補を提示する。

なお、各正答率は 0 以上であり、累積正答率は 1 以下であるので、自然数 n に対して第 n 候補の正答率を R_n と表記すると、

$$R_n \geq 0, \quad (1)$$

$$\sum_{i=1}^n R_i \leq 1. \quad (2)$$

2.3 ユーザの負担

ユーザに予測候補を選択させる場合、候補以外のデータの表示が候補によって隠されるのを最小にするため、ここでは、一つずつ候補を示してその可否を問うことによりコマンドの選択を行う候補提示法を採用する。ユーザは、示されている候

補を拒絶することによって、第二位以下の候補を順番に選択することができる。

このような候補提示法においては、新たにコマンドの平均文字数を L と表すことにより、 S, K, R_1, R_2, \dots および L を用いて、ユーザの負担をそれと等価な入力すべき文字数として表わすことができる。

自然数 n について第 n 候補まで用いる場合の負担 $l(n)$ を考える。この場合、候補の一つを見て選択するためには、 $(S+K)$ 文字分の負担が必要であるから、第 i 候補が正答する場合、 $i(S+K)$ 文字分の負担が必要となる。これと第 i 候補の正答率である R_i との積をとることにより、第 i 候補が正答する場合の負担の期待値が得られる。一方、すべての候補が誤りであった場合、ユーザは自分でコマンドを入力しなければならない。コマンドの打鍵と候補を拒絶するための操作が異なるならば、ユーザは自分でコマンドを入力するだけで、最後の候補を拒絶することができるため、この場合の負担は $n(S+K) + L - K$ となる。負担の期待値は、これにすべての候補が誤答である割合 $1 - \sum_{i=1}^n R_i$ を乗ずれば良い。

従って、 $l(n)$ は次式で与えられる。

$$l(n) = \left\{ \sum_{i=1}^n R_i \cdot i(S+K) \right\} + (1 - \sum_{i=1}^n R_i) \{ n(S+K) + L - K \} \quad \text{[文字分]}. \quad (3)$$

3 ユーザの負担の減少のための条件

ユーザインタフェース部とコマンド予測部の間には一般に関連がなく、 S, K は、ユーザインタフェース部のパラメータ、 R_1, R_2, \dots はコマンド予測部の予測率であるので、 S, K と R_1, R_2, \dots とは独立という仮定を置く。 L は、対象とするシステムのコマンド体系およびユーザのコマンドの使用頻度等に依存するが、一般にユーザインタフェース部、コマンド予測部の個々の実現に左右されないため、ここでは単に定数と考えることに

する。

コマンド予測では、一般に候補の数を多くすればそれだけ累積正答率が大きくなる。一方、ここで用いる候補提示の方法では、下位候補を選択するためには候補の数だけ選択 / 拒絶を行わなければならないため、あまり多くの候補をユーザに提示すると、かえってユーザの負担を重くする可能性がある。そこで、候補の提示個数に注目し、ユーザの負担の大小関係と R_1, R_2, \dots および S, K の値との関係について考察する。

まず、コマンド予測が有効であるための条件を求める。このためには、第一候補の提示が、普通にコマンドを入力する場合と比べてユーザの負担を減少させるような条件を求めればよい。式(3)より、

$$l(1) - L = -R_1(L - K) + S.$$

従って $l(1) < L$ 、すなわち第一候補がユーザの負担を減少させるための条件は、

$$R_1(L - K) > S. \quad (4)$$

式(2)より、 $R_1 \leq 1$ であるから、 $S + K \leq L$ でなければならない。 $K > 0$ だから、 $S + K \leq L$ ならば $L - K > S > 0$ である。 $S > 0$ かつ $R_1 \geq 0$ なので、式(4)の条件は、次式のように書き直すことができる。

$$L > K + S, \quad \text{かつ} \quad (5)$$

$$R_1 > S / (L - K). \quad (6)$$

次に、候補の数とユーザの負担について考察する。式(3)より、2以上の自然数 n について、

$$l(n) - l(n-1) = -R_n(L - K) + (1 - \sum_{i=1}^{n-1} R_i)(S + K).$$

従って $l(n) < l(n-1)$ 、すなわち候補を多く使えばそれだけ負担が減少するための条件は、

$$R_n(L - K) > (1 - \sum_{i=1}^{n-1} R_i)(S + K). \quad (7)$$

第一候補がユーザの負担を減少させる条件を満たしている場合は式(5)が成立しているので、 $L - K > S > 0$ 、また条件式(2)より $(1 - \sum_{i=1}^{n-1} R_i) \geq$

0 である。従って、 $(1 - \sum_{i=1}^{n-1} R_i) > 0$ の場合¹には、式(7)の両辺を正数 $(1 - \sum_{i=1}^{n-1} R_i)(L - K)$ で除することにより、以下のように書き直すことができる。

$$R_n / (1 - \sum_{i=1}^{n-1} R_i) > (S + K) / (L - K). \quad (8)$$

$\sum_{i=1}^{n-1} R_i$ は、第 $n-1$ 候補までの累積正答率であるので、 $(1 - \sum_{i=1}^{n-1} R_i)$ は、第 $n-1$ 候補までに正答がない割合、すなわち第 n 候補を提示する割合である。第 n 候補の正答率をこれで割算したものは第 n 候補が提示された場合における条件付き正答率であるので、以上の条件は次のように簡潔にまとめることができる。

コマンド予測シェルが有効であるのは、 $L > K + S$ かつ第一候補の正答率がユーザインタフェース部のパラメータから決定される定数 $S / (L - K)$ よりも大きい場合に限る。また、第 n 候補が有効であるのは、第 n 候補が提示された場合における条件付き正答率が、ユーザインタフェース部のパラメータから決定される定数 $(S + K) / (L - K)$ より大きい場合に限る。

以下では $S / (L - K)$ を第一候補の正答率の下限、 $(S + K) / (L - K)$ を第 n 候補の条件付き正答率の下限と呼ぶ。

4 コマンドの平均文字数とコマンド予測の正答率

前節の結果では、ある候補がユーザの負担を減少させるか否かは、コマンド予測部の正答率と、ユーザインタフェース部のパラメータのみから決定される値との簡潔な大小関係に依存することを示した。そこで、次に論点となるのは、ユーザの負担を減少させるコマンド予測シェルが実現可能か否か、すなわちこれらの大小関係を満足するようなユーザインタフェース部、コマンド予測部が実現可能であるか否かである。

そのためにまず本節では、コマンド予測の一例とその正答率を示す。続いて次節でインタフェース部のパラメータを求め、候補によってユーザの負担が減少する条件を満たすことが可能なことを

示す。これによって、ユーザの負担を減少させるコマンド予測シェルが実現可能であることが明らかにできる。

コマンドの平均文字数とコマンド予測の正答率を示すため、UNIXの熟練者18名から合計約40万行のコマンドの履歴(ヒストリ)を採集した。

表 1: 予測の正答率

行数	R1 (%)	R2 (%)	L (文字)
4954	21.9	12.1	4.53
14802	30.2	11.4	4.10
12835	22.4	11.7	4.88
5049	30.4	13.1	4.60
6840	31.3	13.9	4.47
85987	31.6	15.3	3.89
33268	33.8	14.2	4.10
3172	23.0	10.9	4.54
15763	27.5	11.8	4.30
13517	29.7	14.9	4.43
11780	43.2	15.4	4.06
6479	29.5	13.9	4.36
3309	26.3	13.4	4.86
13634	31.4	14.9	3.99
21040	32.7	15.9	4.39
42736	24.3	11.1	4.71
13353	24.4	12.5	4.34
61604	36.8	15.4	4.07
平均	29.5	13.4	4.12

4.1 コマンドの平均文字数

UNIXではコマンドによって文字数が異なり、また、ユーザによってその使用頻度も異なる。従って、コマンドの平均文字数は個人差があると考えられる。そこで、採集したヒストリでのコマンドの平均文字数をユーザ別に表1に示す。コマンドの平均文字数は3.89から4.88の間にあり、平均は4.12であった。

4.2 コマンド予測部の正答率

UNIXでは基本的な働きをするコマンドが用意されているため、ユーザはそのいくつかを組み合わせて作業を進めることが多い。従って、前後のコマンドの関係を利用することにより、コマン

ドを予測することができる。ここでは、コマンドとその直後に続き得るコマンドとを組にしてLRUバッファに入力し、ユーザが直前に入力したコマンドに続き得るコマンドをバッファ内から検索することによって予測を行う。ユーザ別の数値を表1に示す。この手法では、第一候補、第二候補の正答率は、それぞれ約30%、約13%であった。

5 ユーザインタフェース部のパラメータの推定

ユーザインタフェース部のパラメータはユーザの負担に関する量であるが、これを定量的に求めるのは簡単でない。人間の負担に関する量を求める際によく用いられるのが、Stimulus-Response compatibility(以下S-R法と呼ぶ)である[2]。これは、人間に何らかの仕事を与えて、応答時間をその仕事に必要な負担であると考えたものである。

例えば、タイプライタのキー配列を最適化する研究においては、キー配列の優劣の判断基準として、ユーザの負担減少が用いられている[3]。この際、一般によく用いられているのは、入力速度である。この分野の研究は以前から行われており、QWERTY配列やDvorak配列等がよく知られている。

しかし、ユーザの仕事に対する応答時間は「実行の淵」と「評価の淵」の負担の和に比例するため、「実行の淵」「評価の淵」それぞれの負担を個々に直接求めることはできない。そこで、ここでは情報理論を応用して、ユーザインタフェース部のパラメータの一つである K を推定する手法を提案する。また、この値を利用し、従来のS-R法の実験結果から、もう一つのパラメータである S の概略値を推定する。これらのパラメータの推定手法は、ユーザインタフェース部の個々の実現法にあまり依存しない。

5.1 K の推定

Cardらは、参考文献[4]で“不確定性原理(Uncertainty Principle)”について述べている。

不確定性原理

人が物事を選択を決定する時間 T は以下の

式で与えられる。

$$T = I_C \cdot H$$

ただし、 H は決定に関する情報理論におけるエントロピーで、 I_C は定数。

また、 a 個の選択では単純に $H = \log_2(a+1)$ とする。

T がユーザの負担に比例していると仮定することにより、キーボードのキーの決定に関するエントロピー H_k および候補選択に関するエントロピー H_c を用いて K を表すことができる。すなわち、

$$K = H_c/H_k.$$

候補選択に関するエントロピー H_c は、予測の正答率によって変化する。しかし、予測の正答率が極端に低いあるいは極端に高い場合を除き、候補が正答しているか否かは事前にユーザにはわからないため、予測の正答率は H_c にあまり影響しないと考えることができる。ここでは、候補選択は単に受諾および拒絶の二者択一の選択をしているものとみなし、

$$H_c = \log_2(n+1) = \log_2(3) \text{ (ビット)}$$

と置く。

タイプライタのキー配列を最適化する研究で対象されている熟練したキーパンチャと異なり、ここで対象とする計算機のユーザでは、キーボードに対する熟練度は非常に広い範囲を持ち、各々の入力速度はさまざまである。従って、キーボードのキーの決定に関するエントロピー H_k も、ユーザの熟練度によって大きく左右されると考えられる。そこでここでは、キーボードに不慣れたユーザとキーボードにある程度慣れたユーザに分け、それぞれのユーザごとに H_k を求めることを考える。以降では、それぞれキーボードに不慣れたユーザをキーボード初心者、キーボードにある程度慣れたユーザをキーボード熟練者と呼ぶ。

キーボード初心者は、各キーの位置を正確に把握していないか、あるいは手指全てではなく、人差し指のような特定の指のみでタイプすることが多い。そのため、キーボード初心者がキータイプをすることは、アルファベットの小文字 26 文字

のいずれかを選択しているものとみなすことができる。従って、

$$H_k = \log_2(n+1) = \log_2(27) \text{ (ビット)}$$

であり、

$$K = H_c/H_k = \log_2(3)/\log_2(27) = 1/3.$$

キーボード熟練者は手指全てを使ってタイプを行うことが可能であり、各キーの位置をある程度把握している。そのためキーボード熟練者におけるキーボードのキーの決定に関するエントロピー H_k は単純な n 個の選択よりも小さいと考えられる。

英語のアルファベットを単純マルコフ情報源の出力とみなした場合の条件付きシンボル確率は F. Pratt によって与えられている。Abramson はこれをもとに、英語のアルファベットの情報量をシンボルあたり 3.32 ビットと計算した [5]。ここでは、この値を各キーの決定に関するエントロピーとみなし、キーボード熟練者の K を求める。

$$H_k = 3.32 \text{ (ビット) の時、}$$

$$K = H_c/H_k = \log_2(3)/3.32 \approx 0.477.$$

5.2 S の推定

$S+K$ は、ユーザが提示された候補を読みとって正否を判断し、その結果をシェルに伝えた場合のユーザの負担を、キー入力文字数で表したものである。従って、S-R 法の応答時間をユーザの負担と考えると、ユーザが提示された候補に回答する時間を、通常のキータイプの時間で除することにより、 $S+K$ を求めることができる。これから前節で推定した K を引くことによって S の推定が可能となる。ここではユーザが提示された候補に回答する時間に注目する。

なお、一般に S-R 法の応答時間は個人差が大きいため、 S の値を一律に定めることはできない。ここでは、実験で得られた数値の範囲をそのまま用いて S の値の範囲と典型的な値を求め、 S の概略値を明らかにすることにする。

Card らがまとめた過去の S-R 法の実験に、Physical matches と Name matches というものがある [4]。両者とも、二つのシンボル (アルファ

ベット)を一回に一つずつ連続して被験者に提示し、二回目のシンボルが最初のシンボルと同じかどうかをボタンを押して答えさせるもので、前者の実験が同じ形をしているか否か問うのに対し、後者は同じ呼び方をするか否か問うものである。これは、ユーザの考えている候補と提示された候補が同じか否かという仕事によく似ていると考えられる。前者の実験では 130ms から 640ms の範囲で、典型的には 310ms、後者の実験では 155ms から 810ms の範囲で、典型的には 380ms の値を持つ。

一方、通常のタイプの時間についても、Cardらは過去の実験をまとめている。タイプの時間はタイピストの技能によって大きく異なるが、キーに慣れていないタイプの初心者の文章入力では 1154ms (Devoe による)、平均的なタイピストをキーボード熟練者とみなすと、その文章入力では 158ms から 231ms (Hershman と Hillix による)である。

キーボード初心者の場合、 $K = 1/3$ と考えると、 S は最大 $810/1154 - 1/3 \approx 0.369$ となる。最小の値および典型的な値を求めると負の数となるが、これは S が K に比べて非常に小さい正数であることを表していると考えられる。

キーボード熟練者の場合、 $K = 0.477$ と考えると S は最低 $130/231 - 0.477 \approx 0.0868$ 、最高では $810/158 - 0.477 \approx 4.65$ の範囲になる。また、 S の典型的な値の範囲は $310/231 - 0.477 \approx 0.865$ から $380/158 - 0.477 \approx 1.93$ までとなる。

5.3 考察

測定した L 、推定した S, K により、第一候補の下限および第 n 候補の条件付き正答率の下限を求めることができる。これを表 2 にまとめる。

S の値の範囲が広いいため、第一候補の正答率の下限および第 n 候補の条件付き正答率の下限の範囲も広がっている。キーボード初心者の場合は S の最大の値が比較的小さいため、第一候補の正答率は 10.4% 以上あれば、どのようなキーボード

初心者に対してもユーザの負担を減少することができる。

キーボード熟練者の場合、第一候補の正答率の下限はほぼ 0% から 100% までの値を持っており、全てのキーボード熟練者に対して負担を減少することはできない。しかし、典型的な値は 30.7% から 40.5% であるので、第一候補の正答率をこの値より大きくすることにより、多くのキーボード熟練者に対して負担を減少させるコマンド予測シェルが実現可能であると考えられる。

次に、4 節で示したコマンド予測法を用いてコマンド予測シェルを構成することを考える。第一候補の正答率は約 30% であるので、キーボード初心者の負担を減少することができる。一方、この正答率は、典型的なキーボード熟練者の第一候補の正答率の下限を下回っているので、多くのキーボード熟練者に対して必ずしも有効ではないが、一部のキーボード熟練者に対しては負担が減少する。多くのキーボード熟練者熟練者を対象とする場合には、第一候補の正答率をさらに 10% 程度は改善する必要がある。以上のことから、4 節のコマンド予測法を用いることによって、キーボード初心者および一部のキーボード熟練者を対象とした、負担を減少させるコマンド予測シェルが実現可能であることが明らかになった。

第一候補の正答率を 1 から引いた値、即ち第一候補の誤答率に、第 n 候補の条件付き正答率の下限を乗ずることにより、第二候補の正答率の下限を求めることができる。第一候補の正答率を 30% とした場合、第二候補の正答率の下限は $(1-0.3) \cdot \{(S+K)/(L-K)\}$ となる。キーボード初心者の場合、この値は最大 13.0% であり、4 節の予測法の第二候補の正答率も約 13.0% であるので、殆どのキーボード初心者にとって第二候補の提示も有効であると考えてよい。一方、典型的なキーボード熟練者の場合、第二候補の正答率の下限は 25.2% となり、4 節の予測法の第二候補の正答率はこれを大きく下回っている。従って、多くのキーボード熟練者にとっては、この予測法の第二候補の提示は負担がかえって増加することがわかる。

¹ $(1 - \sum_{i=1}^{n-1} R_i) = 0$ の場合、式 (2) より $R_n = 0$ であるので、通常は考慮する必要はない (この場合は $l(n) \geq l(n-1)$ である)。

表 2: S , K と 正答率の下限

	キーボード初心者			キーボード熟練者		
	最小	最大	典型的	最小	最大	典型的
S	0*	0.369	0*	0.0868	4.65	0.865 ~ 1.93
K	—	—	1/3	—	—	0.477
第一	0*	0.104	0*	0.0197	1*	0.307 ~ 0.405
第 n	0.0733	0.185	0.0880	0.121	1*	0.360 ~ 0.652

—: 未定義 0*: 値が0未満 1*: 値が1より大

S : 提示された候補を読みとって正否を判断する動作の負担 [文字分]

K : 判断した結果をシェルに伝える動作 [文字分]

第一: 第一候補の正答率の下限 $S/(L-K)$

第 n : 第 n 候補の条件付き正答率の下限 $(S+K)/(L-K)$

6 むすび

コマンドをキーボードで入力する環境において入力を支援する手法として、コマンド予測シェルに着目し、ユーザのキー入力の負担を数値化して予測の正答率と候補提示のユーザインタフェースの関係について考察した。ユーザが候補を選択する際の負担を、等価なキー入力の負担を与える文字数として数値化し、コマンド予測シエルのユーザインタフェース部およびコマンド予測部双方のパラメータ間の関係を明らかにした。また、ユーザインタフェース部のパラメータを求める手法について考察した。

コマンド予測シェルが多くのキーボード熟練者に対して有効となるためには、第一候補の正答率が40%程度必要であることが明らかになった。そこで、より正答率の高い予測法が必要となる。筆者らは、参考文献[7]の手法が有力であると考えているが、その有効性を示すことが今後の課題である。

ユーザインタフェース部のパラメータである S の値は非常に個人差が大きいため、さまざまなユーザ、特に S が非常に広い範囲をもつキーボード熟練者を対象とする場合、ユーザごとに S の値を定める必要がある。各ユーザごとにあらかじめ、あるいはコマンド予測シェルが動作中に、ユーザの応答時間から S の値を定める等の工夫も今後の課題である。

参考文献

- [1] Norman, D. A. and Draper, S. W.: "User Centered System Design", Lawrence Erlbaum Associates (1986).
- [2] John, B. E. and Newell, A.: "Cumulating the Science of HCI: From S-R Compatibility to Transcription Typing", Proc. CHI '89, pp. 109-120 (1989).
- [3] Matias, E., MacKenzie, I. S., Buxton, W.: "Half-QWERTY: A One-handed Keyboard Facilitating Skill Transfer From QWERTY", INTER-CHI '93 Conference Proc., pp. 88-94 (1993).
- [4] Card, S. K., Moran, T. P. and Newell, A.: "The Psychology of Human-Computer Interaction", Lawrence Erlbaum Associates (1983).
- [5] アブラムソン著, 宮川洋訳: "情報理論入門", 好学社 (1969).
- [6] 西村俊和, 廣瀬勝一, 美濃導彦, 池田克夫: "コマンド予測シェル使用時のユーザのキー入力の負担について", 信学技報 HC91-11 (1991).
- [7] 西村俊和, 廣瀬勝一, 美濃導彦, 池田克夫: "マン・マシンコミュニケーションにおける文脈の追跡", 信学論 J78-D-II, No. 1, pp. 105-114 (1995).
- [8] 淵一博 監修, 古川康一 溝口文雄 共編: "インタフェースの科学", 第二章 ユーザのためのシステム設計, 宮田義郎, pp. 23-48, 共立出版 (1987).