

確率過程に基づくアイコンの呈示

守田 了 大塚 達 田中 稔

山口大学工学部

あらまし

我々は、アイコンを用いた作業空間 (アイコンシステム) を開発している。本システムでは、2つのアイコンを重ねあわせるという単純な操作の繰り返しによって、現実世界の作業を模擬している。ユーザは呈示されたアイコンにより、作業空間を把握する。しかし、システムが大規模でアイコンの数が多き場合に、作業と関係のないアイコンの呈示は、目的のアイコンを探す妨げになる。この問題を解決するために、アイコンを階層的に表示する手法が提案されているが、システム設計者が意図する階層とユーザが意図する階層が一致しないため、扱いにくさが生じる。そこで本稿では、アイコンに階層関係を与えずに、操作履歴を用いてユーザが次に使用するアイコン郡を予測し、選択的に表示するユーザ支援環境を作成する。この問題を確率過程に基づき、時系列予測問題として定式化する。バッチ依存の処理とクラス依存の処理について、提案したモデルの有効性を評価する。

和文キーワード ユーザインターフェース 確率過程 アイコンシステム

Iconic Selection based on Stochastic Process

Satoru MORITA Toru OOTUKA Minoru TANAKA

Faculty of Engineering, Yamaguchi University

2557 Tokiwadai, Ube, 755, Japan

Abstract

We have developed an iconic work space, in which the tasks of overlapping an icon on another icon simulate the work in real world. End-users grasp the work space by observing icon sets. When a great number of icons exists, icons, not relating to the users' work, prevent them from identifying for a target icon. To solve this problem hierarchical environment of icons was proposed. However the hierarchy proposed by the system designer sometimes doesn't match with the end-users need. So our system selects icons using the task history without hierarchical relation of icons defined by the system designer. We formulated this problem as time sequence estimation problem using stochastic process. The effectiveness of the proposed model on the batch dependent and class dependent processing were evaluated.

英文 key words user interface stochastic process iconic programming system

1 はじめに

我々は、アイコンを用いた作業空間（アイコンシステム）を開発している [5]。本システムにおいて、アイコンは現実世界のもののメタファであり、2つのアイコンを重ねあわせるという単純な操作の繰り返しによって、現実世界の作業を模擬している。ユーザは定義されたアイコンにより作業空間を把握する。システムが大規模化でアイコンの数が多の場合に作業と関係のないアイコンの表示はユーザが目的のアイコンを探す際に無駄な時間がかかる原因となる。この問題を解決するために、アイコンを階層的に表示する手法が提案されている [1]。しかし、システム設計者が意図する階層とユーザが意図する階層が一致しないため、扱いにくい。そこで本稿では、アイコンに階層関係を与えずに、操作履歴を用いてユーザが次に使用するアイコン群を予測し、選択的に表示するユーザ支援環境を実現する。

近年、グラフィックインターフェースにおいてユーザが操作する時の繰り返しボタンをシステムが自動的に検出して次の操作を予測するシステムやユーザの示す例からの汎化により操作手順を示すプログラムを生成したりするシステムの研究が盛んである [3][4]。しかし、各種の研究が行なわれているにもかかわらず、予測を使用したインターフェースは実際にはほとんど使用されていないのが現状である。

マルコフ確率過程を用いて画像の修復を統計的に行なう手法が提案されている [7]。また、音素認識処理に統計的手法が用いられている例が紹介されている [6]。確率過程に基づくアプローチは、時刻とともに推移する状態空間からの予測に適している。そこで、ユーザの操作履歴を用いて、次に使うアイコンを予測し提示するユーザ支援を確率過程に基づき定式化する。

2章では、ユーザ支援のためのアイコンの選択的表示を、確率過程に基づく時系列予測問題として定式化する。アイコンを通じて行う処理は、アイコン操作の順番が一定の処理（バッチ処理）と、一定の期間同じアイコン群を使う処理（クラス処理）とに分けられる。2つの処理は性格が大きく異なるため、同じ戦略でアイコンを表示することは適当ではない。そこで、操作の流れに応じて処理の識別を行い、処理に応じたアイコン集合の表示を行う。アイコンの選択は次の3つの過程で行われる。まず、確率過程に基づく操作履歴の収集を行い（3章）、次に確率分布による処理の識別を行い（4章）、処理に応じたアイコン集合の表示を行う（5章）。ユーザの操作履歴に基づいて、しだいにユーザの意図を把握しアイコンの選択的表示を実現する。n初期システムにおい

ては、操作履歴がないため使いにくい。そこで疑似的な操作履歴を与え、ある程度のユーザ支援を実現する（6.1章）。提案したモデルの有効性を評価する（6.2章）。

2 問題の定式化

2.1 アイコンシステム実行環境

本システムで定義するアイコンは、ユーザの直観的な理解を促す絵シンボルと、複数の属性および機能をカプセル化したオブジェクトであり、それらは実世界に存在する”もの”のメタファとして設計される。

実在の”もの”を普遍的なオペレータあるいはオペランドに区別できないと同様に、アイコンもその両方の性質を備えている。それぞれのアイコンの振舞いは複数のものが可能であり、その中から適切なものが実行時に、他のアイコンと重ね合わせた時に動的に決定される。図1にアイコンの振舞いの例を示す。

このようなアイコンの振舞いを実現するために、各アイコンは交信可能なメッセージリストを保持する。メッセージリストには active と passive の2つがあり passive メッセージはそのアイコンが持つ機能に対応し、active メッセージは他のアイコンの機能実行のトリガとなる [1](図2)。

ユーザが2つのアイコンを重ね合わせると、各々のアイコンが持つ active メッセージを交換し、自身の passive メッセージと比較する。ここでマッチしたメッセージが2つのアイコン間で実行可能な機能であり、それが実行される。ここで2つ以上のメッセージがマッチした場合には、実行可能な機能が表示され、ユーザがそのうちの一つを選択することにより所望の機能を実行できる。図3に本システムの作業空間を示す。

本稿では、本システムにおけるアイコンの選択的表示法について述べる。

2.2 問題の定式化

ユーザが次に使用するアイコンを予測し、表示するユーザ支援環境を作成する。この問題を確率過程に基づき、時系列予測問題として定式化する。

$S = C_1, C_2, C_3, \dots$ はタスクの集合、

$I = I_1, I_2, I_3, \dots$ はアイコンの集合である。

状態空間はタスクに対応している。タスクの集合は2つのアイコンと関係している。時点 n でタスクが C_j のとき、 $x_t = j$ となる確率変数列を x_t とする。 $x_0 = i_0, x_1 = i_1, \dots, x_{t-1} = i_{t-1}$ であった時、 $x_t = j$ となる条件付確率が条件 $x_{t-1} = i_{t-1}$ だけに関係し、それ以前の経過に

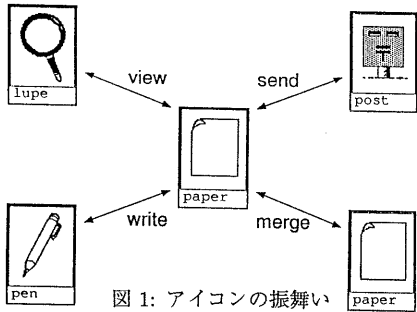


図 1: アイコンの振舞い

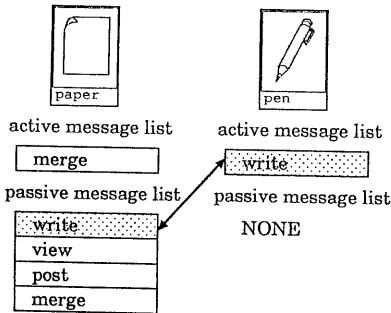


図 2: 双方向メッセージパッシング

無関係であると仮定する $P(x_t = i | x_{t-1} = j)$ を導入する。この確率はマルコフ連鎖の推移確率である。

また、 $x_0 = i_0, x_1 = i_1, \dots, x_{t-1} = i_{n-1}$ であった時、 $x_t = j$ となる条件付確率が条件 $x_{t-1} = i_{n-1}, x_{n-2} = i_{n-2}$ だけに関係し、それ以前の経過に無関係であると仮定する $P(x_{t+1} = i | x_t = j, x_{t-1} = k)$ を導入する。また、時刻 n の関わりと無関係の確率を $P(x_t = j)$ とおく。これらの確率に基づき、ユーザーが次の時系列で使うアイコンを予測する問題を解く。

3 操作の履歴の収集

操作の履歴は使用した操作(タスク)に依存するため、タスクに着目する

一時刻前の操作に基づく推移行列(タスク C_j の次に C_i を行なう確率) $p_{ij}^{(1)} = P(x_{t+1} = i | x_t = j)$ を一次の

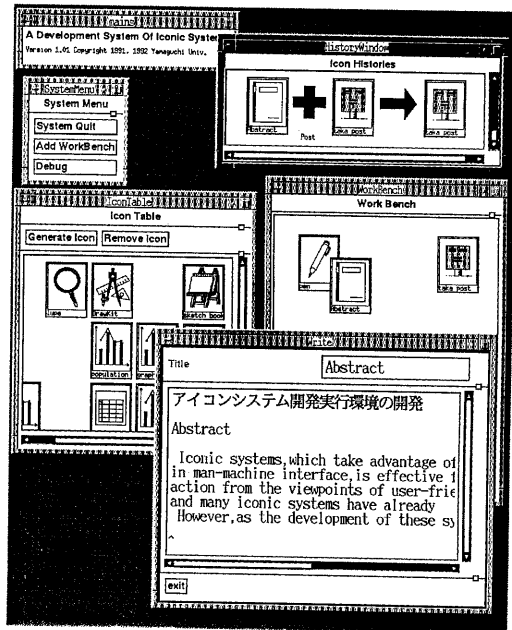


図 3: アイコン作業空間

推移行列とよび、二時刻前からの操作に基づく推移行列(タスク C_k, C_j の次に C_i を行なう確率) $p_{ijk}^{(2)} = P(x_{t+1} = i | x_t = j, x_{t-1} = k)$ を二次の推移行列とよび、 $p_i^{(0)} = P(x_{t+1} = i)$ を零次の推移行列とよぶ。但し、 $\sum_j p_{ij} = 1$ である。操作を始めた時の確率を $P(X_0 = i) = \pi_i^{(0)}$ とする。 $\pi_i^{(0)} = (\dots, \omega_{-1}^{(0)[t]}, \omega_0^{(0)[t]}, \omega_1^{(0)[t]}, \dots)$ は確率分布を行ベクトルで表している。操作履歴の収集は次のステップで行なわれる。一次の推移行列の計算のために、操作の回数を持つ行列 $Pnum_{ij}^{(1)} = Pnum(x_{t+1} = i | x_t = j)$ を用意する。前の時刻に C_j を操作し、次の時刻に C_i を操作した回数が $Pnum_{ij}^{(1)}$ である。一次の推移行列は $p_{ij}^{(1)} = Pnum_{ij}^{(1)} / \sum_j Pnum_{ij}^{(1)}$ で求められる。二次の推移行列も同様に $Pnum_{ijk}^{(2)}$ を計数することにより $p_{ijk}^{(2)} = Pnum_{ijk}^{(2)} / \sum_j Pnum_{ijk}^{(2)}$ で求められる。また、初期確率は $Pnum_{ij}^{(0)}$ を保持することにより、求められる。操作の全回数と推移がおきた回数を更新することにより、3つの推移行列および初期確率の更新は行なわれる。この3つの推移行列と初期確率をもとに、作業のながれに応じたアイコンの選択的表示を実現する。

4 確率分布による処理の識別

本システムでは、ユーザが次に使用するアイコンを予測し表示する。しかし、作業のながれに無関係に支援

をすることは適当でない。2つのアイコンが呈示されるのが適当な場合もあり、10数個のアイコンが呈示されることが適当な場合もある。この立場から、作業のながれに応じた支援を実現する。作業は、その正確から大きく分けて(1)バッチ依存の処理と(2)クラス依存の処理に分けられる。バッチ依存の処理はタスクの順番が決められている定型処理であり、クラス依存の処理は順番は関係ないが、常時複数のアイコンを使う可能性のある処理である。例えば、コンパイル、リンク、実行のように順序づけられた定型処理がバッチ処理にあたり、事務処理のように順序は関係ないが同じ道具を使う処理がクラス処理にあたる。

まず作業のながれに応じた支援を実現するために、クラス処理かバッチ処理かを推測し、次に、クラス処理、バッチ処理に応じたアイコンの選択的表示を実現する。

4.1 アイコン空間とタスク空間

選択的表示を行なうために、各々のアイコンが意味的な距離によって位置付けられているアイコン空間を考える。意味的な距離とは操作中のアイコンを中心として、重ね合わせ可能なアイコンやユーザ操作の履歴情報から得られる各アイコンの使用頻度等から決定されるアイコン間の重みである。一つの操作(タスク)は、2つのアイコンの重ね合わせにより実現される。そのため、アイコンの意味的な距離はタスク間の距離に依存する。そこで、各タスクが意味的な距離によって位置付けられるタスク空間を考える。タスク間の意味的な距離はユーザ操作の履歴情報が反映していなければならない。

タスク間の距離を一次確率と2次確率の総和と定義する。

$$Distance_{ij} = \alpha * p_{ij}^{(1)} + \beta * \sum_{k=0}^n p_{ijk}^{(2)}$$

$Distance_{ij}$ はjの次にiがくる依存関係を定量的に示したものである。 α, β は依存関係の重要度に応じて変更が可能である。但し $\alpha \gg \beta$ である。ここでは、 $\alpha = 1, \beta = 0$ の場合値の範囲は[0,1]である。

タスク空間とアイコン空間の関係を図4に示した。タスク空間は現在操作中のタスクを中心として意味的な距離に基づいて位置付けられている。タスク空間はアイコン空間とタスクを操作するアイコンを介してつながっている。タスク空間を観測し必要なタスクを予測し、タスクを操作するアイコンを選択的に表示する。そのため、今後の議論ではタスク空間を観測する。

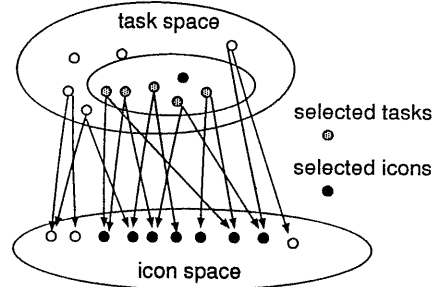


図4: アイコン空間とタスク空間

4.2 確率分布の観測

タスク空間における距離からタスクを配置し、確率分布関数を観測する。

n 個のタスクが存在する時に、 $\pi_t^{(0)} = (\dots, \omega_{-1}^{(0)[t]}, \omega_0^{(0)[t]}, \omega_1^{(0)[t]}, \dots)$ は零時確率である。但し、 $\omega_k^{(0)[t]}$ はタスク C_k の t 時刻の零次確率である。 $\pi_t^{(1)} = (\dots, \omega_{-1}^{(1)[t]}, \omega_0^{(1)[t]}, \omega_1^{(1)[t]}, \dots)$ は一次確率を元にした時刻の確率分布である。但し、 $\omega_k^{(1)[t]}$ はタスク C_k の t 時刻の確率である。 $\pi_t^{(2)} = (\dots, \omega_{-1}^{(2)[t]}, \omega_0^{(2)[t]}, \omega_1^{(2)[t]}, \dots)$ は二次確率を元にした時刻の確率分布である。但し、 $\omega_k^{(2)[t]}$ はタスク C_k の t 時刻の零次確率である。

- $\pi_t^{(1)} = (\dots, \omega_i^{(1)[t]}, \omega_j^{(1)[t]}, \omega_k^{(1)[t]}, \dots)$
- $\pi_t^{(2)} = (\dots, \omega_i^{(2)[t]}, \omega_j^{(2)[t]}, \omega_k^{(2)[t]}, \dots)$
- $\pi_t^{(0)} = (\dots, \omega_i^{(0)[t]}, \omega_j^{(0)[t]}, \omega_k^{(0)[t]}, \dots)$

但し

$$Distance_{ij} > Distance_{ik}$$

が成立する。

$T(\omega_j^{(0)[t]})$ は $\omega_j^{(0)[t]}$ のtask番号を返す。すなわち、 $\omega_j^{(0)[t]}$ は $C_{T(\omega_j^{(0)[t]})}$ に対応している。

$\pi_t^{(1)}, \pi_t^{(2)}, \pi_t^{(0)}$ を正規分布と仮定した時の平均ベクトル $\mu^{(1)[t]}, \mu^{(2)[t]}, \mu^{(0)[t]}$ と分散値 $\sigma^{2(1)[t]}, \sigma^{2(2)[t]}, \sigma^{2(0)[t]}$ を求める。

$\pi_t^{(2)}, \pi_t^{(1)}, \pi_t^{(0)}$ から各処理を識別する。

μ, σ の推定値はもし $\omega_0 = \omega_n = 0$ ならば次のようにして求められる。

$$\mu = \frac{n}{\sum_i \omega_i b_i}$$

$$\sigma^2 = \frac{n}{\omega_i(b_i - \mu^2)}$$

$b_i = (i + t)/2$ とする。

但し $\pi_0 + \pi_k$ があまり小さくない場合もしくは、 $\pi_0 = f_\pi = 0$ でも k が小さくて、 σ に比べて h の値が大きいと考えられる場合は、 μ, σ を求めるために、最尤法を用いる。

4.3 処理の識別

タスク空間において、 t 時刻から $t+1$ 時刻における $\pi_{t+1}^{(0)}, \pi_{t+1}^{(1)}, \pi_{t+1}^{(2)}$ を観測し、処理を識別する (局所的識別)。その後、 t 時刻から $t+\alpha$ 時刻、 $\pi_{t+1}^{(0)}, \pi_{t+1}^{(1)}, \pi_{t+1}^{(2)}$ を観測し、処理の確信度を計算する (大局的識別)。確信度が一定値より大きければ局所的観測に基づく識別を処理の予測値とする。

4.3.1 局所的観測に基づく処理の識別

タスク空間において、 t 時刻から $t+1$ 時刻における $\pi_{t+1}^{(0)}, \pi_{t+1}^{(1)}, \pi_{t+1}^{(2)}$ を観測し、処理を識別する。処理には、クラス処理とバッチ処理があり、さらにそれぞれの処理は、始点、終点、混在点に分類される。各処理で観測される $\pi_{t+1}^{(0)}, \pi_{t+1}^{(1)}, \pi_{t+1}^{(2)}$ の特徴を述べる。

クラス処理の特徴は、零次確率、1次確率の分散値が高い。2次確率の分散値はさらに高い。逆方向の確率も高い。バッチ処理の特徴は、一次確率、2次確率の分散値が極端に低い。それぞれの処理に属性を割り当てる。

- バッチ処理の中点 (B_{mid}) $\sigma^{2(0)[t]} < K1; \sigma^{2(1)[t]} < K1; \sigma^{2(2)[t]} < K1$
- バッチ処理の分岐点 (B_{mix}) $K1 < \sigma^{2(0)[t]} < K1 * 2; K1 < \sigma^{2(1)[t]} < K1 * 2; \sigma^{2(2)[t]} < K1$
- バッチ処理の始点 (B_{start}) B_{mix} に同じ、一時刻前の処理と異なる。
- バッチ処理の終点 (B_{end}) $\sigma^{2(0)[t]} < K2; \sigma^{2(1)[t]} < K2; \sigma^{2(2)[t]} < 2 * K2$
- クラス処理の中点 (C_{mid}) $\sigma^{2(0)[t]} > K2; \sigma^{2(1)[t]} > K2; \sigma^{2(2)[t]} > 2 * K2$
- クラス処理の混在点 (C_{mix}) $\sigma^{2(0)[t]} > K2; \sigma^{2(1)[t]} > K2; \sigma^{2(2)[t]} > K2$
- クラス処理とバッチ処理の混在点 (CB_{mix}) $\sigma^{2(0)[t]} > K2; \sigma^{2(1)[t]} > K2; \sigma^{2(2)[t]} < K1$
- 識別不可能点 ($B_{complex}$) 上の制約以外。

これらの定性的分類に基づいて処理を識別する。識別関数 $f(\mu^{(0)[t]}, \mu^{(1)[t]}, \mu^{(2)[t]}, \sigma^{2(0)[t]}, \sigma^{2(1)[t]}, \sigma^{2(2)[t]})$ によって処理の属性値 ($B_{mid}, B_{start}, B_{mix}$ など) の属性値と呈示するタスク集合を出力する。

4.3.2 大局的観測に基づく処理の識別

t 時刻から $t+\alpha$ 時刻 $\pi_{t+1}^{(0)}, \pi_{t+1}^{(1)}, \pi_{t+1}^{(2)}$ を観測することにより、局所的観測に基づく識別で予測された処理の確信度を計算する。確信度は予測の矛盾から計算する。確信度が小さい時は、他の処理の確率空間を観測する。具体的には、 $f(\mu^{(0)[t]}, \mu^{(1)[t]}, \mu^{(2)[t]}, \sigma^{2(0)[t]}, \sigma^{2(1)[t]}, \sigma^{2(2)[t]})$ によって処理の属性値 $\tilde{Attr}_{t+1}(C^{[t]})$ と予測されるタスク集合 \tilde{Q}_{t+1} を出力する。処理の属性値は $B_{mid}, B_{start}, B_{mix}$ であり、アイコン集合 $C_1^{[t]}, C_2^{[t]}, \dots, C_n^{[t]} \in Q_t$ である。

確信度関数 $Truth(\tilde{Attr}_{t+1} | \tilde{Attr}_t)$ はは次のように定義される。

$$\begin{aligned} Truth(B|B) &= 1.0; Truth(C|C) = 1.0 \\ Truth(C|B) &= 0.7; Truth(B|C) = 0.7 \\ Truth(C_{mix}|B) &= 0.8; Truth(B_{mix}|C) = 0.8 \\ Truth(BC_{mix}|C) &= 1.0 Truth(BC_{mix}|B) = 1.0 \\ Truth(BC_{complex}|C) &= 1.0 \\ Truth(BC_{complex}|B) &= 1.0 \end{aligned}$$

但し

$$\begin{aligned} B &= B_{mid} \vee B_{mix} \vee B_{start} \vee B_{end} \\ C &= C_{mid} \vee C_{mix} \vee C_{start} \vee C_{end} \end{aligned}$$

バッチ処理における確信度の計算は次式で求める。

$$\begin{aligned} TRUTH(\tilde{Attr}_t(C^{[t]}) = B) \\ = \prod_{j=t}^{t+\alpha} Truth(\tilde{Attr}_t(\tilde{C}_i^{[j]})|B) \end{aligned}$$

但し $\tilde{Attr}_t C^{[t+\alpha]} = B_{end}$ である。

クラス処理における確信度の計算は次式で求める。

$$\begin{aligned} TRUTH(\tilde{Attr}_t(C^{[t]}) = C) \\ = \prod_{i=1}^n Truth(\tilde{Attr}_t(\tilde{C}_i^{[t]})|C) \end{aligned}$$

確信度は $0 \leq TRUTH \leq 1$ であり、1に近いほど確信度が高く、0に近いほど確信度が低い。

4.4 処理履歴の収集

本節では、操作履歴の収集とともに、処理履歴を収集し、より確からしいユーザ支援を実現する。

処理を $D_1, D_2, D_3, \dots, D_n$ と記述する。一時刻前の処理に基づく推移行列 (処理 D_j の次に D_i を行なう確率) $pd_{ij}^{(1)} = P_D(x_{t+1} = i | x_t = j)$ を一次の推移行列とよび、二時刻前からの処理に基づく推移行列 (処理 D_k, D_j の次に D_i を行なう確率) $pd_{ijk}^{(2)} = P_D(x_{t+1} = i | x_t = j, x_{t-1} = k)$ を二次の推移行列とよび、 $p_i^{(0)} = P_D(x_{t+1} = i)$ を零次の推移行列 (処理の履歴に関係なく処理 D_i を行なう確率) とよぶ。但し、 $\sum_j p_{ij} = 1$ である。操作を始めた時の確率を $P(X_0 = i) = \pi_i^{(0)}$ とする。 $\pi_i^{(0)} = (\dots, \omega_{-1}^{(n)[t]}, \omega_0^{(n)[t]}, \omega_1^{(n)[t]}, \dots)$ は確率分布を行ベクトルで表している。操作履歴の収集は次のステップで行なわれる。一次の推移行列の計算のために、操作の回数を保持する行列 $P_{Dnum}_{ij}^{(1)} = P_{Dnum}(x_{t+1} = i | x_t = j)$ を用意する。前の時刻に D_j を操作し、次の時刻に D_i を操作した回数が $P_{Dnum}_{ij}^{(1)}$ である。一次の推移行列は $pd_{ij} = P_{Dnum}_{ij} / \sum_j P_{Dnum}_{ij}$ で求められる。二次の推移行列も同様に P_{Dnum}_{ijk} を計数することにより $Pd_{ijk}^{(2)} = P_{Dnum}_{ijk} / \sum_j P_{Dnum}_{ijk}$ で求められる。また、初期確率は $P_{Dnum}_{ij}^{(0)}$ から計算される。操作の全回数と推移がおきた回数を更新することにより、3つの推移行列および初期確率の更新は行なわれる。この3つの推移行列と初期確率をもとに、作業のながれに応じたアイコンの選択的表示を実現する。

5 処理に応じたアイコン集合の表示

$f(\mu^{(0)[t]}, \mu^{(1)[t]}, \mu^{(2)[t]}, \sigma^{(0)[t]}, \sigma^{(1)[t]}, \sigma^{(2)[t]})$ によって処理の属性値 $\hat{Attri}(C^{[t]})$ 予測されるタスク集合 \hat{Q}_{t+1} を出力する。

5.1 クラス処理

$$\mu^{(1)} - K \cdot \sigma^{(1)} < X < \mu^{(1)} + K \cdot \sigma^{(1)}$$

$$TRUTH(\hat{Attri}(C^{[t]} = C)) > J$$

を満足するタスク集合、 \hat{Q}_{t+1} を選択する。但し K はスコープの大きさを制御する変数であり、 J は確信度を制御する変数である。タスク集合、 \hat{Q}_{t+1} を操作するアイコンを表示する。

クラス処理における処理の終点は一般にわかりにくい。処理別の確率分布を求め、ベイズ識別規則により予測する。クラス処理においては、他の処理を行なう確率を計算する。クラス処理においては、クラスに属するタスクすべてに関係するアイコンをすべて表示する。前の時系列でクラス処理の場合は、処理空間の確率分布を観測し、その処理候補を表示する。

5.2 バッチ処理

$$\alpha = \max[\mu^{(1)[t]} - K \cdot \sigma^{(1)[t]}, \mu^{(1)[t]} - K \cdot \sigma^{(2)[t]}]$$

$$\beta = \min[\mu^{(1)[t]} + K \cdot \sigma^{(1)[t]}, \mu^{(1)[t]} + K \cdot \sigma^{(1)[t]}]$$

$$\alpha < X < \beta$$

を満足するタスクを選択する。但し K はスコープの大きさを制御する変数である。また前の時系列における属性がクラスであるかもしくは、違う処理系列である時のみ

$$\alpha = \mu^{(1)[t]} - K \cdot \sigma^{(1)[t]}$$

$$\beta = \mu^{(1)[t]} + K \cdot \sigma^{(1)[t]}$$

$$\alpha < X < \beta$$

を満足するタスクを選択する。この条件を満足するタスクの予測を $\hat{C}_i^{[t]}$ ($t < \alpha$ について $\hat{Attri}(\hat{C}_i^{[t]}) \neq B_{end}$ $t = \alpha$ の時 $\hat{Attri}(\hat{C}_i^{[t]}) = B_{end}$) とすると、バッチ処理において予測されるタスク集合は $\hat{Q}_{t+1} = \{\hat{C}_i^{[t]} | (t = t, t+1, \dots, t+\alpha)\}$ となる。タスク集合、 \hat{Q}_{t+1} を操作するアイコンを表示する。

5.3 アイコンの選択的表示システムの動作

選択的表示のための簡単なシステム構成を図6に示す。まず、すべてのアイコンとユーザの操作の履歴情報を用いて、アイコン空間を決定する。アイコン空間が決定した後、確率分布の局所的観測により処理の識別をする。確率分布の観測に基づき局所的観測による推測を裏付ける確信度を計算し、必要なアイコン集合を選択する。ユーザは選択的に表示されたアイコンを用いて作業を行なう。システムはユーザの操作を履歴情報として保存する。操作が行なわれるたびに、アイコン空間を再定義する。このような一連の動作の繰り返しによって、ユーザ特有のアイコン空間が定義される。このアイコン空間はユーザ特有の作業に対応することができる。

6 システムの評価

6.1 初期システムのユーザ支援

初期システムにおいては、操作履歴がないため使いにくい。そこで疑似的な操作履歴を与え、ある程度のユーザ支援を実現する。

n 個のタスク中に a 個のバッチ処理 $A_i (i = 1, 2, \dots, a)$ と b 個のクラス処理 $B_i (i = 1, 2, \dots, b)$ が存在する。バツ

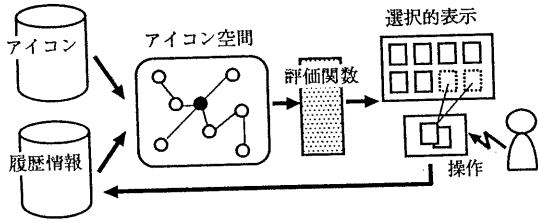


図 5: システムの動作

チ処理中には $m_i (i = 1, 2, \dots, a)$ 個のタスクが含まれる。このタスク集合を M_i とする。クラス処理中には $k_i (i = 1, 2, \dots, b)$ 個のタスクが含まれる。このクラス集合を K_i とする。バッチ処理中のタスクは $f_j(i) (i = 1, 2, \dots, m_i) (j = 1, 2, \dots, a) \in M_i$ で割り当てられる。クラス処理中のタスクは $g_j(i) (i = 1, 2, \dots, k_i) (j = 1, 2, \dots, b) \in K_i$ で割り当てられる。

$$P2(x_{t+1} = \beta | x_t = \alpha) = 1$$

但し, $\beta = f_j(i+1), \alpha = f_j(i)$ とする。

$$P1(x_{t+1} = \beta | x_t = \alpha) = \frac{1}{k_j}$$

但し, $\beta = f_k(j), \alpha = f_k(i)$ とする。

$$P3(x_{t+1} = \beta | x_t = \alpha) = \frac{1}{a+b}$$

但し,

$$\beta = f_j(1), \alpha = f_j(m_j)$$

とする。

$$A1 = \sum_{j=1}^a \sum_{i=1}^{m_j-1} P2(x_{t+1} = f_j(i+1) | x_t = f_j(i)) \delta_{\alpha\beta}$$

但し,

$$\delta_{\alpha\beta} \begin{cases} \text{if}((\alpha = f_j(i)) \wedge (\beta = f_j(i+1))) \text{ then } 1 \\ \text{else } 0 \end{cases}$$

$$B1 = \sum_{k=1}^b \sum_{i=1}^{k_i} \sum_{j=1}^{k_i} P1(x_{t+1} = g_k(j) | x_t = g_k(i)) \delta_{\alpha\beta}$$

但し,

$$\delta_{\alpha\beta} \begin{cases} \text{if}((\alpha = g_k(j)) \wedge (\beta = g_k(i))) \text{ then } 1 \\ \text{else } 0 \end{cases}$$

$$C1 = \sum_{j=1}^a P3(x_{t+1} = f_j(1) | x_t = f_j(m_j)) \delta_{\alpha\beta}$$

但し,

$$\delta_{\alpha\beta} \begin{cases} \text{if}((\alpha = f_j(1)) \wedge (\beta = f_j(m_j))) \text{ then } 1 \\ \text{else } 0 \end{cases}$$

$$A2 = \sum_{j=1}^a \sum_{i=1}^{m_j-1} P2(x_{t+1} = f_j(i+1) | x_t = f_j(i)) \delta_{\alpha}$$

但し,

$$\delta_{\alpha} \begin{cases} \text{if}(\alpha = f_j(i)) \text{ then } 1 \\ \text{else } 0 \end{cases}$$

$$B2 = \sum_{k=1}^b \sum_{i=1}^{k_i} \sum_{j=1}^{k_i} P1(x_{t+1} = g_k(j) | x_t = g_k(i)) \delta_{\alpha}$$

但し,

$$\delta_{\alpha} \begin{cases} \text{if}(\beta = g_k(i)) \text{ then } 1 \\ \text{else } 0 \end{cases}$$

$$C2 = \sum_{j=1}^a P3(x_{t+1} = f_j(1) | x_t = f_j(m_j)) \delta_{\alpha}$$

但し,

$$\delta_{\alpha} \begin{cases} \text{if}(\beta = f_j(m_j)) \text{ then } 1 \\ \text{else } 0 \end{cases}$$

$$P(x_{t+1} = \beta | x_t = \alpha) = \frac{A1 + B1 + C1}{A2 + B2 + C2}$$

同様に、2次確率、零次確率も近似的に計算する。この計算により、疑似的な操作履歴が生成される。

6.2 システムの評価

アイコン呈示環境を模擬して、本システムの評価を行なう。クラス処理 A, B とバッチ処理 C, D がある。

- A にはタスクが A1, A2, A3, A4

- Bにはタスクが B1, B2, B3, B4
- Cにはタスクが C1, C2, C3, C4
- Dにはタスクが D2, D2, D3, D4

が割り当てられている。まず A, B, C, D の各処理は正規分布に従うと仮定する。バッチ処理はたえず C1, C2, C3, C4 の順序で行ない。クラス処理においては B1, B2, B3, B4 を当確率で発生させ、クラス処理から抜ける確率は $\mu - K2 > X; X > \mu + K2$ から決定する。バッチ処理の終点においては C4, D4 については A, B, C, D の各処理は正規分布に従うため、処理空間の確率分布を計算する。上のモデルにおいて $A2=B2$ 、バッチ処理の混在点であり $C2=D2$ はクラス処理の混在点であり、 $A3=C3$ はバッチとクラス処理の混在点である。

実験では、この評価モデルに対してアイコンの選択的表示を行ない、選択的表示の中に目的とするアイコンの存在、および処理の識別、処理とタスクの対応を評価した。

クラス処理から違う処理への予測点を除くと、80%近い確率でユーザ支援が可能になった。第一の処理の候補の他に次に確信度の高い第二候補、第三候補を表示しておくことにより、20%の誤認識を補なえるアイコンの選択的表示が可能になると考えられる。ユーザインターフェースとしては、予測するアイコン群の他に、識別された処理郡を表示しておくことにより、ユーザの過去の履歴にとらわれない作業に対処することが可能になる。また、本研究はタスク空間を基盤として予測を定義しているため、コマンドの予測などにも適用が可能と考えられる。

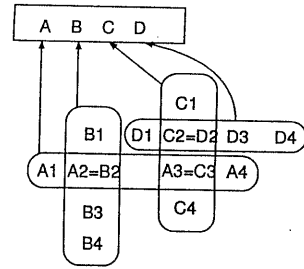
7 まとめ

アイコンにシステム定義の階層関係を与えずに、操作履歴を用いて次に使用するアイコン郡を予測し、選択的に表示するユーザ支援環境を作成した。この問題を確率過程に基づき、時系列予測問題として定式化した。バッチ依存の処理とクラス依存の処理について、提案したモデルが有効であることが示された。

本研究の一部は科学研究費(重点領域(2)、課題番号05220215)の援助によった。

参考文献

- [1] M.Hirakawa, M.Tanaka and T.Ichikawa, "An Iconic Programming System, HI-VISUAL": ,



Batch process A1-A4
B1-B4
Class process C1-C4
D1-D4
B_mix A2=B2
BC_mix A3=C3 B_start A1,B1
C_mix C2=D2 B_end A4,C4

図 6: 評価モデル

IEEE trans. on Software Engineering-16, 10, pp.1178-1184, (1990)

- [2] Shu.N.C, "Visual Programming Languages:A Dimensional Analysis": , Proc of the International Symposium on New Directions in Computing, , pp.326-334, (1985)
- [3] John J.Darragh,Ian H. Witten and Mark L. James, "The reactive keyboard: A predictive typing aid." : , IEEE computer, 23, 11, (pp.41-49) 1990
- [4] Brad A.Myers, "Demonstrational interfaces: A step beyond direct manipulation." : , IEEE Computer, 25, 8, (pp.61-73) 1992
- [5] 山口 西村 田中, "アイコンシステムのためのオブジェクト作成ツールの開発": , 情処研報, 93-PRG-10, 11, (1993)
- [6] Rabiner L.R, "A Tutorial on Hidden Markov Models and selected applications in speech recognition": , Proc.IEEE, 77, 2, (pp.257-286) 1989
- [7] Geman S. and Geman D., "Stochastic relaxation,Gibbs distributions, and the Bayesian restoration of images": , IEEE Trans.PAMI, 6, , (pp.721-741) 1984