

Probabilidades Intervalares em Modelos Ocultos de Markov

André V. Santos¹, Graçaliz P. Dimuro¹,

Luciano V. Barboza^{2,3}, Renata H.S. Reiser¹, Antônio C.R. Costa^{1,4},

¹ Escola de Informática, Universidade Católica de Pelotas,

² Escola de Engenharia e Arquitetura, Universidade Católica de Pelotas,

³ Centro Federal de Educação Tecnológica de Pelotas,
96010-000, Pelotas, RS

⁴ PPGC, PGIE, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS

E-mail: { pablogri, liz, luciano, reiser, rocha }@ucpel.tche.br

Marcília A. Campos

CIN/UFPE - Universidade Federal de Pernambuco

Av. Prof. Moraes Rego, 1235,

50670-901, Recife, PE

E-mail: mac@cin.ufpe.br

1 Introdução

No desenvolvimento de aplicações numéricas em Ciência e Tecnologia surgem erros de computação, os quais originam-se, primordialmente, da impossibilidade de se modelar e representar grandezas contínuas em uma entidade de natureza finitária, como o computador. O sistema de ponto flutuante da máquina não é capaz de representar exatamente os números reais, nem os resultados de operações com esses números. Além disso, como um sistema algébrico, suas características e propriedades algébricas são extremamente pobres quando comparadas com as do sistema de números reais [6].

A matemática intervalar [11] é uma teoria matemática que teve por objetivo inicial responder a questão da exatidão e da eficiência que aparece na prática da computação científica. Desde então, a utilização de técnicas intervalares tem sido uma alternativa para alcançar limites garantidos para resultados de computações, através do controle rigoroso e automático da propagação dos erros dos dados e parâmetros iniciais ao longo do processo computacional, assim como dos erros de arredondamento e truncamento. Algoritmos intervalares, em contraste com os algoritmos pontuais, computam um intervalo como solução, com a garantia de que a resposta exata pertence a este intervalo.

Atualmente, a Matemática Intervalar está sendo utilizada também em várias áreas que lidam com informações incertas (veja, p.ex., [5, 7]).

A probabilidade intervalar foi proposta por Campos [1] como uma metodologia para estender a probabilidade real de forma a possibilitar seu cálculo

automático através do uso de intervalos¹. Esta probabilidade intervalar tem uma conceituação similar à probabilidade usual, sendo então definida como uma função satisfazendo a um conjunto de axiomas. Esta função assume valores no espaço dos intervalos [11], e, portanto, é fundamentada na análise intervalar, justificando o nome *probabilidade intervalar*. Adicionalmente, as operações aritméticas são realizadas segundo a aritmética de exatidão máxima [9]. Esta probabilidade intervalar controla os erros decorrentes da representação dos reais por números de ponto flutuante [6].

Uma aplicação de interesse da probabilidade intervalar é no cálculo das probabilidades de transição de uma cadeia de Markov. Em [3], introduziu-se o conceito de cadeias de Markov intervalares e apresentou-se uma aplicação no Maple [10]. Outra abordagem para cadeias de Markov com intervalos pode ser encontrada em [8].

Os modelos ocultos de Markov [12, 13], surgiram originalmente no domínio de reconhecimento do discurso, e, atualmente têm sido empregados em áreas da Inteligência Artificial, Processamento de Linguagens, Bioinformática e Visão Computacional, assim com em reconhecimento de manuscritos, de formas, gestos e expressões faciais (veja em <http://www-sig.enst.fr/~cappe>).

Este trabalho apresenta uma aplicação de probabilidades intervalares em modelos ocultos de Markov, com o desenvolvimento de algoritmos intervalares para os problemas relacionados a esses modelos (*Problema da Avaliação, Problema da Decodi-*

¹Outras abordagens para probabilidades intervalares podem ser encontradas em [15].

ficção e *Estimação de Parâmetros*). Foram desenvolvidas versões intervalares para os algoritmos *Forward*, *Backward*, *Viterbi* e *Baum Welch*. As implementações foram realizadas utilizando-se a toolbox Inlab para a Matemática Intervalar, no ambiente Matlab. Exemplos de aplicações são apresentados, mostrando-se a validade dos algoritmos desenvolvidos.

O artigo está organizado conforme descrito a seguir. A Seção 2 apresenta os modelos ocultos de Markov e os problemas relacionados a estes modelos. Os principais conceitos da Matemática Intervalar são apresentados na Seção 3. Na Seção 2, introduz o conceitos de Modelos Ocultos de Markov Intervalares e as versões intervalares dos algoritmos *Forward*, *Backward*, *Viterbi* e *Baum Welch*. Exemplos são discutidos na Seção 5. A Seção 6 traz as conclusões do trabalho.

2 Modelos Ocultos de Markov

Um Modelo Oculto de Markov [12, 13] é uma variante das Cadeias de Markov, onde, além da função de distribuição de probabilidades associadas aos estados, existe uma função de distribuição de probabilidades para as observações que podem ser realizadas em cada estado. Consiste de um processo duplamente estocástico composto por um processo oculto (não observável), mas que se manifesta através de um outro processo estocástico que produz a seqüência de símbolos observados. Os dois tipos de parâmetros a que um Modelo Oculto de Markov está associado são: *probabilidades de emissão dos símbolos e probabilidades de transição de estados*.

Um exemplo clássico encontrado na literatura [12, 13], é o problema de “bolas e urnas”. Existem três urnas atrás de uma cortina não transparente, e dentro destas urnas tem-se várias bolas com cores distintas. Uma urna é lecionada ao acaso e uma bola é retirada de dentro dela e mostrada aos expectadores, que, apesar de ficarem sabendo a cor dessa bola, desconhecem de que urna ela foi retirada. Diz-se que as “urnas” são os estados ocultos do modelo e as “cores possíveis de bolas” são as observações que podem ser realizadas em cada estado.

Definition 1 *Um Modelo Oculto de Markov é definido como*

$$\lambda = (A, E, Q, \varepsilon, \pi),$$

onde A é uma matriz de números reais não negativos indexada por $Q \times Q$, denominada de matriz de transição de estados, E é uma matriz de números reais não negativos indexada por $\varepsilon \times Q$ que contém as probabilidades de emissão de símbolos em cada estado, Q é o conjunto de estados do modelo, π é o vetor da distribuição inicial de probabilidades desses estados e ε é o conjunto de símbolos do alfabeto.

Os problemas relacionados a esses modelos são os seguintes:

Problema da Avaliação: é o problema de calcular a probabilidade de uma seqüência de observações para um dado modelo. Para solucionar este problema são utilizados os algoritmos de programação dinâmica *Forward-Backward*.

Problema da Decodificação: é o problema de descobrir a mais provável seqüência de estados (seqüência ótima) capaz de gerar uma dada seqüência de observações, o que pode ser realizado por meio do *Algoritmo de Viterbi*.

Estimação de Parâmetros: é o problema de ajustar os parâmetros de um dado modelo de modo que a probabilidade de uma seqüência de observações seja maximizada. Isto pode ser realizado através do *Algoritmo de Baum Welch*.

3 A Matemática Intervalar

A Matemática Intervalar considera um conjunto de métodos para manipulação de intervalos numéricos que aproximam dados incertos. Estes métodos baseiam-se na definição da aritmética intervalar e do produto escalar ótimo [9]. O princípio da máxima exatidão garante (através dos arredondamentos direcionados) o controle automático dos erros de resultado de computações numéricas.

Um *intervalo real* X é um conjunto não vazio de números reais \mathbb{R} ,

$$X = [x_1, x_2] = \{x \in \mathbb{R} \mid x_1 \leq x \leq x_2\},$$

onde x_1 is o extremo inferior (ou *ínfimo*) e x_2 é o extremo superior (ou *supremo*). O conjunto de intervalos reais é denotado por \mathbb{IR} .

Um intervalo real $X = [x_1, x_2] \in \mathbb{IR}$ pode não ser representável em uma máquina se x_1 and x_2 não são números do sistema de ponto flutuante da máquina. Para obter um intervalo arredondado \tilde{X} tal que $X \subseteq \tilde{X}$ (i.e., \tilde{X} é uma aproximação de X), x_1 e x_2 devem ser arredondados “por falta” e “por excesso”, respectivamente, o que se denomina de *arredondamento direcionado*.

O *ponto médio*, o *diâmetro* e o *raio* de um intervalo X são definidos, respectivamente, como $mid(X) = \check{X} = \frac{1}{2}(x_1 + x_2)$, $diam(X) = x_2 - x_1$ and $rad(X) = \frac{1}{2}diam(X)$. X pode também ser denotado por $X = \langle mid(X), rad(X) \rangle$.

As operações aritméticas intervalares são definidas de forma que o intervalo resultado engloba todos os possíveis resultados reais [11], o que garante a confiabilidade dos resultados intervalares. As operações aritméticas são então definidas como

$$X * Y = \{x * y \mid x \in X, y \in Y\},$$

para $*$ $\in \{+, -, \times, \div\}$, e, para $X = [x_1, x_2], Y = [y_1, y_2] \in \mathbb{IR}$, elas são explicitamente calculadas como:

$$\begin{aligned} X + Y &= [x_1 + y_1, x_2 + y_2] \\ X - Y &= [x_1 - y_2, x_2 - y_1] \\ X \times Y &= [\min \rho, \max \rho] \\ &\quad \text{with } \rho = \{x_1y_1, x_1y_2, x_2y_1, x_2y_2\} \\ X \div Y &= X \times [y_2^{-1}, y_1^{-1}], \text{ if } 0 \notin Y. \end{aligned}$$

Para a adição e multiplicação, as propriedades da associatividade e comutatividade são válidas. Entretanto, exceto em casos especiais, a propriedade da distributividade não é válida, o que frequentemente causa uma superestimação.

A biblioteca IntLab [14] do ambiente Matlab, para a Matemática Intervalar, utiliza a rotina “setround” para controlar o modo de arredondamento. Na utilização desta biblioteca, os intervalos podem ser armazenados tanto como na forma de “ínfimo-supremo” como na de “ponto médio – raio”. A IntLab possibilita que as operações intervalares básicas sejam executadas sobre escalares intervalares reais e complexos, assim como vetores e matrizes. As funções intervalares padrões, como, por exemplo, as trigonométricas, exponencial, logarítmica etc., também estão disponíveis.

A probabilidade intervalar [1] é realizada através de uma peculiar composição de funções envolvendo extensões intervalares de funções reais. A um evento X é associada uma probabilidade $Pr(X) = p$, que, por sua vez, é associada a um intervalo que a contenha. Em [2] foram calculadas probabilidades intervalares para as variáveis aleatórias discretas Bernoulli, Binomial, Poisson, Poisson Truncada, Geométrica, Hipergeométrica e Pascal.

Do ponto de vista da implementação de probabilidades intervalares utilizando-se a biblioteca IntLab², se p tem valor incerto ou não é representável em um específico sistema de ponto flutuante, então será substituído pelo menor intervalo de máquina que o contém propriamente. Para tanto, implementou-se a rotina *probint* que, dados uma probabilidade pontual e sua incerteza, determina a “menor”³ probabilidade intervalar.

4 Modelos Ocultos de Markov Intervalares

Nesta seção introduz-se o conceito de Modelo Oculto de Markov Intervalar, e as versões intervalares propostas para os algoritmos *Forward*, *Backward*, *Viterbi* and *Baum Welch*⁴.

²Para implementações no Maple, veja [3].

³Menor probabilidade intervalar é aquela que apresenta o menor diâmetro.

⁴Por falta de espaço, os algoritmos de *Viterbi Intervalar* e *Baum Welch Intervalar* não serão detalhados.

Definition 2 Um Modelo Oculto de Markov Intervalar é aquele em que se consideram probabilidades intervalares.

Os Algoritmos *Forward* e *Backward* são usados para determinar a probabilidade intervalar de uma seqüência de observações $S^k = \{s^1, \dots, s^k\}$. Cada símbolo de s é observado em um tempo $t = k$. As probabilidades intervalares intermediárias são calculadas pelas variáveis α (*Algoritmo Forward*) e β (*Algoritmo Backward*), que fazem os cálculos em sentidos opostos, conforme mostra-se a seguir.

No caso do *Algoritmo Forward Intervalar*, cujo núcleo principal pode ser visualizado na figura 1, o cálculo das probabilidades intervalares iniciais é feito para todos os estados em $t = 1$. A variável α é uma matriz indexada por $S \times Q$ e calculada como $\alpha(1, i) = \pi_i e_i(s_1)$, com $1 \leq i \leq N$, onde N é o número de estados do modelo. Para cada $t > 1$, uma probabilidade intervalar parcial é calculada em cada estado como $\alpha(t + 1, j) = \left[\sum_{i=1}^N \alpha(t, i) a_{i,j} \right] e_j(s_{t+1})$. A soma de todas as probabilidades intervalares parciais resulta na probabilidade intervalar da seqüência de observações considerada, sendo $Pr(S^k) = \sum_{j=1}^N \alpha(T, j)$, onde T é o tamanho da seqüência de observações.

```
for i = 2:T
    for j = 1:Q
        mult=inf-sup(0,0);
        for
            k = 1:Q
                mult = (alf(i-1,k)*trans_int(k,j))
                    +mult;
            end
            obs_ti = obs_int(i);
            alf(i,j) = mult * emissao_int(obs_ti,j);
        end
    end
end prob_fwd = inf-sup(0,0); for j = 1:Q
    prob_fwd = prob_fwd + alf(T,j);
end
```

Figura 1: *Algoritmo Forward Intervalar*

Para o *Algoritmo Backward Intervalar*, mostrado na Figura 2, os cálculos das probabilidades intervalares são realizados considerando:

Inicialização: $\beta(T, i) = 1$, onde $1 \leq i \leq N$

Recursão: $\beta(t, i) = \sum_{j=1}^N \beta(t + 1, j) a_{i,j} e_j(s_{t+1})$,

onde $t = T - 1, T - 2, \dots, 1, 1 \leq i \leq N$

Finalização: $Pr(S^k) = \sum_{i=1}^N \beta(1, i) e_i(s_1) \pi_i$

O *Algoritmo de Viterbi Intervalar* é utilizado para determinar qual o caminho (seqüência de estados) mais provável na geração de uma seqüência de observações. Este algoritmo leva em consideração tanto as probabilidades intervalares de emissão de

```

prob_back = infsup(0,0);

for j=1:Q
    beta(T,j)=infsup(1,1);
end

r=T-1; while (r >= 1)

    for j=1:Q
        var = infsup(0,0);
        for k=1:Q
            obs_ti = obs_int(r+1);
            var = ( trans_int(j,k)
                    * emissao_int(obs_ti,k)
                    * beta(r+1,k) ) + var;
        end
        beta(r,j) = var;
    end
    r = r-1;
end

```

Figura 2: *Algoritmo Forward Intervalar*

símbolos em um determinado estado quanto as probabilidades intervalares de transição de estados. Para contornar a alta complexidade computacional gerada pelo problema, o algoritmo utiliza técnicas de programação dinâmica. Seja $\delta(i, j)$ a probabilidade intervalar de uma seqüência ótima de estados em um modelo λ que produz as seqüências de observações s_1, \dots, s_i , sendo j o último estado. A seqüência ótima é determinada por:

Inicialização:

$$\delta(1, i) = \pi(i)e_i(s_1); \quad \psi(1, i) = 0$$

Recursão:

$$\delta(t, j) = \max(e_j(s_t)\delta(t-1, i)a_{ij}), 2 \leq t \leq T, 1 \leq j \leq N$$

$$\psi(t, i) = \operatorname{argmax}(\delta(t-1, i)a_{ij}), 2 \leq t \leq T, 1 \leq j \leq N$$

Finalização:

$$P^* = \max(\delta(t, i)), 1 \leq i \leq N$$

$$q^* = \operatorname{argmax}(\delta(t, i)), 1 \leq i \leq N$$

Seqüência Ótima: $q_t^* = \psi_{t+1}q_{t+1}^*, t = T-1, \dots, 1.$

O algoritmo de reestimação *Baum Welch* foi popularizado por [4] como algoritmo EM (*Expectation Maximization*). Resumidamente, este algoritmo tem como objetivo gerar novos modelos a partir de um modelo inicial através de um tipo de iteração. Este processo iterativo deve obedecer a um critério de parada, que pode ser um número definido pelo usuário ou então o menor número de máquina através do qual uma probabilidade possa ser representada. O algoritmo EM não foi desenvolvido especificamente para os Modelos Ocultos de Markov e sim para problemas genéricos de inferência estatística, sendo utilizado para localizar o valor de um parâmetro que maximiza a função de verossimilhança. Cada iteração deste algoritmo consiste de dois passos: o *E-Step* (Expectation) e o

M-Step (Maximization).

A determinação dos parâmetros do modelo, de forma a maximizar a probabilidade intervalar de uma seqüência de observações, não tem uma solução ótima conhecida. Utilizando o conceito de freqüência de ocorrência, o novo modelo $\tilde{\lambda} = (\tilde{A}, \tilde{E}, \tilde{\pi})$ é calculado a partir das iterações do algoritmo EM. As equações de reestimação podem ser escritas da seguinte forma:

$$\begin{aligned} \tilde{\pi}_i &= \gamma_1(i) \\ \tilde{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\ \tilde{b}_j(k) &= \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}. \end{aligned}$$

5 Exemplos e Discussão de Resultados

Para os exemplos que serão apresentados nesta seção, consideram-se os modelos ocultos de Markov λ_1 e λ_2 , definidos, respectivamente, pelos seguintes parâmetros: $\varepsilon_{\lambda_1} = \{a, b\}$, $\pi_{\lambda_1} = [0.25 \quad 0.75]$,

$$A_{\lambda_1} = \begin{bmatrix} 0.70 & 0.30 \\ 0.23 & 0.77 \end{bmatrix} \quad E_{\lambda_1} = \begin{bmatrix} 0.53 & 0.50 \\ 0.47 & 0.50 \end{bmatrix}$$

$$\text{e } \varepsilon_{\lambda_2} = \{a, b, c, d\}, \pi_{\lambda_2} = [0.60 \quad 0.40 \quad 0.00],$$

$$A_{\lambda_2} = \begin{bmatrix} 0.40 & 0.20 & 0.40 \\ 0.30 & 0.30 & 0.40 \\ 0.70 & 0.20 & 0.10 \end{bmatrix}$$

$$E_{\lambda_2} = \begin{bmatrix} 0.30 & 0.10 & 0.40 \\ 0.20 & 0.40 & 0.40 \\ 0.20 & 0.30 & 0.10 \\ 0.30 & 0.20 & 0.10 \end{bmatrix}$$

Para execuções dos *Algoritmos Forward* e *Backward* intervalares utiliza-se o modelo λ_1 . Dada a seqüência de observações $S = \text{“baaaabb”}$, as estimativas de probabilidade da sua ocorrência, obtidas com os algoritmos pontuais são, respectivamente, 0.00802323305992 e 0.00802323305992. Os algoritmos intervalares produziram os resultados apresentados na Tabela 1, onde observa-se que a medida que o grau de incerteza I dos parâmetros do modelo diminui, diminui também o raio do intervalo probabilidade determinado pelo algoritmo, crescendo a convergência para o resultado pontual.

Para analisar o *Algoritmo de Viterbi*, considerou-se o modelo λ_2 . A Figura 3 e a Figura 4 mostram, respectivamente, a execução dos *Algoritmos de Viterbi* pontual e intervalar, como imagens produzida pelo ferramenta implementada. A variável “path” contém a seqüência de estados ótima determinada pelo algoritmo. Observa-se que não houve perda na

I	Probabilidade Intervalar	Raio
10^{-1}	[0.00035246790761, 0.10278902508627]	$5.121827858932000 \times 10^{-2}$
10^{-3}	[0.00780158546935, 0.00825071350391]	$2.245640172781225 \times 10^{-4}$
10^{-5}	[0.00802098794435, 0.00802547875875]	$2.245407195617180 \times 10^{-6}$
10^{-7}	[0.00802321060587, 0.00802325551402]	$2.245407173043224 \times 10^{-8}$
10^{-9}	[0.00802323283537, 0.00802323328446]	$2.245407280249134 \times 10^{-10}$
10^{-11}	[0.00802323305767, 0.00802323306217]	$2.245419128410475 \times 10^{-12}$
10^{-13}	[0.00802323305989, 0.00802323305994]	$2.246813846085161 \times 10^{-14}$
10^{-15}	[0.00802323305991, 0.00802323305992]	$2.341876692568690 \times 10^{-16}$
10^{-17}	[0.00802323305991, 0.00802323305992]	$8.673617379884036 \times 10^{-18}$

Tabela 1: Relação entre a incerteza I dos parâmetros do modelo e a probabilidade intervalar de ocorrência da seqüência $S = \text{“baaaabb”}$

ENTRE COM A OBSERVACAO: aaaabbacdbccdadddcabbd
path= 2 3 3 3 1 2 3 2 2 3 2 2 2 3 2 2 2 3 2 3 2 3 2

Figura 3: Seqüência de estados produzida pelo *Algoritmo de Viterbi Intervalar*

ENTRE COM A OBSERVACAO: aaaabbacdbccdadddcabbd
path= 2 3 3 3 3 2 3 2 2 3 2 2 2 3 2 2 2 3 2 3 2 3 2

Figura 4: Seqüência de estados produzida pelo *Algoritmo de Viterbi pontual*

ENTRE COM A OBSERVACAO: ab

iter=0 intval Probabilidade= [0.2483, 0.2484]
iter=1 intval Probabilidade= [0.1327, 0.1328]
iter=2 intval Probabilidade= [0.1444, 0.1445]
iter=3 intval Probabilidade= [0.2242, 0.2243]
iter=4 intval Probabilidade= [0.5445, 0.5446]
iter=5 intval Probabilidade= [0.9162, 0.9163]
iter=6 intval Probabilidade= [0.9993, 0.9994]
iter=7 intval Probabilidade= [0.9999, 1.0001]
iter=8 intval Probabilidade= [0.9999, 1.0001]
iter=9 intval Probabilidade= [0.9983, 1.0017]

Figura 6: Probabilidades de ocorrência da seqüência de observações $S = ab$ no *Algoritmo Baum Welch intervalar*

significância do resultado fornecido pelo algoritmo intervalar, pois as seqüências de estados produzidas são quase iguais.

Para exemplificar os resultados do *Algoritmo Baum Welch Intervalar* utilizou-se o modelo λ_1 . A Figura 5 mostra, a medida que, em cada iteração do algoritmo pontual, são reestimados os parâmetros do modelo, as probabilidades de ocorrência da seqüência $S = ab$.

ENTRE COM A OBSERVACAO: ab
iter=0 probabilidade= 0.24838000000000
iter=1 probabilidade= 0.13277340743544
iter=2 probabilidade= 0.14449587265466
iter=3 probabilidade= 0.22427457481497
iter=4 probabilidade= 0.54459242516218
iter=5 probabilidade= 0.91624209821461
iter=6 probabilidade= 0.99930694382298
iter=7 probabilidade= 0.9999999886959
iter=8 probabilidade= 1.00000000000000
iter=9 probabilidade= 1.00000000000000

Figura 5: Probabilidades de ocorrência da seqüência de observações $S = ab$ a cada iteração do *Algoritmo Baum Welch pontual*

O *Algoritmo Baum Welch* pontual fornece, na oitava iteração, o modelo otimizado λ_3 , que é capaz de produzir a seqüência $S = ab$ com 100% de probabilidade, cujos novos parâmetros são: $\pi = [0 \ 1]$,

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Com a execução do *Algoritmo Baum Welch* intervalar para o modelo λ_1 , obteve-se as probabilidades intervalares de ocorrência da seqüência $S = ab$

apresentadas na Figura 6. Essas probabilidades são obtidas a medida em que são realizadas novas reestimações dos parâmetros do modelo a cada iteração. Utilizou-se o *format short* para uma melhor visualização dos intervalos.

A Figura 7 apresenta os novos parâmetros dos modelos obtidos com a execução do *Algoritmo Baum Welch intervalar*. Denota-se *trans.int* a matriz de transição A , *emissao.int* a matriz de emissão de símbolos E e *probin.int* o vetor de probabilidades iniciais π .

6 Conclusões

A utilização da Matemática Intervalar no tratamento de dados incertos, imprecisos e/ou de natureza qualitativa vem crescendo atualmente, justificando a necessidade do estudo de versões intervalares para modelos conhecidos, e para os métodos e algoritmos relacionados à análise desses modelos. Esta tarefa não é trivial, devido à alta complexidade computacional das extensões intervalares puras de algoritmos pontuais para intervalares. Além disso, resultados de algoritmos intervalares podem frequentemente produzir resultados com diâmetro muito grande, o que não tem significado.

A extensão do conceito de probabilidade para probabilidade intervalar permitiu que se avançasse no sentido do estudo de modelos estocásticos inter-

```

iter = 5
intval trans_int = [ 0.0000, 0.0001] [ 0.0000, 0.0001] [ 0.9995, 0.9996] [ 0.0000, 0.0001]
intval emissao_int = [ 0.0000, 0.0001] [ 0.9997, 0.9998] [ 0.9999, 1.0001] [ 0.0002, 0.0003]
intval probini_int = [ 0.0000, 0.0001] [ 0.9999, 1.0001]

iter = 7
intval trans_int = [ 0.0000, 0.0001] [ 0.0000, 0.0001] [ 0.9999, 1.0001] [ 0.0000, 0.0001]
intval emissao_int = [ 0.0000, 0.0001] [ 0.9999, 1.0001] [ 0.9999, 1.0001] [ 0.0000, 0.0001]
intval probini_int = [ 0.0000, 0.0001] [ 0.9999, 1.0001]

iter = 9
intval trans_int = [ 0.0000, 0.0001] [ 0.0000, 0.0001] [ 0.9874, 1.0127] [ 0.0000, 0.0001]
intval emissao_int = [ 0.0000, 0.0001] [ 0.9935, 1.0065] [ 0.9935, 1.0065] [ 0.0000, 0.0001]
intval probini_int = [ 0.0000, 0.0001] [ 0.9967, 1.0033]

```

Figura 7: Reestimação de novos modelos ocultos de Markov intervalares

valares. Neste contexto, este trabalho introduziu o conceito de Modelos Ocultos de Markov Intervalares e apresentou versões intervalares para os algoritmos *Forward*, *Backward*, *Viterbi* and *Baum Welch*, que são utilizados na solução dos três problemas relacionados a esses modelos: *Avaliação*, *Codificação* e *Estimação de Parâmetros*.

Foram realizados vários testes que comprovaram a eficiência e eficácia destes algoritmos, que produziram resultados intervalares válidos e de diâmetro aceitável, considerando o grau de incerteza atribuído aos parâmetros dos modelos. Alguns exemplos selecionados foram apresentados.

Trabalhos futuros estão relacionados ao estudo de Modelos Ocultos de Markov Intervalares associados aos processos de trocas sociais em sistemas multiagentes, onde valores de trocas, de natureza qualitativa, são representados por intervalos [5].

Agradecimentos. Este trabalho foi parcialmente financiado pelo CTINFO/CNPq e FAPERGS.

Referências

- [1] M.A. Campos, “Uma Extensão Intervalar para a Probabilidade Real”, Tese de Doutorado, Centro de Informática/UFPE, 1997.
- [2] M.A. Campos, Interval probabilities, application to discrete random variables, “Seleta do XXII CNMAC” (E.X.L. de Andrade et al., eds.), TEMA, Vol. 1, n. 2, pp. 333-344, 2000.
- [3] M.A. Campos, G.P. Dimuro, A.C. Costa, J.F.F. Araújo, and A.M. Dias, Probabilidade Intervalar e Cadeias de Markov Intervalares no Maple, “Seleta do XXIV CNMAC” (E.X.L. de Andrade et al., eds.), TEMA, Vol 3, n. 2, pp. 53-62, 2002.
- [4] A.P. Dempster, N.M. Laird, and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society Series B* 39 (1) (1977) 1-38.
- [5] G.P. Dimuro, G.P., A.C.R.Costa, and L.A.M. Palazzo, System of Exchange Values as Tools for Multi-Agent Organizations, *Journal of the Brazilian Computer Society* (2005). (to appear in July 2005)
- [6] D. Goldberg, What Every Computer Scientist Should know About Floating-Point Arithmetic, “ACM Comp. Surveys, 23(1) (1991) 5-48.
- [7] R.B. Keafort and V. Kreinovich (eds), “Applications of interval computations”, Kluwer, Boston, 1996.
- [8] I.O. Kozine and L.V. Utkin, *Interval-Valued Finite Markov Chains*, *Reliable Computing* 8(2) (2002) 97-113.
- [9] U.W. Kulisch, W.L. Miranker, “Computer Arithmetic in Theory and Practice”, Academic Press, New York, 1981.
- [10] M.B. Monagan, K.O. Geddes, K. M. Heal, G. Labahn, and S. M. Vorkoetter, “Maple V: Program. Guide”, Springer, N. York, 1998.
- [11] R.E. Moore, “Methods and applications of interval analysis”, SIAM, Philadelphia, 1979.
- [12] L.R. Rabiner and B.H. Juang, An Introduction to Hidden Markov Models. *IEEE ASSP Magazine* 3(1) (1989) 4-16.
- [13] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77 (1989) 257-286.
- [14] S.M. Rump, IntLab - Interval Laboratory, in “Developments in Reliable Computing” (T. Csendes, ed.), Kluwer, Dordrecht, 1999.
- [15] K. Weichselberger, Axiomatic foundations of the theory of interval-probability, “Symposia Gaussiana”, Conference B: Statistical Sciences, pp 47-64, Munich, Germany, August 2-7, 1993.