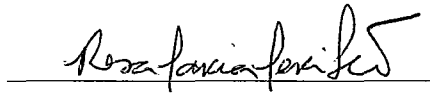
UM MODELO HMM HIERÁRQUICO PARA USUÁRIOS INTERATIVOS ACESSANDO UM SERVIDOR MULTIMÍDIA

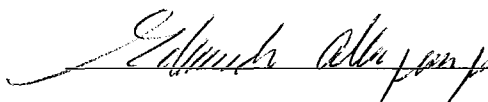
Carolina Cerqueira Le Brun de Vielmond

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

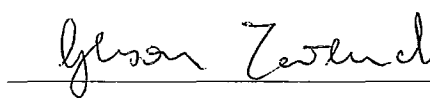
Aprovada por:



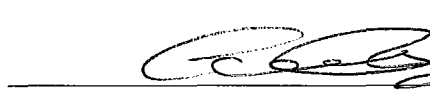
Prof.ª Rosa Maria Meri Leão, Dr.



Prof. Edmundo Albuquerque de Souza e Silva, Ph.D.



Prof. Gerson Zaverucha, Ph.D.



Prof. Célio Vinicius Neves de Albuquerque, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

NOVEMBRO DE 2007

VIELMOND, CAROLINA CERQUEIRA LE BRUN DE

Um modelo HMM hierárquico para usuários interativos acessando um servidor multimídia [Rio de Janeiro] 2007

XI, 85 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2007)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Vídeo sob Demanda
2. Comportamento dos usuários
3. Modelagem
3. Cadeia de Markov Oculta

I. COPPE/UFRJ II. Título (Série)

Agradecimentos

A minha família, pelo incentivo ao meu contínuo aperfeiçoamento profissional e pessoal. Ao João que tem um papel muito especial em minha vida e que tem me apoiado em todos os momentos.

Agradeço aos professores Edmundo e Rosa pela grande ajuda e incentivo na realização deste trabalho. Durante todo o período que passei no LAND, aprendi muito com ambos.

Agradeço também a todos os amigos do LAND, por formarem uma equipe tão unida e maravilhosa, sempre dispostos a ajudar. E, especialmente a Carol por seu carinho e apoio incondicional.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM MODELO HMM HIERÁRQUICO PARA USUÁRIOS INTERATIVOS ACESSANDO UM SERVIDOR MULTIMÍDIA

Carolina Cerqueira Le Brun de Vielmond

Novembro/2007

Orientadores: Rosa Maria Meri Leão
Edmundo Albuquerque de Souza e Silva

Programa: Engenharia de Sistemas e Computação

A maior parte dos mecanismos de compartilhamento de recursos desenvolvidos para tornar os serviços de vídeo sob demanda (VoD) escaláveis tem sido avaliados considerando que o acesso dos usuários é seqüencial. É comum em alguns tipos de aplicações, como ensino a distância, que os usuários realizem ações interativas como parada, avanço e retorno do vídeo. Portanto é importante desenvolver modelos que permitam avaliar o desempenho de servidores VoD em um cenário de interatividade. Neste trabalho apresentamos um novo modelo para representar o comportamento interativo de usuários acessando um servidor multimídia para ensino a distância. O modelo é um HMM hierárquico onde, no primeiro nível, são representadas as dependências em uma escala de tempo proporcional a duração de um *slide* e, no segundo nível, são representadas as dependências em uma escala de tempo que corresponde a duração de uma sessão. Resultados obtidos quando o modelo, parametrizado por logs reais, é usado para dimensionar um servidor mostram a boa acurácia do modelo proposto.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A HIERARCHICAL HMM MODEL FOR INTERACTIVE USERS ACCESSING A MULTIMEDIA SERVER

Carolina Cerqueira Le Brun de Vielmond

November/2007

Advisors: Rosa Maria Meri Leão
Edmundo Albuquerque de Souza e Silva

Department: Systems Engineering and Computer Science

A number of stream sharing mechanisms have recently been evaluated considering user sequential access. However, a high degree of user interactivity has been observed in distance learning applications based on multimedia servers. Therefore, it is important to develop accurate models to evaluate the performance of stream sharing techniques under user interactive access. Focusing on interactive users, we propose a new model to represent the user behavior when accessing a multimedia server. The model is a hierarchical HMM where the temporal dependencies in a short time scale (slide duration) are represented in one level, and the temporal dependencies during one user session are represented in a second level. The results show the good accuracy of the model (parameterized by a real system log) when it is used to size a multimedia server.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Atualmente, aplicações mídia contínua têm se tornado cada vez mais populares, tanto para fins educacionais quanto para de entretenimento. Aplicações deste tipo enfrentam desafios em função da largura de banda necessária, aos requisitos de tempo real sobre a entrega de mídia e da possibilidade de acesso parcial ou interativo à mídia.

O foco deste trabalho está em aplicações de ensino à distância, onde as aulas são previamente gravadas e armazenadas em servidores. O conteúdo das aulas é composto de áudio e vídeo sincronizado com *slides*. Os alunos acessam as aulas através de um *software* cliente, que possui controles tais como pausar, avançar e retroceder a aula. Também é comum disponibilizar um índice, que é uma lista com os tópicos da aula para facilitar a navegação do aluno entre diferentes pontos de interesse da aula. Estas aplicações geralmente armazenam em *traces* as ações realizadas pelos usuários durante uma sessão, i. e. tempo em que o aluno fica assistindo uma determinada aula.

A forma mais simples de atender os clientes que utilizam um serviço de vídeo sob demanda (VoD) é através do escalonamento de um fluxo *unicast* para cada requisição dos clientes. O resultado é então um crescimento linear da banda do servidor. Como esta banda é um recurso limitado, o uso de mecanismos de compartilhamento de

recursos passa a ser importante para permitir a escalabilidade do sistema.

Muitos trabalhos na literatura tratam da caracterização do tráfego total a que um sistema é submetido, sem se preocupar em como esta carga foi gerada. Entretanto, a carga de trabalho é função do comportamento dos usuários do sistema. Estudos mostram que um alto grau de interatividade tem sido observado em diversos tipos de cargas reais [4, 18, 15, 24], principalmente nas aplicações de ensino a distância, onde ações como parada, avanço e retorno do vídeo são muito comuns. Entretanto, a maioria das técnicas propostas na literatura foram avaliadas considerando a premissa de acesso seqüencial - acesso à mídia completa, do início ao fim sem qualquer interrupção.

Um ambiente com um alto grau de interatividade pode comprometer a eficiência de mecanismos de compartilhamento de recursos, aumentando o tempo de acesso ao material armazenado e em conseqüência diminuindo drasticamente a qualidade do serviço fornecida ao usuário. Existem poucos trabalhos na literatura que avaliam o desempenho de técnicas de compartilhamento de banda na presença de usuários com interatividade [18, 5].

O uso de modelos que capturem o comportamento interativo dos usuários é importante para avaliar e desenvolver técnicas para prover serviços de VoD com interatividade. Na literatura encontramos apenas alguns modelos de usuários interativos que são baseados em *traces* coletados de servidores multimídia em operação.

Neste trabalho estudamos o comportamento de usuários acessando um servidor de ensino a distância **em operação** com o objetivo de criar um modelo para geração de carga sintética. A geração de carga sintética é importante para que a avaliação dos protocolos possa ser realizada com uma quantidade bastante grande de *logs*, já que não se tem disponível tantos *logs* reais. Existem trabalhos na literatura que utilizam modelos deste tipo para a busca antecipada de conteúdo no servidor através de um modelo de previsão da próxima ação [8] e para analisar técnicas de compartilhamento de banda [10, 11, 17].

Nosso modelo é baseado em *logs* de ações interativas de alunos acessando as aulas do curso de graduação de Tecnologia em Sistemas de Computação do Consórcio CEDERJ (Centro de Educação Superior a Distância do Rio de Janeiro), iniciado em Março de 2005 e contando atualmente com mais de 1000 alunos matriculados.

O modelo é uma variação da abordagem clássica de modelos de Markov ocultos (*hidden Markov models* - HMMs) [16], onde restringimos a distribuição das observações dentro de um estado oculto. Ele foi inspirado no trabalho de [20], que propôs este novo modelo, além de avaliar diferentes modelos de Markov ocultos como preditores de estatísticas de perda de pacotes em redes de computadores de curto prazo.

No trabalho de [20], o modelo resultante pode ser visto como um HMM hierárquico, com uma cadeia de Markov de 2 estados operando dentro de cada estado oculto. A estrutura deste modelo possui duas propriedades interessantes. Primeiramente, restringindo o modelo, o número total de parâmetros a serem estimados é diminuído, reduzindo assim a complexidade da fase de treinamento. Em segundo, supondo tais padrões no conjunto de parâmetros, as dependências de curto prazo são capturadas nos eventos da perda com um modelo de Gilbert, enquanto a dinâmica de longo prazo é governada por uma cadeia de Markov oculta.

Neste trabalho, realizamos duas adaptações no modelo proposto em [20] para utilizá-lo para caracterizar o comportamento de usuários interativos: (i) permitir que dinâmica de curto prazo fosse capturada por uma cadeia de Markov discreta qualquer, e não apenas por um modelo de Gilbert, que limita a aplicabilidade do modelo; (ii) permitir que o número de observações geradas em cada estado oculto fosse variável, e não mais uma constante como no trabalho original. Vale ressaltar que o modelo resultante deste trabalho é genérico, podendo se adequar a outros sistemas devido às adaptações que realizamos no mesmo.

Algumas vantagens deste modelo em relação a outras propostas da literatura são a capacidade infinita de memória do modelo (em contraste com modelos Markovianos que possuem memória finita), um número reduzido de estados diminuindo a sua

complexidade e, parametrização usando uma quantidade bastante grande de *logs* de comportamento de usuários de um servidor real.

Validamos o modelo, apresentando o desempenho do mesmo em comparação com a abordagem clássica de modelos de Markov ocultos, e analisamos estatísticas de interatividade da carga gerada a partir do modelo proposto em comparação com a carga gerada pelo modelo probabilístico desenvolvido em [17]. Mostramos ainda, ao parametrizar o modelo com *logs* reais dos sistemas de ensino a distância CEDERJ [1] e MANIC (*Multimedia Asynchronous Networked Individualized Courseware*) [22], que nosso modelo é acurado para dimensionar um servidor.

Um resumo dos resultados deste trabalho foi apresentado em [26]. Esta dissertação está organizada da seguinte forma. No Capítulo 2 apresentamos alguns conceitos e notações relevantes, além do modelo em que nosso trabalho foi baseado. No Capítulo 3 faremos uma revisão de trabalhos na literatura que estão relacionados com nosso estudo. No Capítulo 4 apresentamos o modelo proposto e o procedimento para geração de *logs* sintéticos. No Capítulo 5 validamos o modelo e apresentamos uma análise comparativa do modelo proposto em relação a outro modelo encontrado na literatura. O Capítulo 6 apresenta conclusões e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo apresentamos os conceitos básicos de modelos de Markov ocultos, muito utilizados neste trabalho, e o detalhamento de um modelo no qual este trabalho foi baseado.

2.1 Modelos de Markov Ocultos

Modelos de Markov ocultos têm sido usados para modelar séries temporais tais como reconhecimento de palavras, perdas de pacotes em uma rede, dentre outras aplicações. Um modelo de Markov oculto, de forma geral, é composto por dois processos estocásticos dependentes entre si. O primeiro deles é uma cadeia de Markov. O segundo é um processo de observações, cuja distribuição, a qualquer instante de tempo, é completamente determinada pelo estado atual da cadeia. Neste tipo de modelo, os parâmetros desconhecidos são determinados através do processo de observações. Uma referência concisa sobre modelos de Markov ocultos pode ser encontrada em [16].

Seja o seguinte exemplo para ilustrar o significado de uma cadeia de Markov oculta: consideremos três moedas. Uma moeda tem 0.5 de probabilidade de sair cada face. A segunda moeda é viciada e a probabilidade de sair cara é de 0.7 e

coroa 0.3. A terceira moeda também é viciada e possui menor probabilidade de sair cara, 0.3, do que coroa, 0.7. Caso um jogador esteja utilizando mais de uma moeda, observaremos apenas o resultado, a seqüência de observações: cara e coroa. Como não sabemos quando ocorre a troca de moedas, podemos apenas inferir sobre este processo oculto.

Formalmente, um modelo de Markov oculto é definido pelos elementos descritos a seguir. Seja $\{Y_t\}$ a cadeia de Markov de N estados. A distribuição do estado inicial é dada pelo vetor de N dimensões π, com:

$$\pi_i = P(Y_1 = i). \tag{2.1}$$

As probabilidades de transição entre estados são controladas pela matriz $N \times N$, $\mathbf{A} = \{a_{ij}\}$, onde:

$$a_{ij} = P(Y_t = j | Y_{t-1} = i). \tag{2.2}$$

O processo de observações, $\{X_t\}$, tem M estados e é governado pela matriz $N \times M$, $\mathbf{B} = \{b_{ij}\}$, i.e.:

$$b_{ij} = P(X_t = j | Y_t = i). \tag{2.3}$$

Dados os significados probabilísticos de π, $\mathbf{A}$ e $\mathbf{B}$, as restrições a seguir serão sempre satisfeitas:

$$\sum_{i=1}^{N} \pi_i = 1, \tag{2.4a}$$

$$\sum_{j=1}^{N} a_{ij} = 1, \quad \forall i, \tag{2.4b}$$

$$\sum_{j=1}^{M} b_{ij} = 1, \quad \forall i. \tag{2.4c}$$

O modelo pode então ser representado através da tripla $\lambda = (\pi, \mathbf{A}, \mathbf{B})$.

Um primeiro passo em criar um modelo é especificar os espaços de estados sobre os quais $\{X_t\}$ e $\{Y_t\}$ estão definidos. Uma vez que $\{X_t\}$ é o processo de observações, seus estados são geralmente determinados pelo que está sendo modelado. Por outro

lado, caracterizar os estados da cadeia oculta, $\{Y_t\}$, pode ser um pouco mais abstrato. Por exemplo, em modelos para perdas de pacotes, os estados ocultos podem ser vistos como "estados da rede", guardando informação sobre as estatísticas de perdas em um dado momento.

Consideremos um vetor com T valores para o processo de observações, $\mathbf{x} = [x_1, \ldots, x_T]$. Sempre que não houver ambigüidade, usaremos a forma abreviada $X_{i:j}$ (ou a correspondente, $Y_{i:j}$) para denotar o evento composto que cada variável X_t (Y_t) no alcance $t = i, \ldots, j$ assume o valor x_t (y_t). No caso particular em que $i = j$, escreveremos X_i (ou de maneira equivalente, Y_i). Por outro lado, usaremos $\mathbf{X}$ (ou $\mathbf{Y}$) quando os sub-índices se referem a todas as variáveis $1, \ldots, T$, i.e., $X_{1:T}$ ($Y_{1:T}$).

Finalmente, também definimos as seguintes medidas de probabilidade, seguindo a notação de [16]:

$$\alpha_t(i) = P(X_{1:t}, Y_t = i|\lambda), \tag{2.5a}$$

$$\beta_t(i) = P(X_{t+1:T}|Y_t = i, \lambda), \tag{2.5b}$$

$$\gamma_t(i) = P(Y_t = i|\mathbf{X}, \lambda), \tag{2.5c}$$

$$\xi_t(i,j) = P(Y_t = i, Y_{t+1} = j|\mathbf{X}, \lambda). \tag{2.5d}$$

Onde $\alpha_t(i)$ e $\beta_t(i)$ são calculados através do algoritmo *forward-backward* e as seguintes identidades podem ser estabelecidas:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}, \tag{2.6a}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_{jx_{t+1}}\beta_{t+1}(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}. \tag{2.6b}$$

A principal dificuldade da abordagem de modelos de Markov ocultos é o problema de estimação de parâmetros, isto é, como inferir valores para λ dado um caminho amostral do processo de observações. Apesar dessa dificuldade, o algoritmo *Baum-Welch* é uma técnica muito bem sucedida para estimação iterativa de parâmetros de máxima verossimilhança para modelos de Markov ocultos. O método começa a partir de uma atribuição arbitrária de valores para λ e produz estimativas

sucessivamente melhores, garantindo convergência para um máximo local na função de verossimilhança, sempre que um existir, [7].

A função de verossimilhança dos parâmetros, $\lambda = (\pi, \mathbf{A}, \mathbf{B})$, para uma amostra, $\mathbf{X}$, pode ser escrita como:

$$\begin{aligned} L(\lambda|\mathbf{X}) &= P(\mathbf{X}|\lambda) \\ &= \sum_{\forall \mathbf{y}} P(\mathbf{X}, \mathbf{Y}|\lambda). \end{aligned} \tag{2.7}$$

Onde a medida $P(\mathbf{X}, \mathbf{Y}|\lambda)$ é chamada *verossimilhança dos dados completos*, uma vez que envolve dados observáveis e ocultos, representados por $x_{1:T}$ e $y_{1:T}$, respectivamente. Para um modelo de Markov oculto, essa função é definida como:

$$P(\mathbf{X}, \mathbf{Y}|\lambda) = P(Y_1|\lambda)P(X_1|Y_1, \lambda)\prod_{t=2}^{T} P(Y_t|Y_{t-1}, \lambda)P(X_t|Y_t, \lambda), \tag{2.8}$$

onde temos as correspondências:

$$P(Y_1|\lambda) = \pi_{y_1}, \tag{2.9a}$$

$$P(Y_t|Y_{t-1}, \lambda) = a_{y_{t-1}, y_t}, \tag{2.9b}$$

$$P(X_t|Y_t, \lambda) = b_{y_t, x_t}. \tag{2.9c}$$

Logo, a Equação (2.8) pode ser reescrita como:

$$P(\mathbf{X}, \mathbf{Y}|\lambda) = \pi_{y_1} b_{y_1, x_1} \prod_{t=2}^{T} a_{y_{t-1}, y_t} b_{y_t, x_t}. \tag{2.10}$$

A cada iteração, o algoritmo Baum-Welch maximiza a *função auxiliar*, $Q(\lambda|\overline{\lambda})$, em relação a λ, fazendo uso da estimativa atual dos parâmetros, $\overline{\lambda}$:

$$Q(\lambda|\overline{\lambda}) = \sum_{\forall \mathbf{y}} \log P(\mathbf{X}, \mathbf{Y}|\lambda) P(\mathbf{Y}|\mathbf{X}, \overline{\lambda}). \tag{2.11}$$

Usando a definição da verossimilhança conjunta, $P(\mathbf{X}, \mathbf{Y}|\lambda)$, a Equação (2.11) pode ser dividida em três termos independentes:

$$Q(\lambda|\overline{\lambda}) = Q_1(\pi|\overline{\lambda}) + Q_2(\mathbf{A}|\overline{\lambda}) + Q_3(\mathbf{B}|\overline{\lambda}), \tag{2.12}$$

onde $Q_1(\pi|\overline{\lambda})$, $Q_2(\mathbf{A}|\overline{\lambda})$, $Q_3(\mathbf{B}|\overline{\lambda})$ são dados por:

$$Q_1(\pi|\overline{\lambda}) = \sum_{i=1}^{N} \log \pi_i \, \gamma_1(i), \tag{2.13a}$$

$$Q_2(\mathbf{A}|\overline{\lambda}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \log a_{ij} \sum_{t=1}^{T-1} \xi_t(i,j), \tag{2.13b}$$

$$Q_3(\mathbf{B}|\overline{\lambda}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \log b_{ij} \sum_{t=1}^{T} \mathbb{I}\{x_t = j\}\gamma_t(i). \tag{2.13c}$$

Na Equação (2.13c), utilizamos a notação $\mathbb{I}\{c\}$, para representar a *função indicadora* de uma condição c, que vale 1 quando a condição é satisfeita, ou 0 no caso contrário.

Maximizando cada termo de (2.13) e levando em consideração as restrições estocásticas de (2.4), chegamos às fórmulas de estimação de parâmetros:

$$\pi_i = \gamma_1(i), \tag{2.14a}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{2.14b}$$

$$b_{ij} = \frac{\sum_{t=1}^{T} \mathbb{I}\{x_t = j\}\gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}. \tag{2.14c}$$

2.2 Modelo HMM Hierárquico [20]

O modelo é uma variação da abordagem clássica de modelos de Markov ocultos (*hidden Markov models* - HMMs) [16], onde restringimos a distribuição das observações dentro de um estado oculto. Este modelo foi inicialmente definido em [20], que propôs este novo modelo, além de avaliar diferentes modelos de Markov ocultos como preditores de estatísticas de perda de pacotes em redes de computadores de curto prazo. O raciocínio por trás do modelo HMM hierárquico é que as correlações de curto prazo podem ser capturadas por um processo simples, enquanto a dinâmica em escalas de tempo maiores é governada pela cadeia de Markov oculta. Um resumo dos resultados daquele trabalho também podem ser encontrados em [21].

No trabalho de [20], o modelo resultante pode ser visto como um HMM hierárquico, com uma cadeia de Markov de 2 estados operando dentro de cada estado

oculto. A Figura 2.1 ilustra o modelo proposto em [20], onde as as dependências de curto prazo são capturadas nos eventos da perda com um modelo de Gilbert, enquanto a dinâmica de longo prazo é governada por uma cadeia de Markov oculta.

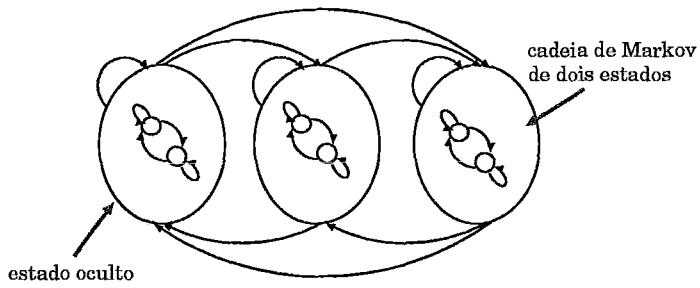


Figura 2.1: Modelo HMM hierárquico proposto em [20]

A estrutura deste modelo possui duas propriedades interessantes. Primeiramente, restringindo o modelo, o número total de parâmetros a serem estimados é diminuído, reduzindo assim a complexidade da fase de treinamento. Em segundo, supondo tais padrões no conjunto de parâmetros, as dependências de curto prazo são capturadas nos eventos da perda com um modelo de Gilbert, enquanto a dinâmica de longo prazo é governada por uma cadeia de Markov oculta.

Suponhamos que as transições entre estados ocultos ocorram apenas a cada S observações, sendo assim as medições de pacotes estão segmentadas em conjuntos de tamanho S. Mais especificamente, o símbolo x_t denota um vetor de medições, $[x_{t,1}, \ldots, x_{t,S}]$, representando o resultado para cada um dos pacotes no t-ésimo grupo. De forma análoga, redefinimos as variáveis das observações, X_t, como vetores das variáveis, $[X_{t,1}, \ldots, X_{t,S}]$.

Para cada estado oculto, i, temos os parâmetros da cadeia de Markov de 2 estados, ilustrados na Figura 2.2, e dados por:

$$r_i = P(X_{t,1} = 1 | Y_t = i); \tag{2.15a}$$

$$p_i = P(X_{t,s} = 1 | X_{t,s-1} = 0, Y_t = i), \quad 1 < s \leq S; \tag{2.15b}$$

$$q_i = P(X_{t,s} = 0 | X_{t,s-1} = 1, Y_t = i), \quad 1 < s \leq S. \tag{2.15c}$$

Nos referimos ao modelo como a tupla $\lambda = (\pi, \mathbf{A}, \mathbf{r}, \mathbf{p}, \mathbf{q})$, onde $\mathbf{r}$, $\mathbf{p}$, e $\mathbf{q}$ são vetores, contendo os respectivos parâmetros r_i, p_i, q_i, para cada estado, i.

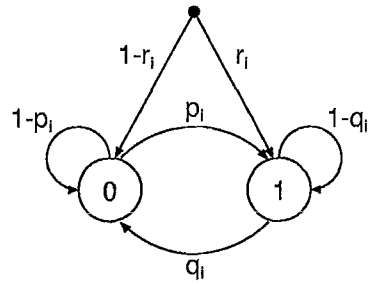


Figura 2.2: Parâmetros da cadeia de Markov de 2 estados para cada estado oculto do modelo proposto.

Para calcular a probabilidade de uma observação, não é necessário o conhecimento completo da medição de perda para cada pacote. É suficiente manter registro apenas das seguintes estatísticas, em cada grupo de medidas, x_t:

$$x_{t,1} = \text{resultado do primeiro pacote em } x_t; \tag{2.16a}$$

$$S_t^{ij} = \text{número de transições de } i \text{ para } j \text{ em } x_t, \quad i,j \in \{0,1\}; \tag{2.16b}$$

onde, para S_t^{ij}, é válida a restrição:

$$\sum_{i\in\{0,1\}} \sum_{j\in\{0,1\}} S_t^{ij} = S - 1, \quad \forall t. \tag{2.17}$$

Dada uma instância de x_t, estamos interessados em computar a probabilidade do evento $X_t = x_t$, dado o estado oculto no t-ésimo lote, y_t. Usando as estatísticas definidas acima, temos:

$$b_{y_t,x_t} = \begin{cases} r_{y_t}(p_{y_t})^{S_t^{01}}(1-p_{y_t})^{S_t^{00}}(q_{y_t})^{S_t^{10}}(1-q_{y_t})^{S_t^{11}} & , \quad \text{se } x_{t,1} = 1, \\ (1-r_{y_t})(p_{y_t})^{S_t^{01}}(1-p_{y_t})^{S_t^{00}}(q_{y_t})^{S_t^{10}}(1-q_{y_t})^{S_t^{11}} & , \quad \text{se } x_{t,1} = 0. \end{cases} \tag{2.18}$$

Na Seção 4.3.2 e no Apêndice A.5 do trabalho de [20] foi mostrado que é possível encontrar de forma eficiente os parâmetros do modelo de Gilbert do modelo HMM hierárquico proposto. Nesta dissertação, estenderemos o modelo HMM hierárquico

de 2 estados para uma cadeia qualquer. Os passos necessários são semelhantes ao trabalho de [20], e podem ser verificados na Seção 4.2.

$$r_i = \frac{\sum_{t=1}^{T} \mathbb{I}\{x_{t,1} = 1\}\gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{2.19a}$$

$$p_i = \frac{\sum_{t=1}^{T} S_t^{01}\gamma_t(i)}{\sum_{t=1}^{T} (S_t^{01} + S_t^{00})\gamma_t(i)}, \tag{2.19b}$$

$$q_i = \frac{\sum_{t=1}^{T} S_t^{10}\gamma_t(i)}{\sum_{t=1}^{T} (S_t^{10} + S_t^{11})\gamma_t(i)}. \tag{2.19c}$$

Capítulo 3

Trabalhos Relacionados

Ainda são poucos os trabalhos que caracterizam o comportamento interativo de usuários acessando um servidor de vídeo, principalmente para vídeos educacionais, e propõem um modelo para representar o comportamento desses usuários. Neste capítulo apresentamos trabalhos encontrados na literatura sobre modelos parametrizados através de cargas reais.

3.1 Modelo de comportamento de alunos baseado na busca antecipada de *slides* em sistemas educacionais *online* [8]

Em [8] foi feita a análise de *logs* coletados do sistema MANIC (*Multimedia Asynchronous Networked Individualized Courseware*) de aulas com conteúdo de áudio sincronizado com *slides*. Naquele trabalho foi proposto o uso de um modelo HMM (*Hidden Markov Model*) [16] para capturar o comportamento individual de cada aluno. Os autores estudaram o uso de modelos HMMs para implementar algoritmos de predição com o objetivo de realizar a busca antecipada de *slides*.

Os registros dos usuários foram transformados em uma seqüência de símbolos

que representam as suas ações, i. e. próximo *slide*, *slide* anterior, índice e início. A Figura 3.1 ilustra o modelo proposto.

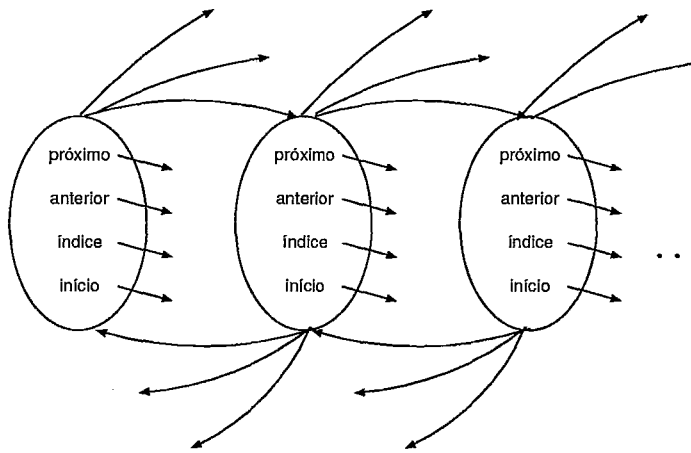


Figura 3.1: Modelo HMM proposto em [8]

A predição das ações dos usuários é realizada em duas etapas: (i) a cadeia é re-treinada a partir dos *logs* reais; (ii) a próxima ação é prevista. Em análises feitas neste trabalho, foi verificado que em torno de 70% das observações geradas pelo modelo coincidiam com as reais. Sendo que, das ações *next* observadas aproximadamente 60% foram previstas corretamente e das ações diferentes de *next* aproximadamente 70% foram previstas erroneamente.

O modelo de [8] tem como objetivo caracterizar o comportamento de usuários individuais para predizer a próxima ação do mesmo. Sendo assim, o modelo especifica apenas ações interativas relacionadas a movimentações entre os *slides*, não sendo possível por exemplo obter informações sobre as ações de pausa realizadas pelo usuário. Outros dados importantes para a geração de carga sintética também não estão disponíveis, tais como a duração dos tempos em *play* e em pausa. O modelo também não prevê o fim da sessão do usuário. Desta maneira, para que o modelo de [8] possa ser usado para gerar carga sintética, seriam necessários estudos para realizar algumas adaptações no mesmo.

3.2 Escalabilidade de Protocolos com Compartilhamento de Banda para Cargas de Mídia Contínua Realista [17]

O objetivo principal do modelo de [17] é avaliar o desempenho de protocolos de compartilhamento de banda, em particular o *Bandwith Skimming* [27], utilizando cargas de mídia contínua realistas. As cargas de mídia contínua realistas são geradas por um modelo probabilístico que captura os aspectos essenciais dos perfis de comportamento interativo, e que tem como entradas um *trace* real de sessões para um certo objeto, além da taxa de chegada de sessões.

As cargas reais são oriundas de diversos domínios de aplicação de sistemas de mídia contínua e incluem diversos perfis de comportamento interativo. Estes perfis incluem cargas de áudio (com baixo nível de interatividade), vídeos curtos (de entretenimento e educacionais, ambos com nível médio de interatividade), e vídeos longos (educacionais, com alto nível de interatividade). O trabalho [18] descreve como foi feita esta classificação de acordo com níveis de interatividade. Os perfis de alta interatividade são aqueles em que a duração média das requisições está abaixo de 20% da duração da mídia, a posição média inicial está entre 30% e 60% do tamanho da mídia e o número de requisições por sessão é de no mínimo 3. Nos perfis de média interatividade, a duração média das requisições está abaixo de 20% da duração da mídia, a posição média inicial está abaixo de 30% ou acima 60% do tamanho da mídia e o número de requisições por sessão é menor que 3. Já os perfis de baixa interatividade, a duração média das requisições são mais longas (pelo menos 20% da duração da mídia), a posição média inicial está concentrada no início da mídia e o número de requisições por sessão é menor que 2. Esta variedade de perfis permite cobrir as características de uma carga que afetam significamente a escalabilidade do protocolo de compartilhamento de banda *Bandwith Skimming* [4]: (i) posição inicial (o ponto, na mídia, onde uma requisição se inicia); (ii) a duração do segmento de mídia requisitado; (iii) o tamanho dos saltos; (iv) o tipo de requisição (salto, pausa);

(v) número de requisições por sessão.

Uma carga de mídia contínua contém um conjunto de sessões de clientes e cada sessão é composta por uma ou mais requisições interativas. A chegada de sessões foi modelada por um processo de Poisson. A chegada de requisições em uma sessão, a posição inicial da requisição e a sua duração são geradas pelo modelo probabilístico.

O modelo probabilístico é representado por meio de um diagrama de estados e a mídia é representada por segmentos de 10 segundos. Com a varredura da carga, são determinados os estados, as transições possíveis a partir de cada estado, a duração média de cada estado e a freqüência de cada transição. Um exemplo deste diagrama é mostrado na Figura 3.2. Todas as probabilidades de transição são calculadas a partir das freqüências relativas medidas na carga real. Os tipo de estados presentes no diagrama são:

- Início de Sessão (SOSession) - É o estado inicial do diagrama e não tem duração associada. A ele sempre segue a exibição de um segmento (Segmento Comum ou Fim de Requisição).
- Segmento Comum (representado no diagrama pelos círculos) - É um segmento que é seguido por outro, com duração fixa de 10 segundos (tamanho dos segmentos). Esse estado pode ser seguido por outro estado Segmento Comum, ou um estado Fim de Requisição, caso a requisição termine, ou um estado Fim de Sessão, caso a sessão e a requisição terminem.
- Fim de Requisição (EOR) - Este estado termina uma requisição, mas não a sessão. Possui duração exponencial, onde seu parâmetro é igual à média da duração deste estado. Este estado é seguido pelo estado Pausa ou pelo estado de Fim de Sessão.
- Pausa (Pause) - Representa o período de pausa após um estado de Fim de Requisição e antes do início da próxima requisição. Possui duração exponencial, onde seu parâmetro é igual à média dos tempos de pausa das requisições que percorreram este estado.

- Fim de Sessão (EOSession) - É o estado final do diagrama.

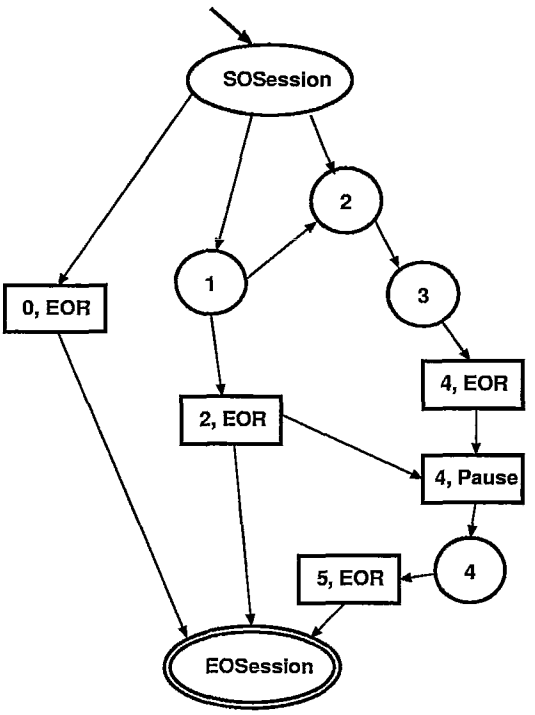


Figura 3.2: Exemplo do modelo probabilístico proposto em [17]

Como resultado, pôde-se constatar que a posição inicial das requisições geradas pelo modelo é similar à posição inicial da carga real. No caso da duração média das requisições, esta similaridade é mais evidente nos perfis de interatividade média e baixa. Como a duração média das requisições é um fator muito importante para a análise do protocolo de compartilhamento, foi determinada a sua distribuição e constatou-se que estas coincidem com as distribuições e faixas de valores encontrados na caracterização ampla das cargas de mídia contínua efetuada em [4].

Em [18], foi realizada uma validação deste modelo, comparando um conjunto de cargas reais com um conjunto de cargas sintéticas (com taxas de chegadas similares). Foram computadas as distribuições acumuladas da posição inicial das requisições e da duração das requisições, para cada par dos *traces* real e sintético, e constatou-se que o erro médio quadrático (*mean squared error* - MSE) [19] estava sempre

abaixo de 0.03. Em seguida, foram calculados o número médio de fluxos simultâneos transmitidos pelo servidor para cada par de *traces* e obteve-se uma aproximação da carga sintética à real com erro abaixo de 27%. Como conclusão de [18], dada a complexidade dos padrões de interatividade da carga real, espera-se que a carga gerada pelo modelo, pelo menos com relação às estatísticas de primeira ordem, capture as características essenciais da carga real.

O modelo probabilístico de [17] possui algumas limitações dentre as quais podemos citar: (i) quando o usuário assiste continuamente o vídeo, ele percorre uma seqüência de estados de tamanho fixo de 10 s; (ii) o tempo em pausa é caracterizado por uma distribuição exponencial (a caracterização da carga do CEDERJ feita no trabalho de [1], mostrou que o tempo em pausa é melhor caracterizado por uma distribuição hiperexponencial de 4 estágios); (iii) os saltos estão associados a ocorrência de pausas (quando o estado de retorno da pausa é diferente do estado de origem).

3.3 Caracterização e Modelagem do Comportamento de Usuários Acessando um Vídeo de Ensino a Distância [24]

Em [24] foi analisado um conjunto de *logs* do MANIC, com conteúdo de vídeo e áudio sincronizados com *slides*. Aquele trabalho apresentou a caracterização da carga dos usuários acessando o sistema multimídia, através de um conjunto de estatísticas que descrevem a interatividade dos usuários. O trabalho foi além da caracterização da carga e também sugeriu um modelo para capturar as estatísticas de comportamento do usuário, e gerar carga sintética para análise do desempenho de um sistema multimídia. O modelo consiste em um HMM embutido nos instantes em que o usuário realiza interações ou transiciona entre os *slides*.

A parametrização do modelo foi realizada através de cargas reais e buscava especificar a posição do usuário a cada instante de tempo. As observações do HMM

são símbolos em um alfabeto com 7 elementos: próximo *slide*, pausa, salto de 1 *slide* para frente, salto de 1 *slide* para trás, salto de 2 *slides* para frente, salto de 2 *slides* para trás e final da sessão. A Figura 3.3 ilustra o modelo proposto. Experimentos foram realizados com as cargas real e sintética de forma a validar o modelo.

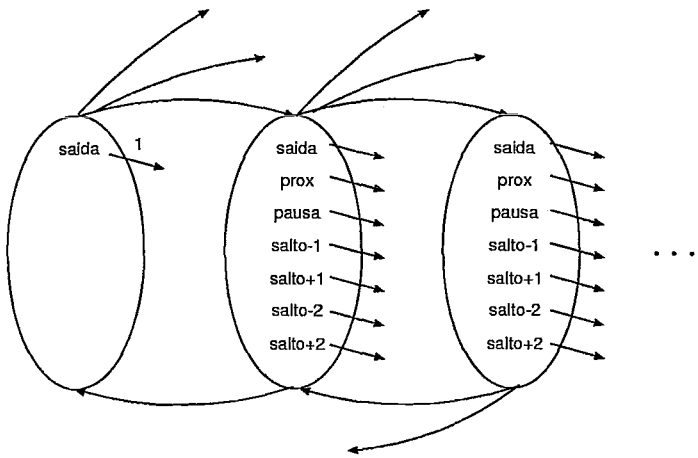


Figura 3.3: Modelo HMM proposto em [24]

Assim como em [24], o modelo HMM hierárquico proposto neste trabalho também é parametrizado através de cargas reais e busca especificar a posição do usuário a cada instante de tempo. Podemos citar algumas propriedades interessantes da estrutura do modelo HMM hierárquico em relação a abordagem clássica de modelos de Markov ocultos [20]: (i) diminuição do número total de parâmetros a serem estimados, reduzindo assim a complexidade da fase de treinamento; (ii) capturar as dependências de curto prazo através da cadeia de Markov discreta que governa os estados ocultos do modelo.

Capítulo 4

Modelo Proposto

Neste capítulo descrevemos o modelo proposto para representar o comportamento interativo de usuários acessando um servidor multimídia para ensino a distância.

Primeiramente, na Seção 4.1 apresentamos uma visão geral do modelo proposto, antes de apresentá-lo em maiores detalhes na Seção 4.2. Na Seção 4.3 apresentamos a metodologia utilizada para geração de carga sintética, a partir de cargas reais, usando o modelo proposto.

4.1 Visão geral

Neste trabalho estudamos o comportamento de usuários acessando um servidor de ensino a distância **em operação** com o objetivo de criar um modelo para geração de carga sintética. Nosso modelo é baseado em *logs* de ações interativas de alunos acessando as aulas no servidor multimídia do CEDERJ. Vale ressaltar que nosso modelo é genérico, podendo se adequar a outros sistemas multimídia. Maiores detalhes sobre o sistema multimídia usado no CEDERJ e dos *logs* de interatividade dos alunos serão apresentados na Seção 5.1.

O modelo proposto é um modelo HMM hierárquico inspirado no trabalho de [20],

descrito em maiores detalhes na Seção 2.2. Neste trabalho, realizamos duas adaptações no modelo proposto em [20] para utilizá-lo para caracterizar o comportamento de usuários interativos. Em nosso modelo, a cadeia de Markov oculta governa a dinâmica de uma sessão do usuário. Geralmente, em sistemas educacionais, as aulas são compostas de vários *slides*. Optamos então por usar os estados ocultos para capturar a dependência de curto prazo das ações do usuário no contexto de um *slide*. Sendo assim, dentro de um estado oculto temos que representar as possíveis ações interativas do usuário. Para isso, adaptamos o modelo para permitir que, dentro de cada estado oculto, esta dinâmica de curto prazo fosse capturada por uma cadeia de Markov discreta qualquer. Como dentro de cada *slide*, o número de ações interativas realizadas pelo usuário é variável, a segunda modificação no modelo permitiu que o número de símbolos emitidos em cada estado oculto pudesse ser variável ao invés de uma constante, como em [20].

4.2 Definição do modelo

Nesta seção apresentamos as adaptações realizadas no modelo original de [20]. A notação adotada segue como a do Capítulo 2. A primeira adaptação realizada no modelo HMM hierárquico, foi permitir que dinâmica de curto prazo fosse capturada por uma cadeia de Markov discreta qualquer, e não apenas por um modelo de Gilbert. A Figura 4.1 ilustra o modelo HMM hierárquico proposto neste trabalho.

Como citado na Seção 2.2, os passos necessários para a adaptação do modelo são semelhantes aos realizados no trabalho original de [20].

As transições entre estados ocultos ocorrem apenas a cada S observações, sendo assim as observações estão segmentadas em conjuntos de tamanho S. Mais especificamente, o símbolo x_t denota um vetor de medições, $[x_{t,1}, \ldots, x_{t,S}]$, representando o resultado para cada uma das observações no t-ésimo grupo. Seja portanto X_t, o vetor das variáveis, $[X_{t,1}, \ldots, X_{t,S}]$.

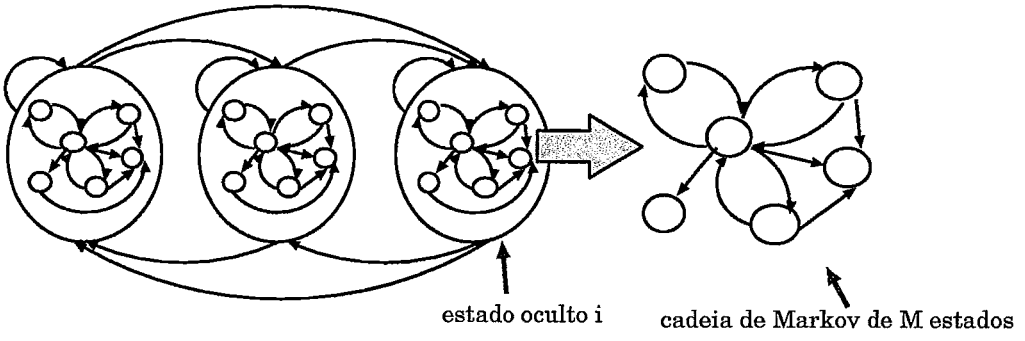


Figura 4.1: Exemplo de um modelo HMM hierárquico

Primeiramente, definiremos os parâmetros da cadeia de Markov discreta que governa os estados ocultos. Sejam $\mathbf{r}$ e $\mathbf{p}$, os parâmetros da cadeia de Markov de M estados dentro de cada estado oculto, i. A Figura 4.2 ilustra a cadeia de Markov de M estados e seus respectivos parâmetros.

$$\begin{aligned} r_{ij} &= P(X_{t,1} = j | Y_t = i), \quad 1 \leq t \leq T, \quad j \in \{1, 2, \ldots, M\} \\ p_{ijk} &= P(X_{t,s} = k | X_{t,s-1} = j, Y_t = i), \quad 1 \leq t \leq T, \quad 1 \leq s \leq S, \quad j, k \in \{1, 2, \ldots, M\} \end{aligned} \tag{4.1}$$

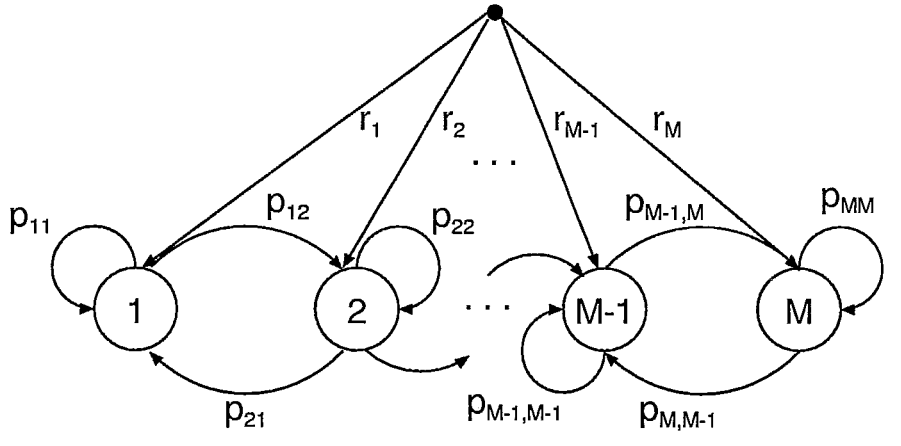


Figura 4.2: Parâmetros da cadeia de Markov de M estados para cada estado oculto do modelo proposto

Para cada grupo de observações, $\mathbf{x}_t$, é suficiente manter o registro apenas das

seguintes estatísticas:

$$x_{t,1} = \text{primeira observação no grupo } t \tag{4.2a}$$

$$S_t^{ij} = \text{número de transições do estado } i \text{ para o } j \text{ em } \mathbf{x}_t, \text{ onde } i, j \in \{1, 2, \ldots, M\} \tag{4.2b}$$

onde, para S_t^{ij}, é válida a restrição:

$$\sum_{i=1}^{M} \sum_{j=1}^{M} S_t^{ij} = S - 1, \quad 1 \leq t \leq T. \tag{4.3}$$

Dada uma instância de $\mathbf{x}_t$, estamos interessados em computar a probabilidade do evento $X_t = \mathbf{x}_t$, dado o estado oculto no t-ésimo grupo, y_t. Usando as estatísticas definidas acima, reescrevemos o parâmetro b_{ij}, que governa o processo de observações, dada pela Equação (2.3) em função dos parâmetros $\mathbf{r}$ e $\mathbf{p}$, definidos em (4.1). Com a adoção da cadeia de Markov discreta para governar os estados ocultos, restringimos o processo de observações dado por $b_{y_t,\mathbf{x}_t}$, de maneira que esta será a probabilidade de iniciar a cadeia de Markov discreta no estado $k \in \{1, 2, \ldots, M\}$ e realizar S transições entre estados $i, j \in \{1, 2, \ldots, M\}$, dado que está no estado oculto y_t.

$$b_{y_t,\mathbf{x}_t} = r_{y_t,k} \prod_{i=1}^{M} \prod_{j=1}^{M} (p_{y_t,i,j})^{S_t^{ij}}, \quad \text{se } x_{t,1} = k, \quad k \in \{1, 2, \ldots, M\} \tag{4.4}$$

Procedemos calculando as estatísticas (4.2a) e (4.2b) e usando esses valores em conjunto com a Equação (4.4), no procedimento *forward-backward* [16]. Maiores detalhes sobre o algoritmo *forward-backward* podem ser verificados na Seção 2.1. Estendemos o algoritmo Baum-Welch para adicionar restrições à Equação (4.4). A cada iteração, este algoritmo maximiza a *função auxiliar*, $Q(\lambda|\overline{\lambda})$ dada pela Equação (2.11), em relação a λ e fazendo uso da estimativa atual dos parâmetros, $\overline{\lambda}$. Usando a definição da verossimilhança conjunta, $P(\mathbf{X}, \mathbf{Y}|\lambda)$, esta função pode ser dividida

em três termos independentes, $Q_1(\pi|\overline{\lambda})$, $Q_2(\mathbf{A}|\overline{\lambda})$ e $Q_3(\mathbf{B}|\overline{\lambda})$, dados pelas Equações (2.13). Maximizando cada um destes termos, chegamos às fórmulas de estimação de parâmetros.

De acordo com o Teorema A.2 da Seção A.4 do trabalho de [20], uma vez que restringimos apenas os parâmetros de observação, $\mathbf{B}$, as fórmulas para π e $\mathbf{A}$ permanecerão idênticas àquelas das equações (2.14a) e (2.14b). Sendo assim, podemos restringir a nossa análise apenas à função auxiliar correspondente, $Q_3(\mathbf{B}|\overline{\lambda})$:

$$Q_3(\mathbf{B}|\overline{\lambda}) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{\forall j} \log b_{ij}\mathbb{I}\{\mathbf{x}_t = j\}\gamma_t(i), \quad \text{onde } j \in \{1,2,\ldots,M\}^S \tag{4.5}$$

Na Equação (4.5), utilizamos a notação $\mathbb{I}\{c\}$, para representar a *função indicadora* de uma condição c, que vale 1 quando a condição é satisfeita, ou 0 caso contrário.

Aplicando a definição da probabilidade de observação de um grupo, dada pela Equação (4.4), no termo da função auxiliar correspondente aos parâmetros de observação, dada pela Equação (4.5), temos:

$$\begin{aligned}
Q_3(\mathbf{B}|\overline{\lambda}) &= \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{\forall j} \log(r_{i,m}\prod_{k=1}^{M}\prod_{l=1}^{M}(p_{i,k,l})^{S_t^{kl}})\mathbb{I}\{\mathbf{x}_t = j\}\gamma_t(i), \\
&\quad \text{se } j_{t,1} = m, \text{ onde } m \in \{1,2,\ldots,M\} \\
&= \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{\forall j} \log(r_{i,m}\prod_{k=1}^{M}\prod_{l=1}^{M}(p_{i,k,l})^{S_t^{kl}})\mathbb{I}\{\mathbf{x}_t = j\}\gamma_t(i)\mathbb{I}\{j_{t,1} = m\} \\
&= \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{\forall j} \left[\log r_{i,m} + \log\left(\prod_{k=1}^{M}\prod_{l=1}^{M}(p_{i,k,l})^{S_t^{kl}}\right)\right]\mathbb{I}\{\mathbf{x}_t = j\}\gamma_t(i)\mathbb{I}\{j_{t,1} = m\}
\end{aligned} \tag{4.6}$$

É fácil verificar que podemos dividir a função auxiliar dada (4.6) em dois termos

independentes, cada um em função de um dos parâmetros de observação $\mathbf{r}$ e $\mathbf{p}$. Temos portanto que redefinir λ, a tupla que representa o nosso modelo, como $\lambda = (\pi, \mathbf{A}, \mathbf{r}, \mathbf{p})$.

As expressões que buscamos são os pontos de máximos das seguintes funções, para cada estado oculto i.

$$Q_4(\mathbf{r}_i|\overline{\lambda}) = \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{\forall j} \log r_{i,m}\mathbb{I}\{\mathbf{x}_t = j\}\gamma_t(i)\mathbb{I}\{j_{t,1} = m\} \tag{4.7}$$

$$Q_5(\mathbf{p}_i|\overline{\lambda}) = \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{\forall j} \log\left(\prod_{k=1}^{M}\prod_{l=1}^{M}(p_{i,k,l})^{S_t^{kl}}\right)\mathbb{I}\{\mathbf{x}_t = j\}\gamma_t(i)\mathbb{I}\{j_{t,1} = m\} \tag{4.8}$$

Para efeito dos futuros cálculos, vale atentar para a validade da seguinte igualdade, para toda instância $\mathbf{x}_t$:

$$\sum_{\forall j}\mathbb{I}\{\mathbf{x}_t = j\} = 1, \quad \text{onde } j \in \{1, 2, \ldots, M\}^S$$

Após algumas manipulações algébricas podemos reescrever os sub-problemas (4.7) e (4.8) da seguinte maneira:

$$\begin{aligned}
Q_4(\mathbf{r}_i|\overline{\lambda}) &= \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{\forall j} \log r_{i,m}\mathbb{I}\{\mathbf{x}_t = j\}\gamma_t(i)\mathbb{I}\{j_{t,1} = m\} \\
&= \sum_{m=1}^{M}\log r_{i,m}\sum_{t=1}^{T}\sum_{\forall j}\mathbb{I}\{\mathbf{x}_t = j\}\mathbb{I}\{j_{t,1} = m\}\gamma_t(i) \\
&= \sum_{m=1}^{M}\log r_{i,m}\sum_{t=1}^{T}\sum_{\forall j}\mathbb{I}\{\mathbf{x}_t = j\}\mathbb{I}\{x_{t,1} = m\}\gamma_t(i) \\
&= \sum_{m=1}^{M}\log r_{i,m}\sum_{t=1}^{T}\mathbb{I}\{x_{t,1} = m\}\gamma_t(i)
\end{aligned} \tag{4.9}$$

$$
\begin{aligned}
Q_5(\mathbf{p}_i|\overline{\lambda}) &= \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{\forall j}\log\left(\prod_{k=1}^{M}\prod_{l=1}^{M}(p_{i,k,l})^{S_t^{kl}}\right)\mathbb{I}\{\mathbf{x}_t=j\}\gamma_t(i)\mathbb{I}\{j_{t,1}=m\} \\
&= \sum_{t=1}^{T}\sum_{\forall j}\log\prod_{k=1}^{M}\prod_{l=1}^{M}(p_{i,k,l})^{S_t^{kl}}\mathbb{I}\{\mathbf{x}_t=j\}\gamma_t(i) \\
&= \sum_{t=1}^{T}\log\prod_{k=1}^{M}\prod_{l=1}^{M}(p_{i,k,l})^{S_t^{kl}}\sum_{\forall j}\mathbb{I}\{\mathbf{x}_t=j\}\gamma_t(i) \\
&= \sum_{t=1}^{T}\sum_{k=1}^{M}\sum_{l=1}^{M}S_t^{kl}\log p_{i,k,l}\gamma_t(i) \\
&= \sum_{k=1}^{M}\sum_{l=1}^{M}\log p_{i,k,l}\sum_{t=1}^{T}S_t^{kl}\gamma_t(i)
\end{aligned}
\tag{4.10}
$$

Fixando k na Equação (4.10) para maximizar $\mathbf{p}_{ik}$, i.e. a probabilidade de transição a partir do estado k, dado o estado oculto i, temos:

$$
Q_5(\mathbf{p}_{ik}|\overline{\lambda}) = \sum_{l=1}^{M}\log p_{i,k,l}\sum_{t=1}^{T}S_t^{kl}\gamma_t(i) \tag{4.11}
$$

Resolveremos os sub-problemas (4.9) e (4.11) como problemas de otimização, através da aplicação do Lema 2 do trabalho [9], que pode ser verificado no Apêndice A.1. Como resultado temos as seguintes equações para $\mathbf{r}$ e $\mathbf{p}$:

$$
r_{i,m} = \frac{\sum_{t=1}^{T}\mathbb{I}\{x_{t,1}=m\}\gamma_t(i)}{\sum_{t=1}^{T}\gamma_t(i)} \tag{4.12}
$$

$$
p_{ikl} = \frac{\sum_{t=1}^{T}S_t^{kl}\gamma_t(i)}{\sum_{j=1}^{M}\sum_{t=1}^{T}S_t^{kj}\gamma_t(i)} \tag{4.13}
$$

No modelo proposto em [20], as transições entre os estados ocultos ocorriam a cada S emissões de símbolos. Para a nossa aplicação do modelo, o número de interações que ocorrem dentro de um *slide* é variável. Precisamos adaptar o modelo para o caso geral onde o número de símbolos emitidos em cada estado oculto possa ser variável. Para isso, incluímos um símbolo para marcar a saída de um estado oculto, ou seja, fim do *slide*, que será representado na cadeia de Markov dentro

do estado oculto como um estado absorvente. Agora temos que S, o número de símbolos dentro de um grupo de medições, é uma variável aleatória que depende do grupo t, $1 \leq t \leq T$. Seja Z_t o tamanho do t-ésimo grupo, podemos reescrever a restrição (4.3) da seguinte forma:

$$\sum_{i=1}^{M}\sum_{j=1}^{M} S_t^{ij} = Z_t, 1 \leq t \leq T$$

Esta alteração não invalida os resultados das equações (4.12) e (4.13). Porém deve-se atentar ao fato de que no desenvolvimento realizado a partir das equações (4.9) e (4.11), os valores de j dependem do tamanho do t-ésimo grupo. É fácil verificar que, $j \in \{1, 2, \ldots, M\}^S$ será tal que $S = Z_t$, e não mais uma constante como anteriormente.

Como última etapa para a definição do modelo, apresentamos os símbolos que serão emitidos em cada estado oculto. Cabe ressaltar que cada um destes símbolos representa um estado da cadeia de Markov discreta que governa os estados ocultos. Os símbolos usados neste modelo são: *play*, pausa, salto para frente, salto para trás, próximo *slide* e saída da sessão. Este conjunto foi escolhido baseado na caracterização do comportamento dos usuários do sistema multimídia do CEDERJ feita em [1]. A cadeia de Markov discreta que governa o estado oculto sempre é iniciada no estado *play*. O símbolo *próximo slide* causa uma transição entre os estados ocultos, já que representa o fim de um *slide*. A partir dos estados *salto para frente* e *salto para trás* é possível voltar para o estado de *play* (caso o salto não gere uma mudança de *slide*) ou transicionar para o estado *próximo slide* (caso contrário). Como dito anteriormente, em nosso modelo, as dependências a curto prazo das interações realizadas dentro de um *slide* são capturadas pela cadeia de Markov que governa o estado oculto, já a dinâmica da sessão do usuário é capturada pela cadeia de Markov oculta. A Figura 4.3 ilustra o modelo HMM hierárquico proposto.

Uma vantagem do modelo HMM hierárquico, comparado com a abordagem clássica de HMM é que a estrutura da cadeia dentro do estado oculto não permite que

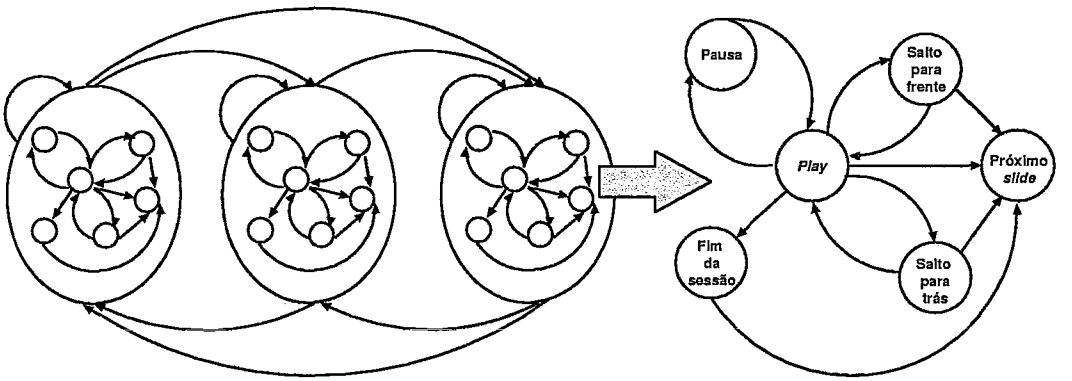


Figura 4.3: Modelo HMM hierárquico

seqüências de ações que notadamente não ocorrem na aplicação real sejam geradas. Um exemplo de seqüência inválida é a ocorrência de duas pausas sem que haja um *play* entre elas. No modelo HMM clássico existe a possibilidade de ocorrência destes tipos de seqüências.

Dada a definição do modelo HMM hierárquico e a determinação da cadeia de Markov discreta que governa os estados ocultos, é necessário ainda especificar a quantidade de estados da cadeia de Markov oculta. Vale ressaltar que nem sempre o ganho em precisão do modelo dado por um número grande de estados ocultos, pode compensar o aumento em complexidade do modelo (o tempo para treinar o modelo também será maior). O valor apropriado depende do tipo de aplicação do modelo e da carga usada para parametrizá-lo, pois em alguns casos, mesmo com poucos estados ocultos é possível se obter bons resultados. No Capítulo 5, utilizamos diversos valores para o número de estados ocultos durante a análise das medidas de interesse para avaliar o ganho em precisão do modelo.

4.3 Geração da carga sintética

Para gerar carga sintética com o modelo proposto, usamos um conjunto de *logs* reais para treiná-lo. Após a etapa de treinamento do modelo podemos usá-lo para simular

uma seqüência de ações interativas realizadas pelo usuário. Esta seqüência de ações é composta pelos símbolos *play*, pausa, salto para frente, salto para trás, próximo *slide* e fim de sessão. Porém, não estão especificados o instante e a posição no vídeo associados a cada uma das ações interativas. A Figura 4.4 ilustra um trecho de uma sessão de usuário gerado pelo modelo. Neste exemplo, pode-se perceber que o modelo não fornece quanto tempo o vídeo ficou pausado, qual o tamanho do salto realizado nem quanto tempo o vídeo ficou em *play*.

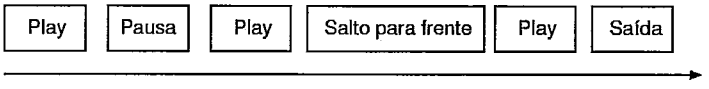


Figura 4.4: Exemplo de uma sessão de usuário gerada pelo modelo

Precisamos então analisar dados dos *logs* reais tais como o tamanho dos saltos, tempo em *play* e tempo em *pause*, para inserir esses dados na geração da carga sintética. Para obter as distribuições de probabilidade dos tempos associados às ações interativas do usuário, procedemos com uma metodologia similar a adotada em [24, 1]. A Figura 4.5 mostra um exemplo de uma sessão de usuário após a obtenção das medidas através das distribuições de probabilidade.

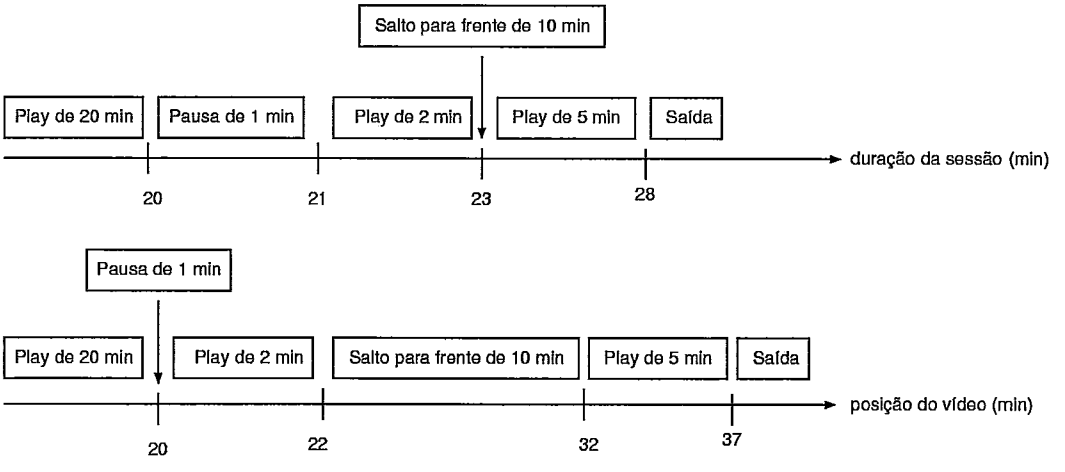


Figura 4.5: Exemplo de uma sessão de usuário gerada pelo modelo após a obtenção das medidas de tempo através das distribuições de probabilidade

Primeiramente, calculamos os parâmetros para diversas famílias de distribuições,

a partir das amostras coletadas para as medidas de interesse, e depois usamos um método para escolher a distribuição mais adequada. Os parâmetros das distribuições de probabilidade são calculados pelo método de estimação por máxima verossimilhança (*maximum likelihood estimation* - MLE) [25, 19] . Este método consiste em selecionar, como estimativa, valores para os parâmetros de forma a maximizar a probabilidade da amostra ocorrer. Sejam as amostras $X_1, X_2, \cdots, X_n$, a função *likelihood* $L(p)$ é a função de probabilidade de massa (pmf) conjunta das amostras e o valor de p, digamos $\widehat{p}$, que maximiza o logaritmo natural de $L(p)$ é o *maximum likelihood estimate* de p. Calculamos o MLE para as seguintes distribuições: Uniforme, Exponencial, Gamma, Weibull, Normal e Lognormal. Para estes cálculos utilizamos o software *MATLAB* [23]. Parametrizamos também a distribuição hiperexponencial com o software *EMpht* [14], que realiza a estimação iterativa dos parâmetros da distribuição através de um algoritmo EM (*Expectation-maximization*), que também é baseado na máxima verossimilhança.

Para determinar o tipo de distribuição que melhor aproxima os dados empíricos de uma dada variável fizemos uma análise visual e quantitativa. A análise visual consiste em plotar os gráficos da distribuição complementar (*complementary cumulative distribution function* - CCDF) com o eixo das ordenadas em escala logarítmica para evidenciar a cauda da distribuição e avaliarmos visualmente aquela que mais se assemelha à distribuição empírica. Entretanto, esta simples análise não deve ser considerada decisiva.

Para complementar a análise visual, comparamos o erro médio quadrático (*mean squared error* - MSE) [19] entre as distribuições empírica e estimada. Além disso, aplicamos o teste de Kolmogorov-Smirnov [25], no qual testamos a hipótese nula de que o conjunto de amostras pertencia a alguma das distribuições escolhidas. Utilizamos um grau de significância de 5%, o que corresponde à probabilidade de rejeitarmos a hipótese nula erroneamente.

Outra análise realizada foi o teste gráfico chamado QQPlot (*Quantile-Quantile Plot*) [13]. Este mostra se dois conjuntos de amostras vêm de uma mesma população,

ou seja, se possuem distribuições provenientes de uma mesma família. Inicialmente são calculados os quantiles (fração de amostras menores que um dado valor) usando dois conjuntos de amostras: o conjunto de amostras empíricas e o de amostras geradas segundo uma distribuição escolhida. No QQPlot são representados os valores dos quantiles obtidos para ambos os conjuntos de amostras. Se os dois conjuntos de amostras são provenientes de uma mesma distribuição, os pontos do gráfico devem formar, aproximadamente, uma reta com inclinação de 45 graus.

Através destes métodos, que permitem a análise visual da cauda das distribuições (gráficos da distribuição complementar), calcular a distância quadrática média entre as curvas da distribuição (MSE), computar os pontos mais distantes entre elas (teste de Kolmogorov-Smirnov), e permitir avaliar se dois conjuntos de amostras pertencem a uma mesma distribuição (QQPlot), acreditamos que podemos obter resultados bastantes confiáveis.

Uma questão surge quando as métricas possuem correlação com a posição do vídeo. Um exemplo é o tamanho de um salto, no qual o seu ponto de destino não deve ultrapassar os limites do vídeo. A princípio seria necessário obter uma distribuição para cada intervalo do vídeo. No entanto, assim como em [24] e [1], preferimos testar a hipótese de utilizarmos uma única distribuição, a partir de todas as amostras.

Sendo assim, geramos amostras a partir de uma única distribuição e, caso a amostra sorteada ultrapasse os limites do vídeo sugerimos três abordagens distintas. A primeira consiste em truncar o seu valor nos limites do vídeo. A segunda, que denominamos reestimação, consiste em realizar sorteios consecutivos até que uma amostra que não ultrapasse os limites do vídeo seja gerada. A terceira abordagem consiste em descartar as sessões onde isto ocorra.

Em suma, determinamos as distribuições segundo os métodos descritos e geramos as cargas utilizando cada uma das três abordagens sugeridas. Para escolher dentre as três cargas geradas usamos os métodos já apresentados, além de comparar métricas

das cargas sintéticas em relação à carga real, tais como como número médio de interações por sessão, as distribuições dos tempos em *play* e em pausa, o número médio

Capítulo 5

Validação e análise comparativa

Neste capítulo primeiramente apresentamos os sistemas educacionais que disponibilizaram os *logs* de interatividade de usuários utilizados para parametrizar os modelos. Em seguida, realizamos uma validação do modelo proposto, comparando o seu desempenho com a abordagem clássica de HMM [16, 24]. Posteriormente, verificamos se a carga sintética gerada pelo modelo proposto e pelo modelo probabilístico apresentado em [17] são estatisticamente similares à carga real. Além disso, comparamos o impacto gerado por ambas as cargas usando um simulador de um servidor multimídia desenvolvido em [5].

5.1 Logs dos sistemas utilizados

Nesta seção apresentamos maiores detalhes sobre os sistemas multimídia CEDERJ e MANIC, e os *logs* de interatividade de usuários utilizados para parametrizar os modelos de geração de carga sintética.

5.1.1 Sistema CEDERJ

Neste trabalho são utilizados *logs* com registros de ações e acessos dos usuários do sistema multimídia do Consórcio CEDERJ (Centro de Educação Superior a Distância do Rio de Janeiro). Este projeto, gerenciado pelo Governo do Estado do Rio de Janeiro, visa possibilitar o acesso à educação, de forma semi-presencial, de alunos de cidades do interior do estado. Foram estudados *logs* de alunos acessando as aulas do curso de graduação de Tecnologia em Sistemas de Computação, elaborado em parceria entre a UFRJ (DCC/IM e PESC/COPPE) e a UFF (Instituto de Computação). Este curso de graduação foi iniciado no primeiro semestre de 2005 e conta atualmente com mais de 1000 alunos matriculados. Para assistir às aulas, os alunos visitam o pólo onde estão matriculados, acessam a plataforma educacional do CEDERJ e, a partir desta, se conectam no servidor RIO para acessar a aula desejada. Consideramos como uma sessão, à visualização de uma aula do curso. Cada interação do aluno com a aula é gravada em um arquivo e, quando a sessão é encerrada, este arquivo é armazenado no servidor do pólo.

O sistema que armazena, gerencia e disponibiliza o conteúdo do curso foi desenvolvido pelo Laboratório LAND da COPPE/UFRJ, a partir de um protótipo inicial projetado em parceria com a UCLA e a UFMG. O servidor multimídia RIO (Random I/O System) [12] é um sistema de armazenamento multimídia universal que usa alocação aleatória e replicação de blocos. Sendo um servidor universal, o RIO suporta vários tipos de mídias: vídeo, áudio, texto, imagem, além de ser capaz de gerenciar aplicações com ou sem restrição de tempo. Para o servidor, todos os tipos de mídias são chamados de objetos e são armazenados da mesma forma. Os objetos são divididos em blocos de dados e estes são aleatoriamente armazenados. O Servidor RIO é composto por um servidor principal que gerencia os pedidos dos clientes e os repassa a um ou mais servidores de armazenamento que enviam diretamente ao cliente os dados solicitados, não sobrecarregando o servidor principal. O servidor principal e os de armazenamento não precisam estar localizados na mesma máquina, permitindo uma arquitetura totalmente distribuída, com os com-

ponentes em localidades distintas interligadas por rede. Para acessar o conteúdo armazenado no servidor, os usuários utilizam um software cliente (desenvolvido pelo LAND/UFRJ) que possibilita a interação do aluno com o conteúdo que está sendo apresentado. A interface do cliente pode ser vista na Figura 5.1.

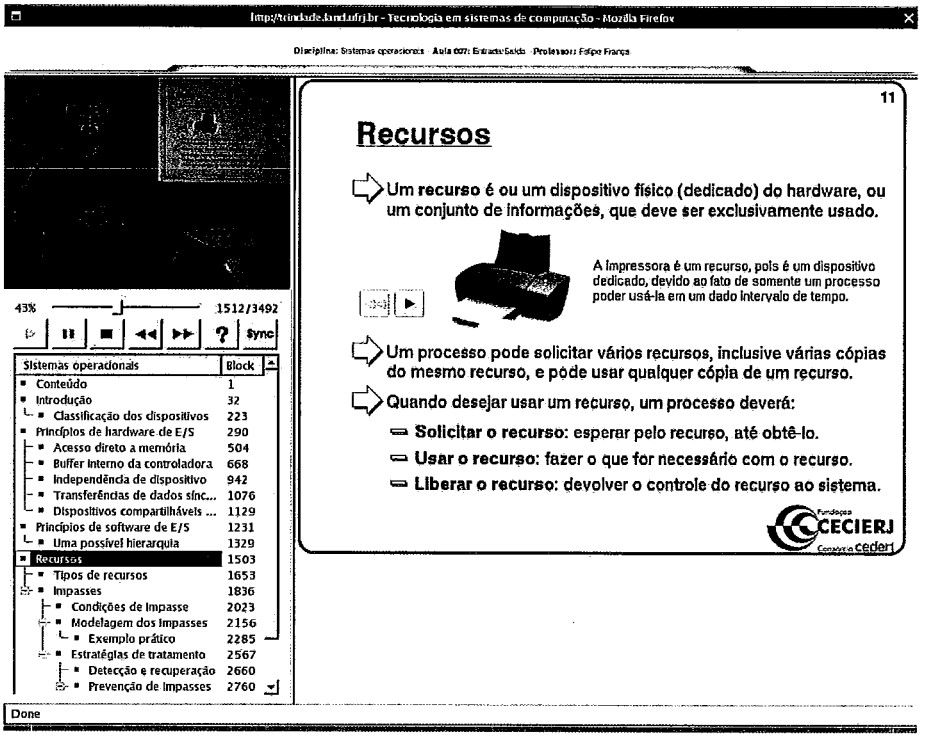


Figura 5.1: Cliente do servidor RIO

As aulas usadas no projeto CEDERJ são compostas por vídeo e *slides* sincronizados e a transmissão se dá sob demanda em tempo real. Também está disponível a todo momento para o aluno um índice com os tópicos apresentados na aula, através do qual o aluno pode selecionar o tópico que deseja ver da aula. Através do cliente os usuários podem paralisar a exibição da aula (pressionar *pause*); saltar para outro ponto da aula através dos controles: *fast forward*, *fast rewind*, arrastar a barra de progresso ou clicar no índice de um *slide*; e parar (através do comando *stop*) a exibição do conteúdo a qualquer instante. Quando o usuário deseja encerrar a sessão ele clica no comando *quit*. Para este trabalho tínhamos disponíveis um conjunto de mais de 11000 *logs*, sendo aproximadamente 5100 deles com duração da sessão maior

do que 5 minutos. Os *logs* correspondem as aulas de todos os cursos ministrados entre o primeiro período de 2005 e o primeiro período de 2007.

Em [1], foram analisadas características de interatividade dos usuários do sistema multimídia RIO, no ambiente CEDERJ. Naquele trabalho pode ser encontrada uma descrição detalhada de diversas medidas de interatividade. O trabalho revelou que o usuários do sistema CEDERJ mostraram ser bem mais interativos do que os estudados em trabalhos anteriores, o que torna muito interessante o uso de sua carga para parametrizar o nosso modelo. Assim como no trabalho de [1], apenas *logs* de sessões de alunos acessando o sistema, com duração maior do que 5 minutos serão utilizados em nosso trabalho.

5.1.2 Sistema MANIC

Neste trabalho também utilizamos *logs* de comportamento de usuário acessando vídeos educativos, coletados do sistema MANIC (*Multimedia Asynchronous Networked Individualized Courseware*) [22]. Duas versões do sistema MANIC foram desenvolvidas. Na primeira o conteúdo das aulas fica pré-armazenado em um servidor multimídia e é enviado pela Internet até os usuários. Em [8], os *logs* foram obtidos deste sistema, quando ele possuía integração apenas entre os *slides* e o áudio. Versões mais recentes possuem integração também com vídeo. A outra versão consiste no CD-MANIC [3]. Neste caso o conteúdo das aulas - *slides*, áudio e vídeo - está em um CD entregue ao usuário. Os registros dos usuários acessando as aulas são armazenados localmente e depois enviados pela rede para um servidor. Os *logs* utilizados neste trabalho foram obtidos a partir do CD-MANIC. O sistema possui diversos comandos como *pause*, *index*, *fast forward*, *rewind*, que possibilitam a interação do aluno com a aula. Os *logs* são gerados a cada interação do usuário, ou quando há um evento de mudança de tópico, e contém informações tais como, a data, o tipo de ação e o *slide* em que o aluno se encontra. O final de uma sessão pode ser identificado por uma ação explícita de saída do usuário.

Assim como no trabalho [24] consideramos que o aluno também pode sair da sessão corrente ao: solicitar um *slide* de outra aula, assistir até o final do último *slide* da aula em questão ou ultrapassar 30 minutos (um *Session Gap Threshold* - SGT [22]) sem fazer interação alguma. No nosso trabalho utilizamos registros de usuários acessando o curso *Computer Networking* ministrado pelo Professor Jim Kurose da Universidade de Massachusetts. Tínhamos disponíveis um total de 1100 *logs*, sendo 980 com duração da sessão maior do que 5 minutos.

5.2 Validação

Neste capítulo apresentamos a validação do modelo HMM hierárquico proposto. Para isso, realizamos uma comparação entre nosso modelo e uma variação do modelo HMM proposto em [24]. Desta maneira pretendemos avaliar o ganho em usar o modelo HMM hierárquico com relação a abordagem clássica de HMM. Neste ponto não comparamos com outros modelos Markovianos pois estes possuem capacidade finita de memória, em contraste com modelos de Markov ocultos.

Para parametrizar o modelo, utilizamos *logs* reais de aulas do curso de graduação de Tecnologia da Computação do CEDERJ. Filtros foram aplicados para que apenas fossem considerados *logs* de alunos com duração da sessão maior do que 5 minutos [1].

O modelo originalmente proposto em [24] possui os seguintes símbolos observáveis: próximo *slide*, pausa, salto de 1 *slide* para frente, salto de 1 *slide* para trás, salto de 2 *slides* para frente, salto de 2 *slides* para trás e final da sessão. Analisando a carga do CEDERJ, observamos que saltos dentro de um *slide* são muito freqüentes. Temos que 24% dos saltos para frente e 37% dos saltos para trás são deste tipo. É de nosso interesse que estes saltos ocorram também nos *logs* sintéticos gerados pelo modelo HMM, portanto incluímos um novo símbolo para representar saltos dentro dos limites de um *slide*. A Figura 5.2 ilustra a adaptação realizada no modelo HMM proposto em [24], onde o símbolo *salto 0* representa um salto dentro do mesmo *slide*.

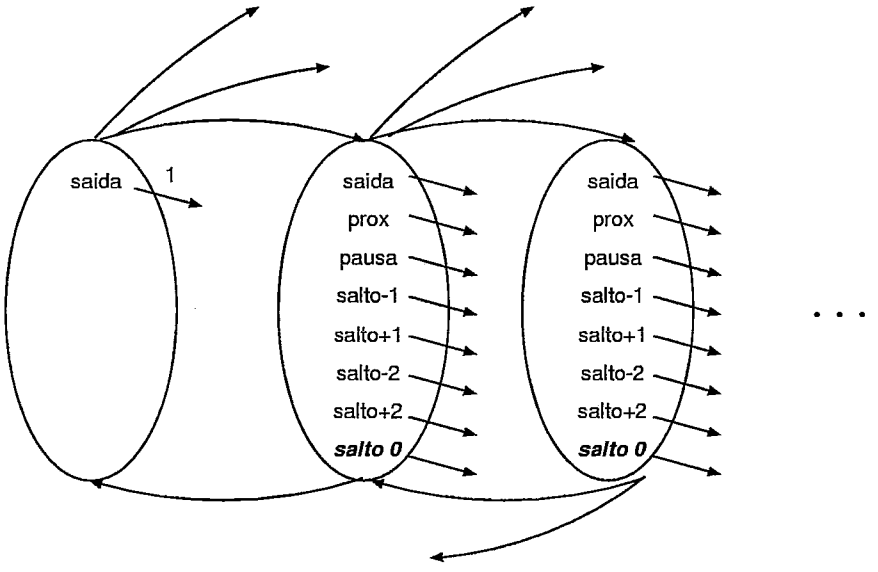


Figura 5.2: Modelo HMM proposto em [24] com a inclusão do símbolo salto dentro dos limites de um *slide* (salto 0)

Para realizar a validação, separamos os *logs* disponíveis do CEDERJ em 2 conjuntos. O primeiro foi utilizado para treinar os modelos, composto de 3580 *logs* e o outro, com os 1559 restantes, para calcular a probabilidade dos mesmos terem sido gerados pelos modelos previamente treinados. Realizamos 20 treinamentos independentes para cada modelo e escolhemos aquele cujo o logaritmo da medida de verossimilhança, $\log P(\mathbf{X}|\lambda)$, fosse maior. Posteriormente, calculamos a probabilidade dos *logs* do segundo conjunto terem sido gerados por cada um dos modelos. Realizamos este procedimento variando a quantidade de estados ocultos nos valores entre 2 e 10. A Figura 5.3 mostra a comparação entre os modelos. Claramente o modelo HMM hierárquico tem melhor desempenho, mesmo para poucos estados ocultos.

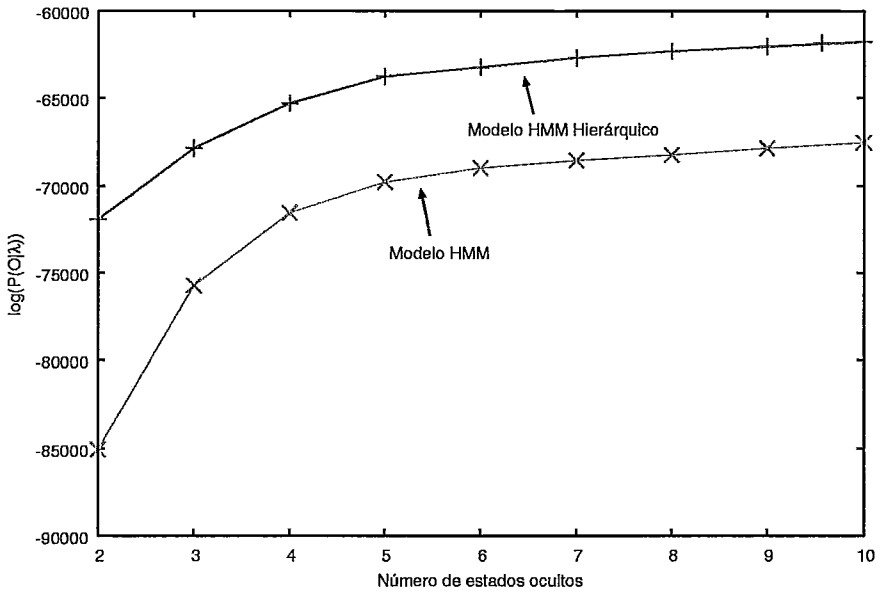


Figura 5.3: Probabilidade das observações terem sido geradas pelo modelo para cada número de estados ocultos

5.3 Análise comparativa

Nesta seção apresentamos uma análise comparativa do modelo HMM hierárquico proposto. Avaliamos a acurácia do modelo quando este é utilizado para dimensionar um servidor que implementa técnicas de compartilhamento de banda. Utilizamos para parametrizar o modelo *logs* reais de aulas do curso de graduação de Tecnologia da Computação do CEDERJ e *logs* reais do sistema MANIC, apresentados na Seção 5.1. Comparamos o nosso modelo com o modelo proposto em [17], descrito em maiores detalhes na Seção 3.2. Não comparamos o modelo proposto com o modelo HMM de [8], pois o mesmo é usado para modelar uma versão do sistema MANIC com apenas áudio sincronizado com transparências (não há vídeo).

A comparação entre o modelo HMM hierárquico e o modelo de [17] foi feita de duas formas: (i) cálculo das estatísticas das cargas sintéticas geradas pelos modelos e comparação com estatísticas da carga real; (ii) comparação da taxa de chegada de requisições e da banda média no servidor obtida através do modelo de [5] quando

este é alimentado pelas cargas sintéticas e real.

O modelo de simulação de um servidor multimídia foi criado usando a ferramenta Tangram-II [6] e consiste de diversos clientes acessando um único objeto de um servidor de vídeo que implementa uma técnica para compartilhar o canal de transmissão do servidor, denominada PIE (*Patching Interativo Eficiente*) [5]. Os clientes executam comandos como avanço, retrocesso e pausa do vídeo, de acordo com os *logs*. Por sua vez, o servidor é responsável pelo envio dos dados solicitados pelos clientes segundo a técnica de compartilhamento de banda implementada.

Os *logs* sintéticos gerados pelo modelo HMM hierárquico foram obtidos através da metodologia descrita na Seção 4.3. O modelo é alimentado com um conjunto de *logs* reais, e posteriormente é usado para simular uma seqüência de ações interativas realizadas pelo usuário. As informações relacionadas ao instante e a posição no vídeo associados a cada uma das ações interativas são calculados através de distribuições que melhor caracterizem os dados reais. A escolha destas distribuições foi realizada da seguinte maneira: um conjunto de distribuições tem os seus parâmetros estimados através do método MLE (*maximum likelihood estimation*). Através de métodos, que permitem a análise visual da cauda das distribuições (gráficos da distribuição complementar), calcular a distância quadrática média entre as curvas da distribuição (MSE), computar os pontos mais distantes entre elas (teste de Kolmogorov-Smirnov), e permitir avaliar se dois conjuntos de amostras pertencem a uma mesma distribuição (QQPlot), escolhemos a distribuição que mais se aproximasse aos dados reais.

Variamos o número de estados ocultos na faixa de valores entre 2 e 20, para também avaliar o ganho em precisão do modelo HMM hierárquico ao usarmos mais estados ocultos. A escolha do número de estados ocultos mais adequado para cada tipo de carga, foi feita através da análise de algumas métricas de interatividade.

Na Seção 4.3 comentamos que existe um problema a ser contornado na geração dos *logs* sintéticos. Aquele que diz respeito a ultrapassar os limites do vídeo no

momento em que inserimos as informações de tempo e posição no vídeo, para cada ação interativa. Geramos *logs* sintéticos adotando cada uma das abordagens anteriormente citadas. São elas: descartar a seqüência de ações referente a sessão onde ocorreu o problema de limite, realizar sorteios consecutivos de novas amostras até que seja encontrada uma que não ultrapasse os limites do vídeo ou truncar a amostra nos limites do vídeo. Os resultados obtidos neste trabalho com a abordagem de descartar, para todos os conjuntos de *logs* utilizados não foram tão satisfatórios quanto para as demais abordagens. Por isso, preferimos omitir esses dados e apenas mostrar os resultados para as abordagens de truncar e reestimar.

Geramos carga sintética com o modelo probabilístico [17] como descrito no próprio trabalho.

Por fim, realizamos simulações usando o modelo [5] usando as cargas sintéticas geradas pelos modelos, além da própria carga real. Também variamos os valores para a taxa de chegada de clientes e, não houveram alterações significativas nos resultados.

Para comparar os resultados, calculamos a taxa de chegada de requisições e a banda média no servidor das cargas real e sintética. A banda é medida em número de canais de transmissão de dados simultâneos em uso no sistema.

Com relação às métricas de interatividade calculadas para as cargas sintéticas e real, computamos o número médio de requisições, número de interações de cada tipo e tamanho médio do segmento. Comparamos também a distribuição obtida para o tempo em *play* e tempo em *pause*, para as cargas real e sintéticas.

Primeiramente apresentamos os resultados obtidos com os modelos ao serem alimentados com os *logs* do CEDERJ e depois para os *logs* do sistema MANIC.

5.3.1 *Logs* CEDERJ

Dispúnhamos de um conjunto de aproximadamente 11000 *logs* reais de comportamento de usuários utilizando o sistema CEDERJ. Primeiramente analisamos os *logs* das duas aulas mais populares e em seguida, agrupamos *logs* de aulas diferentes mas que possuíssem sua duração dentro de um intervalo pré-determinado, que denominamos *logs* com duração parecida. Desta maneira, analisamos um conjunto maior de *logs* do que os disponíveis para cada aula individualmente.

Realizamos simulações com as cargas sintéticas, além da carga real. Cada simulação foi feita com o mesmo número de sessões que o da carga real. Assim como em outros trabalhos [17, 5], as chegadas eram determinadas por um processo de Poisson. Em nosso trabalho, utilizamos uma taxa de aproximadamente 3 sessões por minuto. O processo de chegadas ainda não foi estudado utilizando os dados dos *logs* reais pois este dado não estava disponível em uma parte dos *logs*, e o conjunto restante não totalizava uma quantidade suficiente para realizar este tipo de análise.

Aulas mais populares

Dentre os *logs* disponíveis do CEDERJ, selecionamos as aulas mais populares - as aulas 7 e 8 do curso de Introdução à Informática. A aula 7 possui duração de aproximadamente 2 horas e tínhamos disponíveis um conjunto de 130 *logs*. A aula 8 possui duração de 2 horas e 50 minutos, com um conjunto de 90 *logs*. As distribuições de probabilidade para as métricas de interatividade da carga real destas aulas pode ser verificada nas Tabelas 5.1 e 5.2.

Treinamos o modelo HMM hierárquico para diferentes valores de estados ocultos. A Figura 5.4 apresenta o gráfico do logaritmo da medida de verossimilhança para a aula 7. É possível verificar que após 10 estados ocultos, o ganho em precisão do modelo cresce mais lentamente. É importante avaliar se este ganho compensa o aumento em complexidade e de tempo de treinamento do modelo. Neste trabalho, optamos por realizar a escolha do número de estados ocultos mais adequado para o

Tabela 5.1: Aula 7 do curso de Introdução à Informática do CEDERJ: Distribuições de probabilidade para as métricas da carga real

	Distribuição de probabilidade	**Média (em s)**
Play	hiperexponencial (6 fases)	91.9
Pausa	hiperexponencial (2 fases)	142.1
Salto para frente para fora do *slide*	lognormal	470.7
Salto para frente dentro do mesmo slide	exponencial	25.7
Salto para trás para fora do *slide*	hiperexponencial (2 fases)	658.2
Salto para trás dentro do mesmo slide	lognormal	26.1

Tabela 5.2: Aula 8 do curso de Introdução à Informática do CEDERJ: Distribuições de probabilidade para as métricas da carga real

	Distribuição de probabilidade	**Média (em s)**
Play	hiperexponencial (3 fases)	65.0
Pausa	hiperexponencial (3 fases)	151.1
Salto para frente para fora do *slide*	lognormal	465.4
Salto para frente dentro do mesmo slide	exponencial	43.0
Salto para trás para fora do *slide*	hiperexponencial (3 fases)	498.9
Salto para trás dentro do mesmo slide	lognormal	29.9

modelo, através da análise de algumas métricas de interatividade da carga sintética em comparação com a real.

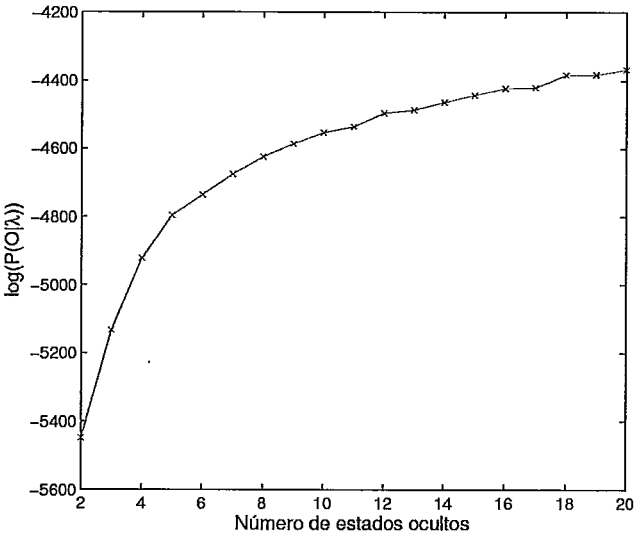


Figura 5.4: Logaritmo da verossimilhança para cada número de estados ocultos para a aula 7 do curso de Introdução à Informática do CEDERJ

Além de variar o número de estados ocultos, geramos a carga sintética utilizando as diferentes abordagens descritas na Seção 4.3, para resolver a questão das métricas correlacionadas com a posição do vídeo. Para simplificar a comparação com o modelo de [17], escolhemos o melhor resultado dentre as diferentes abordagens e limitamos nossa análise a apenas 2 valores de estados ocultos, 4 e 20 estados. As Tabelas 5.3 e 5.4, mostram algumas estatísticas das cargas sintéticas e real com relação a interatividade dos alunos. As estatísticas referentes ao número médio de saltos foi calculada incluindo os saltos dentro do mesmo *slide* e saltos para fora do *slide*. A técnica de reestimar apresentou resultados melhores e não houve diferença significativa entre as métricas geradas pelos modelos de 4 e 20 estados. Para a carga do modelo HMM hierárquico da Aula 7, selecionamos a abordagem de reestimar para 4 estados ocultos. Já para a Aula 8, escolhemos a abordagem de reestimar para 20 estados ocultos. Esta escolha foi feita através da análise das métricas de

interatividade, e não somente através do ganho do logaritmo da verossimilhança. As Tabelas 5.5 e 5.6 apresentam estas mesmas métricas computadas para a carga gerada pelo modelo de [17].

Tabela 5.3: Aula 7 do curso de Introdução à Informática do CEDERJ: Comparação entre métricas geradas pelo modelo HMM hierárquico

	Carga real	**4 est**		**20 est**	
		reestima	**trunca**	**reestima**	**trunca**
Número médio de interações	17.1	16.8	14.1	17.4	14.6
Número médio de pausas	3.4	3.3	2.9	3.8	3.4
Número médio de saltos para frente	9.9	9.7	7.9	8.9	7.1
Número médio de saltos para trás	3.7	3.7	3.2	4.6	4.1
Tempo médio em *play* (em s)	91.9	90.0	101.1	93.4	102.3
Tempo médio em pausa (em s)	142.1	128.4	128.6	123.7	125.1
Tamanho médio do salto para frente (em s)	378.7	425.8	515.0	437.9	596.6
Tamanho médio do salto para trás (em s)	385.0	275.7	291.6	254.9	236.7

Alimentamos o modelo de simulação de [5] com as cargas sintéticas de ambos os modelos, além da carga real. As Figuras 5.5 e 5.6 mostram a comparação entre diversas métricas da carga real e das cargas sintéticas. Pode-se verificar que a carga sintética gerada pelo modelo HMM hierárquico mostrou-se similar à carga real para ambas as aulas, com uma boa estimativa para o número médio de interações, número médio de pausas, número médio de saltos, distribuições dos tempos em *play* e pausa

Tabela 5.4: Aula 8 do curso de Introdução à Informática do CEDERJ: Comparação entre métricas geradas pelo modelo HMM hierárquico

	Carga real	**4 est**		**20 est**	
		reestima	**trunca**	**reestima**	**trunca**
Número médio de interações	24.8	23.2	20.7	24.1	21.5
Número médio de pausas	4.1	4.2	3.7	3.7	3.7
Número médio de saltos para frente	15.1	14.0	12.3	14.4	12.4
Número médio de saltos para trás	5.5	4.9	4.6	5.8	5.3
Tempo médio em *play* (em s)	65.0	64.3	77.8	66.3	70.0
Tempo médio em pausa (em s)	151.1	120.6	134.7	135.6	142.0
Tamanho médio do salto para frente (em s)	332.9	463.2	461.1	434.1	491.7
Tamanho médio do salto para trás (em s)	339.1	333.3	300.4	287.4	315.4

Tabela 5.5: Aula 7 do curso de Introdução à Informática do CEDERJ: Comparação entre métricas geradas pelo modelo de [17]

	Carga real	**Carga HMM Hierárquico**	**Carga de [17]**
Número médio de interações	17.1	16.8	9.9
Número médio de pausas	3.4	3.3	5.5
Número médio de saltos para frente	9.9	9.7	3.2
Número médio de saltos para trás	3.7	3.7	1.2
Tempo médio em *play* (em s)	91.9	90.0	144.2
Tempo médio em pausa (em s)	142.1	128.4	227.1
Tamanho médio do salto para frente (em s)	378.7	425.8	465.0
Tamanho médio do salto para trás (em s)	385.0	275.7	427.6

Tabela 5.6: Aula 8 do curso de Introdução à Informática do CEDERJ: Comparação entre métricas geradas pelo modelo de [17]

	Carga real	**Carga HMM Hierárquico**	**Carga de [17]**
Número médio de interações	24.8	24.1	12.4
Número médio de pausas	4.1	3.7	6.6
Número médio de saltos para frente	15.1	14.4	4.8
Número médio de saltos para trás	5.5	5.8	1.1
Tempo médio em *play* (em s)	65.0	66.3	115.1
Tempo médio em pausa (em s)	151.1	135.6	165.8
Tamanho médio do salto para frente (em s)	332.9	434.1	519.4
Tamanho médio do salto para trás (em s)	339.1	287.4	602.3

e para a taxa de chegada de requisições no servidor. Já o modelo de [17], apresentou um número médio de interações sensivelmente menor do que o da carga real. Por outro lado, o número de pausas do modelo é maior do que o de pausas da carga real, enquanto que o número médio de saltos é aproximadamente 2 vezes menor do que o da carga real. Com relação a distribuição dos tempos em *play* e em pausa, o modelo de [17] não conseguiu uma boa aproximação das curvas das distribuições da carga real. Como o modelo de [17] subestimou o número de interações, conseqüentemente a taxa de chegada de requisições ao servidor também foi subestimada.

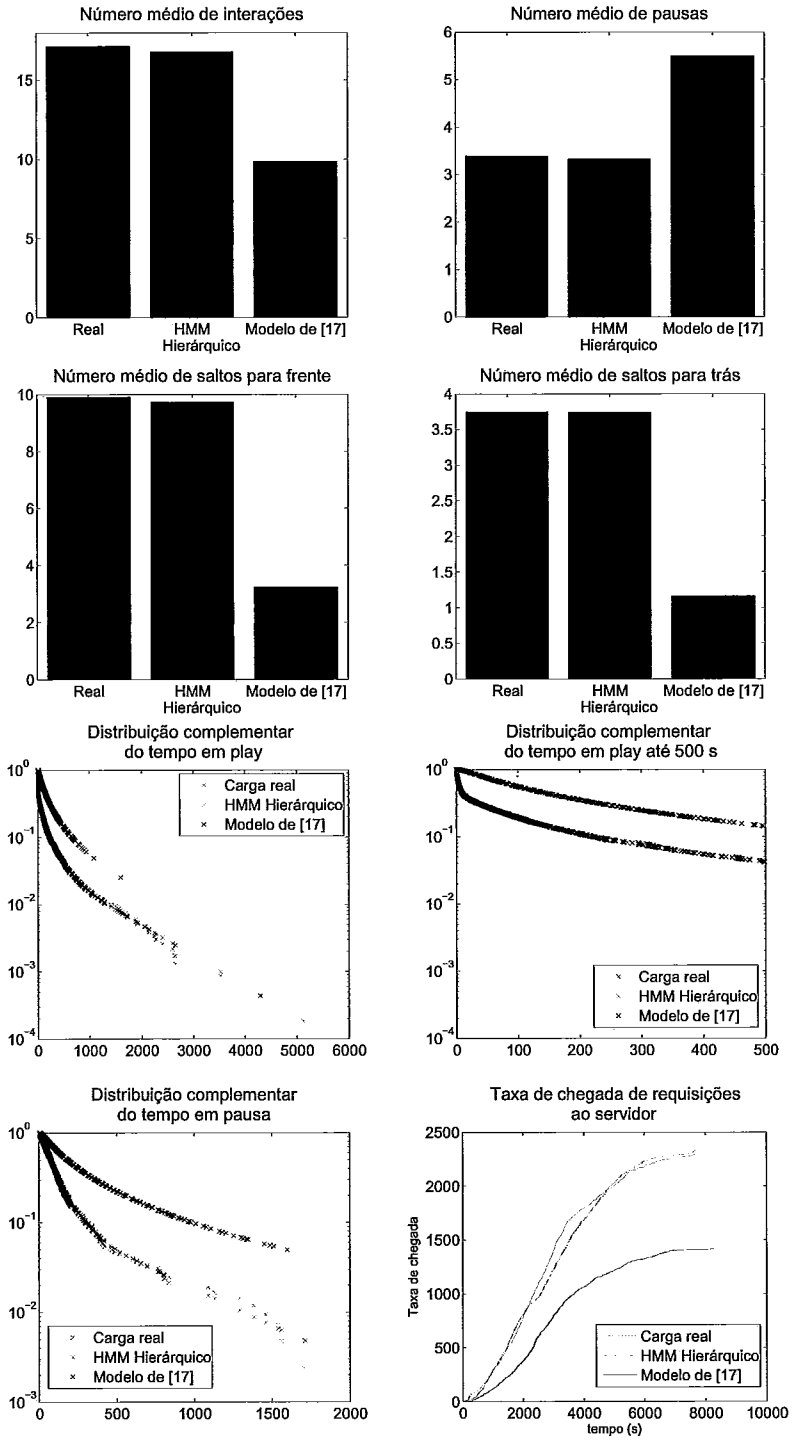


Figura 5.5: Métricas para a aula 7 do curso de Introdução à Informática do CEDERJ

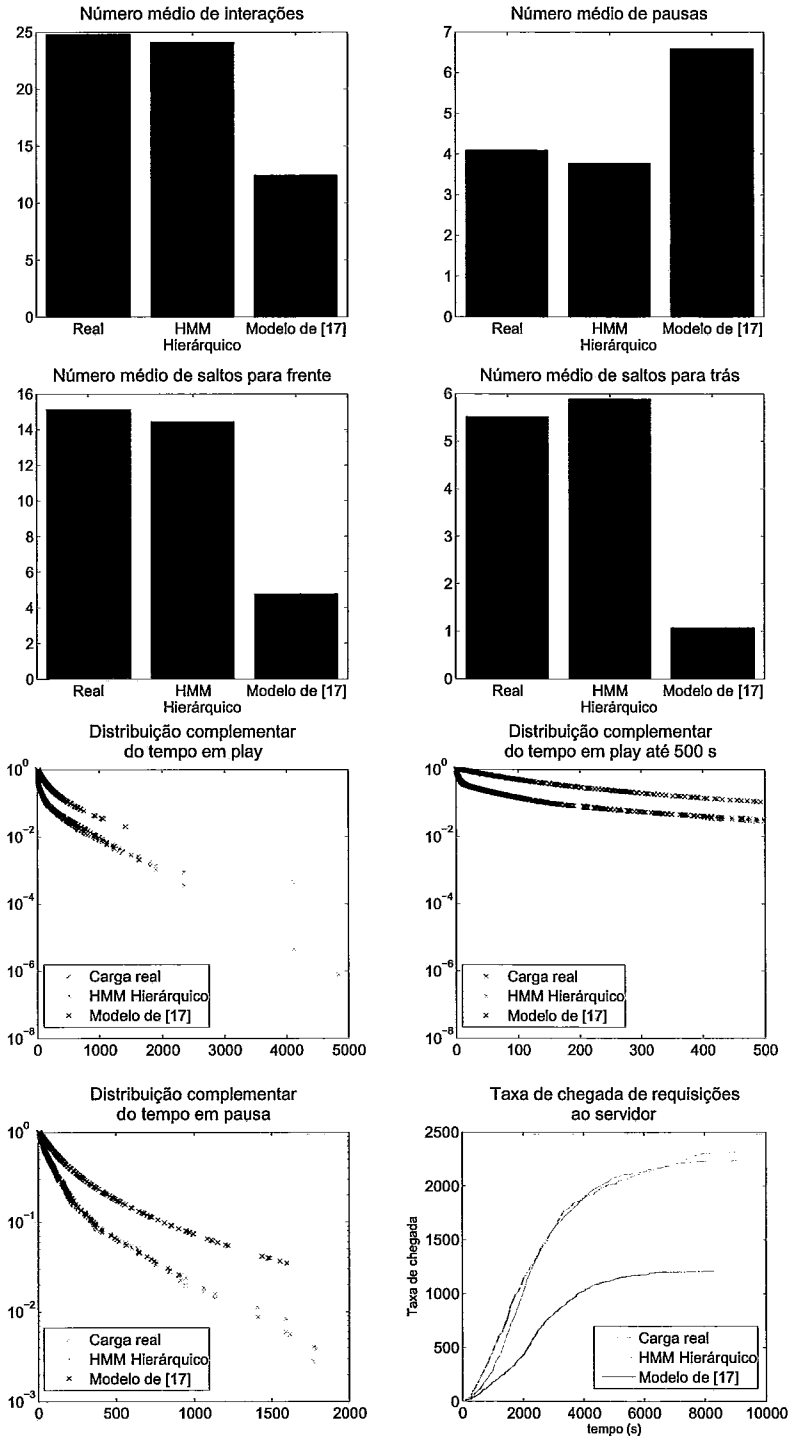


Figura 5.6: Métricas para a aula 8 do curso de Introdução à Informática do CEDERJ

Aulas com duração parecida

Realizamos outra análise com a finalidade de verificar o comportamento do modelo com o aumento de clientes acessando um mesmo objeto no servidor. Como não dispúnhamos de um grande número de sessões para uma mesma aula, agregamos *logs* de sessões de diversas aulas que tivessem sua duração dentro de um determinado intervalo.

Escolhemos um conjunto de 476 *logs*, de mais de 90 aulas diferentes, onde a aula de maior duração possui 57 minutos e a menor, 37. Outro filtro usado neste conjunto, para torná-lo mais homogêneo foi restringir o tamanho da sessão do usuário entre aproximadamente 10 e 20 minutos. A Figura 5.7 mostra o histograma do tamanho das sessões para este conjunto de *logs*.

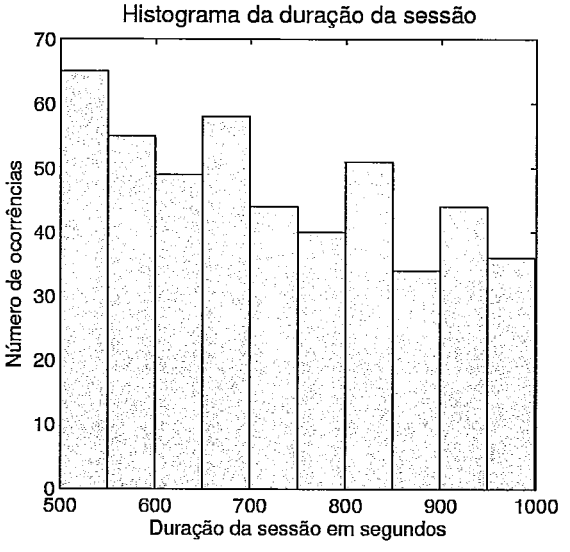


Figura 5.7: Histograma da duração da sessão para o conjunto de 476 *logs* do CEDERJ

As distribuições de probabilidade para as métricas de interatividade da carga real podem ser verificadas na Tabela 5.7.

Assim como para as aulas mais populares, geramos carga sintética com o modelo HMM hierárquico para as abordagens de truncar e reestimar, além de utilizar dois

valores extremos para a quantidade de estados ocultos. A Tabela 5.8 apresenta estes resultados para diferentes métricas. A técnica de reestimar é ligeiramente melhor do que a de truncar e não houve diferença significativa entre as métricas geradas pelos modelos de 4 e 20 estados.

A comparação entre as métricas do modelo HMM hierárquico e o modelo de [17] pode ser verificada através da Tabela 5.9. Para a carga do modelo HMM hierárquico, selecionamos a abordagem de reestimar para 4 estados ocultos.

Alimentamos o modelo de simulação de [5] com as cargas sintéticas e real. A Figura 5.9 apresenta a comparação entre diversas métricas da carga real e das cargas sintéticas. Para este conjunto de *logs*, o modelo HMM hierárquico apresentou uma boa aproximação na geração de carga sintética. O nosso modelo mostrou uma boa estimativa para o número médio de interações, número médio de pausas, número médio de saltos, distribuições dos tempos em *play* e em pausa e para a taxa de chegada de requisições no servidor. Já o modelo de [17] subestimou o número de

Tabela 5.7: Conjunto de 476 *logs* do CEDERJ: Distribuições de probabilidade para as métricas da carga real

	Distribuição de probabilidade	**Média (em s)**
Play	hiperexponencial (3 fases)	41.1
Pausa	hiperexponencial (2 fases)	101.1
Salto para frente para fora do *slide*	lognormal	205.6
Salto para frente dentro do mesmo slide	exponencial	30.8
Salto para trás para fora do *slide*	lognormal	315.6
Salto para trás dentro do mesmo slide	lognormal	27.7

Tabela 5.8: Conjunto de 476 *logs* do CEDERJ: Comparação entre métricas geradas pelo modelo HMM hierárquico

	Carga real	**4 est**		**20 est**	
		reestima	**trunca**	**reestima**	**trunca**
Número médio de interações	13.9	13.1	10.8	12.4	10.1
Número médio de pausas	1.1	1.1	1.0	1.0	1.0
Número médio de saltos para frente	10.7	10.1	8.0	9.4	7.2
Número médio de saltos para trás	2.0	1.9	1.8	2.0	1.8
Tempo médio em *play* (em s)	41.1	44.3	56.5	47.7	59.0
Tempo médio em pausa (em s)	101.1	109.3	101.9	106.6	104.2
Tamanho médio do salto para frente (em s)	169.0	324.7	396.8	331.6	429.1
Tamanho médio do salto para trás (em s)	233.4	203.9	209.7	237.1	222.0

requisições que chegam ao servidor. Este resultado é uma conseqüência das seguintes características do modelo de [17]: o número médio de interações de saltos para frente e saltos para trás da carga real é aproximadamente 3 vezes maior do que o valor encontrado para o modelo; por outro lado, o número médio de pausas do modelo é 3 vezes maior do que as pausas da carga real. Como o modelo de [17] gerou uma carga com menor interatividade, menor será a taxa de chegada de requisições ao servidor (como pode ser verificado na Figura 5.9).

Um outro conjunto de *logs* foi criado para mostrar o desempenho dos modelos

Tabela 5.9: Conjunto de 476 *logs* do CEDERJ: Comparação entre métricas geradas pelo modelo de [17]

	Carga real	**Carga HMM Hierárquico**	**Carga de [17]**
Número médio de interações	13.9	13.1	7.44
Número médio de pausas	1.1	1.1	3.7
Número médio de saltos para frente	10.7	10.1	3.3
Número médio de saltos para trás	2.0	1.9	0.5
Tempo médio em *play* (em s)	41.1	44.3	98.5
Tempo médio em pausa (em s)	101.1	109.3	63.9
Tamanho médio do salto para frente (em s)	169.0	324.7	325.0
Tamanho médio do salto para trás (em s)	233.4	203.9	157.4

quando a carga usada para parametrizá-los não é submetida a uma análise adequada. Este conjunto possuía 290 *logs* com duração da aula entre 52 e 53 minutos. Mesmo com uma diferença pequena entre a duração das aulas, as sessões dos usuários estavam entre 5 e 80 minutos. Uma diferença muito grande entre o tamanho das sessões influencia diretamente no nível de interatividade. A Figura 5.8 mostra o histograma do tamanho das sessões para este conjunto de *logs*. As distribuições de probabilidade para as métricas de interatividade da carga real podem ser verificadas na Tabela 5.10.

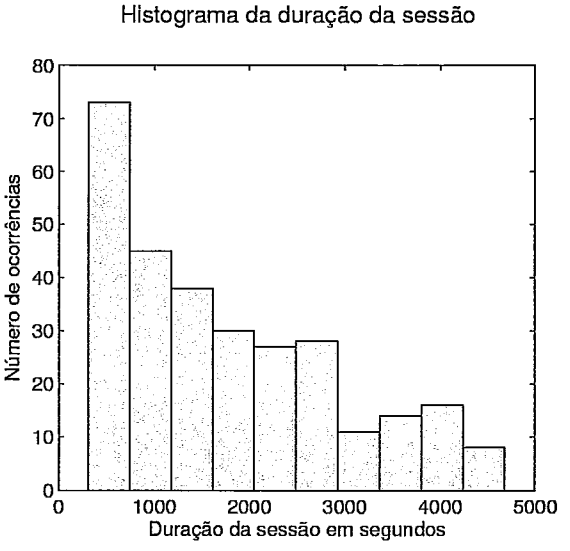


Figura 5.8: Histograma da duração da sessão para o conjunto de 290 *logs* do CEDERJ

As métricas de interatividade obtidas das cargas sintéticas geradas pelo modelo HMM hierárquico, variando o número de estados ocultos e utilizando diferentes abordagens podem ser analisadas na Tabela 5.11. A técnica de reestimar apresentou melhores resultados que a de truncar para os dois valores de estados ocultos. Não houve grande diferença entre os resultados para 4 e 20 estados ocultos.

A comparação entre as métricas do modelo HMM hierárquico e o modelo de [17] pode ser verificada através da Tabela 5.12. Para a carga do modelo HMM hierárquico, selecionamos a abordagem de reestimar com 4 estados ocultos.

Alimentamos o modelo de simulação de [5] com as cargas sintéticas e real. A Figura 5.10 apresenta a comparação entre diversas métricas da carga real e das cargas sintéticas. Para este conjunto, a carga gerada pelo modelo HMM hierárquico não conseguiu uma aproximação tão boa quanto para os conjuntos anteriores. O nosso modelo apresentou um número médio de interações e número médio de saltos para frente 20% inferior ao da carga real. Em contrapartida, apresentou uma boa estimativa para o número médio de pausas, número médio de saltos para trás e para a distribuição do tempo em pausa. Com relação a distribuição do tempo em *play*, nenhuma das distribuições escolhidas conseguiu uma boa aproximação a da carga real. Vale lembrar que as distribuições dos tempos em *play* e em pausa não são uma saída do modelo HMM hierárquico, elas são estimadas segundo a metodologia descrita na Seção 4.3. Sendo assim, é possível que se consiga obter uma melhor aproximação à carga real através da escolha de uma outra distribuição, que não pertença ao conjunto que utilizamos neste trabalho. O modelo de [17] também não apresentou boas estatísticas: o número médio de interações e o número médio de saltos foi menor

Tabela 5.10: Conjunto de 290 *logs* do CEDERJ: Distribuições de probabilidade para as métricas da carga real

	Distribuição de probabilidade	**Média (em s)**
Play	hiperexponencial (3 fases)	71.6
Pausa	hiperexponencial (2 fases)	136.2
Salto para frente para fora do *slide*	lognormal	170.5
Salto para frente dentro do mesmo slide	lognormal	23.2
Salto para trás para fora do *slide*	hiperexponencial (2 fases)	245.5
Salto para trás dentro do mesmo slide	exponencial	23.1

Tabela 5.11: Conjunto de 290 *logs* do CEDERJ: Comparação entre métricas geradas pelo modelo HMM hierárquico

	Carga real	4 est		20 est	
		reestima	trunca	reestima	trunca
Número médio de interações	17.4	14.5	11.2	13.0	11.0
Número médio de pausas	2.9	2.6	2.3	2.5	2.3
Número médio de saltos para frente	10.7	8.6	6.1	7.5	6.1
Número médio de saltos para trás	3.6	3.3	2.8	3.0	2.6
Tempo médio em *play* (em s)	71.6	63.3	80.8	72.9	79.0
Tempo médio em pausa (em s)	136.2	130.6	130.7	130.2	131.8
Tamanho médio do salto para frente (em s)	140.5	246.4	328.6	243.5	301.5
Tamanho médio do salto para trás (em s)	152.7	166.6	166.0	169.2	149.4

do que o da carga real e o número médios de pausas foi maior. Este conjunto serve como exemplo do que ocorre quando não há um estudo e tratamento adequado dos dados usados para parametrizar os modelos. As características de interatividade de um usuário que efetuou uma sessão de 5 minutos podem ser muito diferentes das características de um usuário de uma sessão de 60 minutos. Portanto, *logs* gerados por estes dois usuários não devem ser usados em conjunto para parametrizar os modelos pois podem reduzir a acurácia das cargas sintéticas geradas pelos modelos. Neste caso, uma possível abordagem para melhorar a saída dos modelos, seria classificar o conjunto de *logs* em diferentes grupos, onde cada grupo seria usado para treinar um

Tabela 5.12: Conjunto de 290 *logs* do CEDERJ: Comparação entre métricas geradas pelo modelo de [17]

	Carga real	**Carga HMM Hierárquico**	**Carga de [17]**
Número médio de interações	17.4	14.5	11.1
Número médio de pausas	2.9	2.6	6.0
Número médio de saltos para frente	10.7	8.6	4.0
Número médio de saltos para trás	3.6	3.3	1.1
Tempo médio em *play* (em s)	71.6	63.3	127.9
Tempo médio em pausa (em s)	136.2	130.6	177.7
Tamanho médio do salto para frente (em s)	140.5	246.4	252.0
Tamanho médio do salto para trás (em s)	152.7	166.6	198.4

modelo distinto. Cada modelo seria responsável por gerar carga sintética similar ao grupo de *logs* utilizado para treiná-lo. Posteriormente, as cargas sintéticas geradas pelos modelos poderiam ser usadas em conjunto para representar a carga real.

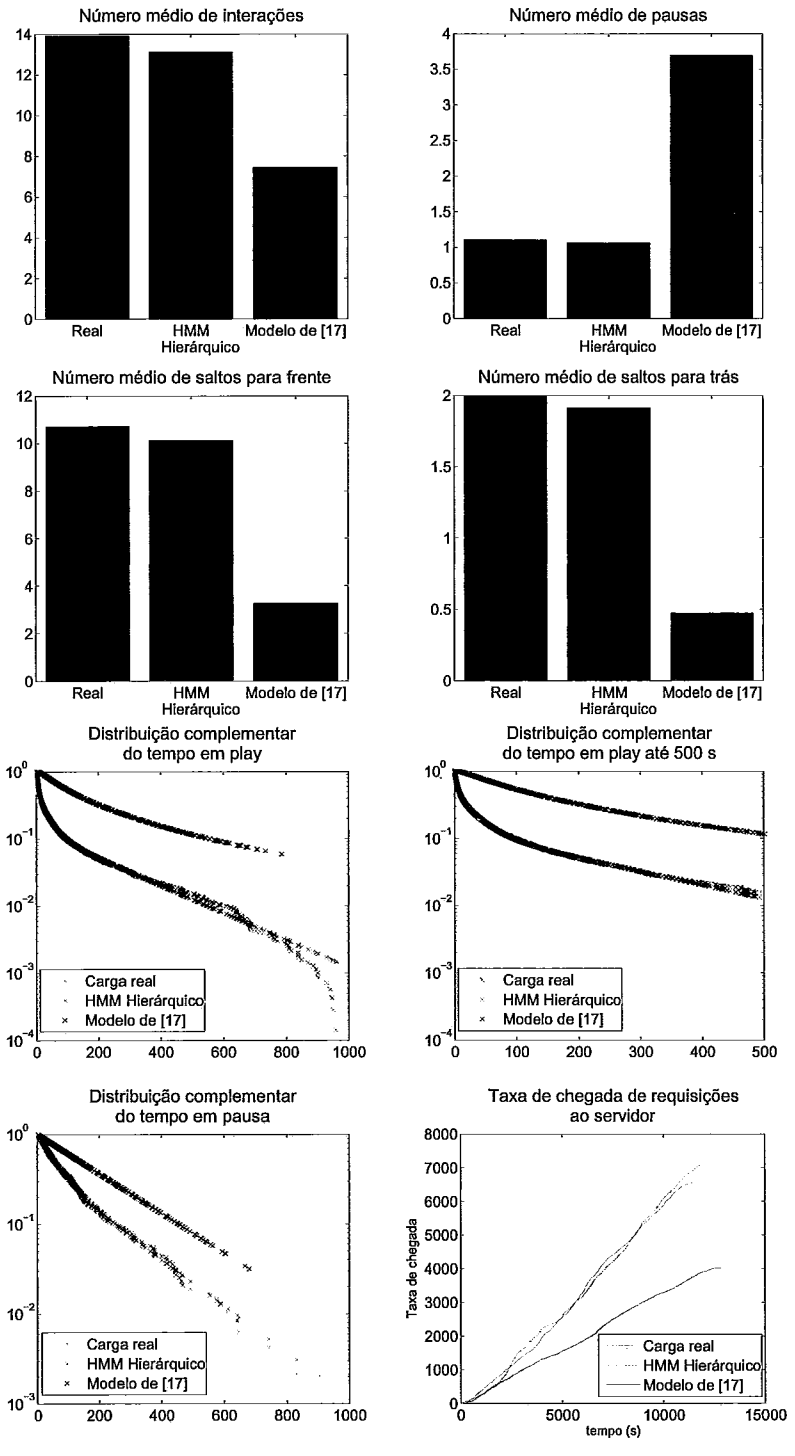


Figura 5.9: Métricas para o conjunto de 476 *logs* do CEDERJ

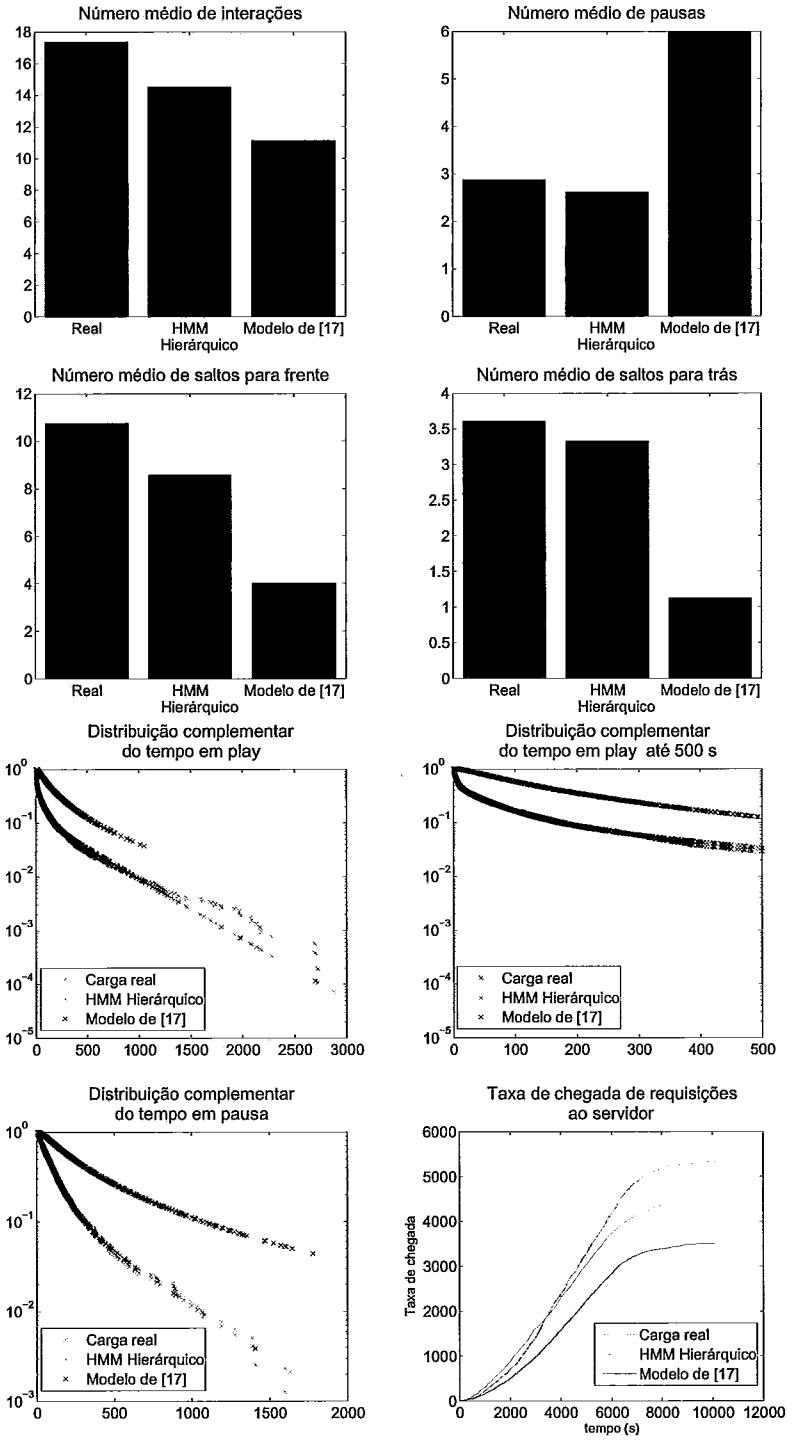


Figura 5.10: Métricas para o conjunto de 290 *logs* do CEDERJ

5.3.2 Logs do MANIC

Parametrizamos o modelo com cargas do sistema MANIC, descrito em maiores detalhes na Seção 5.1.2. Como o conjunto de *logs* disponíveis é menor do que o do CEDERJ analisamos apenas os resultados para a aula mais popular e para o conjunto completo de *logs*.

Assim como para os *logs* do CEDERJ, realizamos simulações com as cargas sintéticas, além da carga real. Cada simulação foi feita com o mesmo número de sessões que o da carga real e as chegadas eram determinadas por um processo de Poisson com taxa de aproximadamente 3 sessões por minuto.

Aula mais popular

A aula mais popular do MANIC, com 65 acessos, possuía duração de aproximadamente 1 hora. As distribuições de probabilidade para as métricas de interatividade da carga real podem ser verificadas na Tabela 5.13.

Os resultados variando o número de estados ocultos e utilizando diferentes abordagens podem ser analisados na Tabela 5.14. Novamente a técnica de reestimar apresentou resultados melhores. Para este exemplo, o modelo de 20 estados apresentou valores médios mais próximos da carga real do que o modelo de 4 estados.

A comparação entre as métricas do modelo HMM hierárquico e o modelo de [17] pode ser verificada através da Tabela 5.15. Para a carga do modelo HMM hierárquico, selecionamos a abordagem de reestimar para 20 estados ocultos.

Alimentamos o modelo de simulação de [5] com as cargas sintéticas e real. A Figura 5.11 apresenta a comparação entre diversas métricas da carga real e das cargas sintéticas. A carga gerada pelo modelo HMM hierárquico apresentou uma boa aproximação para a carga real, porém não foi tão preciso para estimar a distribuição dos tempos em *play* e pausa, como ocorreu para os *logs* do CEDERJ. Estas distribuições não são resultado do modelo HMM hierárquico, pois foram obtidas através

de estimadores de máxima verossimilhança descritos na Seção 4.3.

Para este exemplo, podemos notar que as estatísticas obtidas para a carga do modelo de [17] estão mais próximas das estatísticas da carga real quando comparamos com os exemplos apresentados anteriormente. O número médio de pausas não foi superestimado (como nos exemplos anteriores) e o número médio de saltos para frente é um pouco menor do que o encontrado para a carga real. Observamos também um melhor casamento entre as distribuições dos tempos em *play* e em pausa. Porém, podemos notar através da Figura 5.11 que o modelo de [17] ainda continuou subestimando a taxa de chegada de requisições no servidor, o que é conseqüência do número médio de interações do modelo ser 40% inferior à média de interações da carga real.

Tabela 5.13: Aula mais popular do MANIC: Distribuições de probabilidade para as métricas da carga real

	Distribuição de probabilidade	**Média (em s)**
Play	lognormal	440.0
Pausa	lognormal	179.3
Salto para frente para fora do *slide*	hiperexponencial (2 fases)	182.3
Salto para frente dentro do mesmo slide	lognormal	66.2
Salto para trás para fora do *slide*	hiperexponencial (3 fases)	269.3
Salto para trás dentro do mesmo slide	lognormal	34.4

Tabela 5.14: Aula mais popular do MANIC: Comparação entre métricas geradas pelo modelo HMM hierárquico

	Carga real	**4 est**		**20 est**	
		reestima	**trunca**	**reestima**	**trunca**
Número médio de interações	7.2	6.4	4.3	7.5	4.9
Número médio de pausas	3.4	3.0	1.9	3.3	2.2
Número médio de saltos para frente	2.1	1.9	1.3	2.2	1.4
Número médio de saltos para trás	1.7	1.6	1.1	1.9	1.2
Tempo médio em *play* (em s)	440.0	363.6	516.5	335.2	462.9
Tempo médio em pausa (em s)	179.3	219.5	165.8	187.8	214.2
Tamanho médio do salto para frente (em s)	95.8	88.9	114.8	104.4	99.7
Tamanho médio do salto para trás (em s)	124.9	99.5	101.0	102.6	115.0

Tabela 5.15: Aula mais popular do MANIC: Comparação entre métricas geradas pelo modelo de [17]

	Carga real	**Carga HMM Hierárquico**	**Carga de [17]**
Número médio de interações	7.2	7.5	4.1
Número médio de pausas	3.4	3.3	2.2
Número médio de saltos para frente	2.1	2.2	1.5
Número médio de saltos para trás	1.7	1.9	0.5
Tempo médio em *play* (em s)	440.0	335.2	511.6
Tempo médio em pausa (em s)	179.3	187.8	486.8
Tamanho médio do salto para frente (em s)	95.8	104.4	243.8
Tamanho médio do salto para trás (em s)	124.9	102.6	106.4