Um modelo HMM hierárquico para usuários interativos acessando um servidor multimídia *

Carolina C. L. B. de Vielmond, Rosa M. M. Leão, Edmundo de Souza e Silva

[1]COPPE/Prog. de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro
Caixa Postal 68511 – Rio de Janeiro, RJ – 21941-972

{carolina, rosam, edmundo}@land.ufrj.br

Abstract. *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 an hierarchical HMM (hidden Markov model) 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 the second level. The results show the good accuracy of the model (parameterized by a real system log) when it is used to dimension a multimedia server.*

Resumo. *A maior parte dos mecanismos de compartilhamento de recursos desenvolvidos para tornar os serviços de vídeo sob demanda (VoD) escaláveis tem sido avaliadas 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 (hidden Markov model) 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.*

1. Introdução

A maioria das técnicas propostas na literatura para prover serviços de vídeo sob demanda (VoD) escalável foram desenvolvidas considerando que o acesso dos usuários é seqüencial. Porém, um alto grau de interatividade têm sido observado em diversos

*Este trabalho é parcialmente financiado pelo CNPq e Faperj.

tipos de cargas reais [Costa et al. 2004, Rocha et al. 2005], [Padhye and Kurose 1997], [Tomimura et al. 2006], principalmente nas aplicações de ensino a distância, onde ações como parada, avanço e retorno do vídeo são muito comuns.

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.

Em [Ji et al. 2001] 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 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*.

Em [Rocha et al. 2005], os autores propuseram um modelo para geração de carga sintética baseado em *traces* coletados dos servidores eTeach e MANIC (ensino a distância), e UOL (entretenimento). As ações dos clientes representadas no modelo são: *play*, *stop*, *pause/resume*, *jump forwards* e *jump backwards*. O gerador tem como entradas um *trace* real de sessões para um certo objeto e a taxa de chegada de sessões (o processo de chegada de sessões é considerado Poisson). Inicialmente é construído um modelo onde cada estado corresponde ao usuário executar um comando *play* em um segmento de tamanho fixo do objeto. São adicionados estados para representar ações de *pause* e *stop*. As probabilidades (inicial e de transição entre estados) do modelo são obtidas a partir do *trace* real. O tempo de permanência em cada estado é exponencial (estados de *pause* e estados de *play* que representam segmentos que não são tocados integralmente no *trace* real) e determinístico nos estados de *play* que representam segmentos que sempre são tocados integralmente no *trace* real.

Por fim, em [Tomimura et al. 2006] foi analisado um conjunto de *logs* do MANIC, com conteúdo de vídeo e áudio sincronizados com *slides*. O trabalho vai além da caracterização da carga e também sugere 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*. As observações do modelo 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. Experimentos foram realizados com as cargas real e sintética de forma a validar o modelo.

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 do curso de graduação de Tecnologia da Computação do CEDERJ (Centro de Educação Superior a Distância do Rio de Janeiro), iniciado em Março de 2005 e contando atualmente com mais de 700 alunos matriculados. Um *log* representa os acessos de um usuário a uma aula (vídeo-aula). O modelo é uma variação da abordagem clássica de modelos de Markov ocultos (*hidden Markov models* - HMMs), onde restrin-

gimos a distribuição das observações dentro de um estado oculto, assumindo que essas são geradas por uma cadeia de Markov discreta qualquer. Ele foi inspirado no trabalho de [Silveira and de Souza e Silva 2006], que avaliou diferentes modelos de Markov ocultos como preditores de estatísticas de perda de curto prazo. Vale ressaltar que nosso modelo é genérico, podendo se adequar a outros sistemas multimídia. Como vantagens em relação a outras propostas da literatura podemos citar, a capacidade infinita de memória do modelo, em contraste com modelos Markovianos que possuem memória finita, número reduzido de estados diminuindo a complexidade, parametrização usando uma quantidade bastante grande de *logs* de um servidor real. Mostramos ainda, através de comparação com *logs* reais do CEDERJ, que nosso modelo é acurado para dimensionar um servidor.

Este trabalho está organizado da seguinte forma. Na seção 2 apresentamos o modelo proposto e o procedimento para geração de *logs* sintéticos. Na seção 3 validamos o modelo. A seção 4 apresenta conclusões e trabalhos futuros.

2. O Modelo

Usamos a caracterização da carga dos cursos do CEDERJ apresentada em [Alves 2006] para definir e parametrizar o modelo. Naquele trabalho foram analisados 4274 *logs* referentes ao primeiro e segundo semestres de 2005 e ao primeiro semestre de 2006 dos cursos do CEDERJ. No curso do CEDERJ, as aulas são previamente gravadas e posteriormente ficam armazenadas no servidor multimídia RIO [Muntz et al. 1998, Netto et al. 2005] (*Randomized I/O Multimedia Storage Server*). O servidor RIO é um sistema de armazenamento multimídia universal que usa alocação aleatória e replicação de blocos. Através do programa cliente, os alunos podem escolher a mídia desejada para visualização, e têm disponíveis funcionalidades que propiciam interatividade como pausar, avançar e retroceder o vídeo.

O modelo proposto é baseado no HMM hierárquico de [Silveira and de Souza e Silva 2006], que possui uma cadeia de Markov com dois estados operando dentro de cada estado da cadeia oculta. A estrutura hierárquica possui duas propriedades interessantes. Primeiramente, o número total de parâmetros a serem estimados é menor, reduzindo assim a complexidade da fase de treinamento. Em segundo, dependências de curto prazo são capturadas pela cadeia dentro do estado oculto, enquanto que a dinâmica de longo prazo é governada pela cadeia de Markov oculta.

Em nosso modelo, a cadeia de Markov oculta governa a dinâmica de uma sessão de usuário. Por outro lado, os estados ocultos capturam a dependência das ações do usuário dentro do contexto de um *slide*. Sendo assim, dentro de um estado oculto temos a dinâmica das ações do usuário, e não é possível que esta dinâmica seja governada por um modelo de Gilbert como em [Silveira and de Souza e Silva 2006]. A primeira adaptação no modelo foi permitir que, dentro de cada estado oculto, esta dinâmica a curto prazo fosse capturada por uma cadeia de Markov discreta qualquer. A segunda modificação realizada permitiu que o número de símbolos emitidos em cada estado oculto possa ser variável ao invés de uma constante.

2.1. Definição do modelo

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. A notação adotada segue como a de [Silveira and de Souza e Silva 2006] e [Rabiner 1989].

Seja $\{Y_t\}$ a cadeia de Markov de N estados. A distribuição do estado inicial é dada pelo vetor de N dimensões π, com:

$$\pi_i = P(Y_1 = i).$$

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

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

Seja M, o número de símbolos observáveis. Uma vez que o processo entra em um estado oculto, ele emite um grupo de S observações. Cada grupo de observações será representado pelo vetor $\mathbf{x}_t = [x_{t,1}, x_{t,2}, \ldots, x_{t,S}]$. Seja $\{X_t\}$ o processo de observações, governado pela matriz $\mathbf{B} = \{b_{y_t,\mathbf{x}_t}\}$, i.e.:

$$b_{y_t,\mathbf{x}_t} = P(X_t = \mathbf{x}_t | Y_t = y_t). \tag{1}$$

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

$$\sum_{i=1}^{N} \pi_i = 1, \quad \sum_{j=1}^{N} a_{ij} = 1, \ \forall i, \quad \sum_{\forall j} b_{ij} = 1, \ \forall i, \ \text{ onde } j \in \{1, 2, \ldots, M\}^S.$$

Nos referimos ao modelo como a tupla, $\lambda = (\pi, \mathbf{A}, \mathbf{B})$. Consideremos T valores para o processo de observações, sendo assim os grupos de medições serão $\mathbf{x}_t$, onde $1 \leq t \leq T$.

Para cada estado oculto, i, os parâmetros da cadeia de Markov de M estados são dados por:

$$r_{ij} = P(X_{t,1} = j | Y_t = i), \quad 1 \leq t \leq T$$
$$p_{ijk} = P(X_{t,s} = k | X_{t,s-1} = j, Y_t = i), \quad 1 \leq t \leq T, \quad 1 \leq s \leq S$$

Para cada grupo de medidas, $\mathbf{x}_t$, é suficiente manter o registro apenas das seguintes estatísticas:

$$x_{t,1} = \text{primeira observação no grupo } t \tag{2a}$$

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

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

$$\sum_{i=1}^{M} \sum_{j=1}^{M} S_t^{ij} = S - 1, \quad 1 \leq t \leq T. \tag{3}$$

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 a Equação (1) em função dos parâmetros $\mathbf{r}$ e $\mathbf{p}$:

$$b_{y_t,\mathbf{x}_t} = r_{y_t,k} \prod_{i=1}^{M} \prod_{j=1}^{M} (p_{y_t,i,j})^{S_t^{ij}}, \quad \text{se } x_{t,1} = k, \quad \forall k \in \{1, 2, \ldots, M\} \tag{4}$$

Procedemos calculando as estatísticas (2a) e (2b) e usando esses valores em conjunto com a Equação (4), no procedimento *forward-backward* [Rabiner 1989]. Estendemos o algoritmo Baum-Welch, encontrado em [Baum et al. 1970] e [Levinson et al. 1983], para adicionar as restrições da Equação (4). A cada iteração, este algoritmo maximiza a *função auxiliar*, $Q(\lambda|\overline{\lambda})$, em relação a λ, fazendo uso da estimativa atual dos parâmetros, $\overline{\lambda}$:

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

Esta função pode ser dividida em três termos independentes, $Q_1(\pi|\overline{\lambda})$, $Q_2(\mathbf{A}|\overline{\lambda})$ e $Q_3(\mathbf{B}|\overline{\lambda})$. Maiores detalhes podem ser encontrados em [Rabiner 1989]. Adicionando outras restrições envolvendo π, $\mathbf{A}$ e $\mathbf{B}$, *uma vez que elas permaneçam independentes*, as fórmulas de re-estimação mudarão apenas para as variáveis sobre as quais as novas restrições se aplicam. Sendo assim, podemos restringir a nossa análise apenas a função auxiliar correspondente, $Q_3(\mathbf{B}|\overline{\lambda})$, enquanto que as demais permanecem como em [Rabiner 1989]. Seja a variável $\gamma_t(i) = P(Y_t = i|\mathbf{X}, \lambda)$, a probabilidade de estar no estado oculto i no tempo t, dada a seqüência de observações $\mathbf{X}$ e o modelo λ.

$$Q_3(\mathbf{B}|\overline{\lambda}) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{\forall j} \log b_{ij} \mathbb{I}\{\mathbf{x}_t = j\} \gamma_t(i), \quad \text{onde } j \in \{1, 2, \ldots, M\}^S \tag{5}$$

Na Equação (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), no termo da função auxiliar correspondente aos parâmetros de observação, dada pela Equação (5), temos:

$$Q_3(\mathbf{B}|\overline{\lambda}) = \sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{\forall j} \left[\log r_{i,m} + \log \left(\prod_{k=1}^{M} \prod_{l=1}^{M} (p_{i,k,l})^{S_t^{kl}} \right) \right] \mathbb{I}\{\mathbf{x}_t = j\} \gamma_t(i) \mathbb{I}\{j_{t,1} = m\} \tag{6}$$

É fácil verificar que podemos dividir a função auxiliar dada (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.

$$Q_4(\mathbf{r}_i|\overline{\lambda}) = \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{\forall j} \log r_{i,m}\mathbb{I}\{\mathbf{x}_t = j\}\gamma_t(i)\mathbb{I}\{j_{t,1} = m\} \tag{7}$$

$$Q_5(\mathbf{p}_i|\overline{\lambda}) = \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{\forall j} \log\left(\prod_{k=1}^{M}\prod_{l=1}^{M}(p_{i,k,l})^{S_t^{kl}}\right)\mathbb{I}\{\mathbf{x}_t = j\}\gamma_t(i)\mathbb{I}\{j_{t,1} = m\} \tag{8}$$

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

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

Podemos reescrever os sub-problemas (7) e (8) da seguinte maneira:

$$Q_4(\mathbf{r}_i|\overline{\lambda}) = \sum_{m=1}^{M}\log r_{i,m}\sum_{t=1}^{T}\mathbb{I}\{x_{t,1} = m\}\gamma_t(i) \tag{9}$$

$$Q_5(\mathbf{p}_i|\overline{\lambda}) = \sum_{k=1}^{M}\sum_{l=1}^{M}\log p_{i,k,l}\sum_{t=1}^{T}S_t^{kl}\gamma_t(i) \tag{10}$$

Fixando k na Equação (10) para maximizar $\mathbf{p}_{ik}$, i.e. a probabilidade de transição a partir do estado k, dado o estado oculto i, temos:

$$Q_5(\mathbf{p}_{ik}|\overline{\lambda}) = \sum_{l=1}^{M}\log p_{i,k,l}\sum_{t=1}^{T}S_t^{kl}\gamma_t(i) \tag{11}$$

Resolveremos os sub-problemas (9) e (11) como problemas de otimização, através da aplicação do Lema 2 do trabalho [Levinson et al. 1983]. As seguintes expressões são resultado da maximização dos termos das funções auxiliares $Q_4(\mathbf{r}_i|\overline{\lambda})$ e $Q_5(\mathbf{p}_{ik}|\overline{\lambda})$.

$$r_{i,m} = \frac{\sum_{t=1}^{T}\mathbb{I}\{x_{t,1} = m\}\gamma_t(i)}{\sum_{t=1}^{T}\gamma_t(i)} \qquad p_{ikl} = \frac{\sum_{t=1}^{T}S_t^{kl}\gamma_t(i)}{\sum_{j=1}^{M}\sum_{t=1}^{T}S_t^{kj}\gamma_t(i)} \tag{12}$$

No modelo proposto em [Silveira and de Souza e Silva 2006], 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, a saída do *slide*, que será representado na cadeia de Markov dentro do estado oculto como um estado absorvente. Agora temos que S, número de símbolos

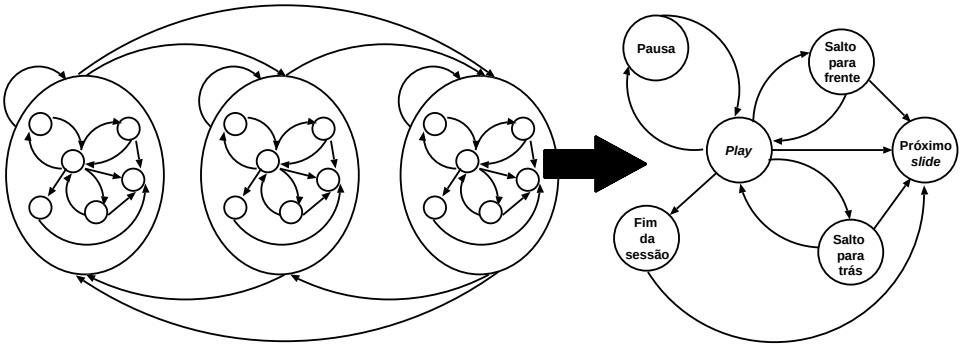


Figura 1. Modelo HMM hierárquico

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 (3) da seguinte forma: $\sum_{i=1}^{M} \sum_{j=1}^{M} S_t^{ij} = Z_t$.

Este fato não altera os resultados das equações em (12). Porém deve-se atentar ao fato de que no desenvolvimento realizado a partir das equações (9) e (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 discreta representada dentro do estado oculto. 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 do usuário feita em [Alves 2006]. As transições dentro de um estado oculto foram inspiradas nas possíveis ações realizadas na aplicação real. O símbolo *próximo slide* causa uma transição entre os estados ocultos, já que representa o fim de um *slide*. A cadeia de Markov discreta que governa o estado oculto sempre é iniciada no estado *play*. 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 dentro do estado oculto, já a dinâmica da sessão do usuário é capturada pela cadeia de Markov oculta. A Figura 1 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 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, já no modelo HMM hierárquico não, devido a estrutura da cadeia de Markov dentro do estado oculto ilustrada na Figura 1.

É necessário ainda especificar a quantidade de estados da cadeia de Markov oculta. O valor apropriado depende do tipo de aplicação do modelo e da carga usada para parametrizá-lo. Na seção 3 é feita uma análise para definir a quantidade de estados ocultos mais apropriada para realizar a validação do modelo. O custo de parametrização da HMM

é $O(N^2T)$, onde N é o número de estados ocultos da cadeia (de 4 a 20 nos nossos experimentos) e T o número de interações por aula (aproximadamente 40).

2.2. Geração dos *logs* sintéticos

Para gerar *logs* sintéticos com o modelo proposto, usamos um conjunto de *logs* reais do CEDERJ para treiná-lo. Os *logs* compreendem o período entre o primeiro semestre de 2005 e o primeiro semestre de 2006. Apenas *logs* de alunos acessando o sistema foram utilizados, mais precisamente, não incluímos no conjunto de *logs* aqueles gerados pelo acesso de técnicos, tutores ou professores. Em [Alves 2006] foi feita uma análise dos *logs* do CEDERJ, aplicando diferentes filtros de acordo com a duração mínima da sessão, visto que estes apresentam características diferentes de acordo com a sua duração. Em nosso estudo, somente *logs* com duração da sessão maiores que 5 minutos foram utilizados.

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. Precisamos então analisar dados dos *logs* reais tais como a distribuição de probabilidade do tamanho dos saltos, do tempo em *play* e do tempo em *pause*, para inserir esses dados na geração do *log* sintético.

Para obter as distribuições de probabilidade que caracterizam as medidas de interesse para o conjunto de *logs* escolhido, procedemos com uma metodologia similar a adotada em [Tomimura et al. 2006, Alves 2006]. 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). Para determinar o tipo de distribuição que melhor aproxima os dados empíricos de uma dada variável fizemos uma análise visual e outra quantitativa. A análise visual consiste em plotar os gráficos da distribuição complementar com o eixo das ordenadas em escala logarítmica para evidenciar a cauda da distribuição e avaliamos visualmente aquela que se assemelhava mais com a distribuição empírica. Também comparamos o erro médio quadrático (*mean squared error* - MSE) entre as distribuições empíricas e a estimada. Além disso, aplicamos o teste de Kolmogorov-Smirnov, 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. Maiores detalhes sobre os métodos utilizados podem ser encontrados em [Trivedi 1982] e [Ross 1990].

Uma questão surge quando as métricas possuem correlação com a posição do vídeo. A princípio seria necessário obter uma distribuição para cada intervalo do vídeo. No entanto, assim como em [Tomimura et al. 2006] e [Alves 2006], preferimos utilizar uma única distribuição, obtida a partir de todas amostras. Geramos amostras a partir desta 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 o *log* sintético referente a esta sessão.

3. Validação e análise comparativa do modelo proposto

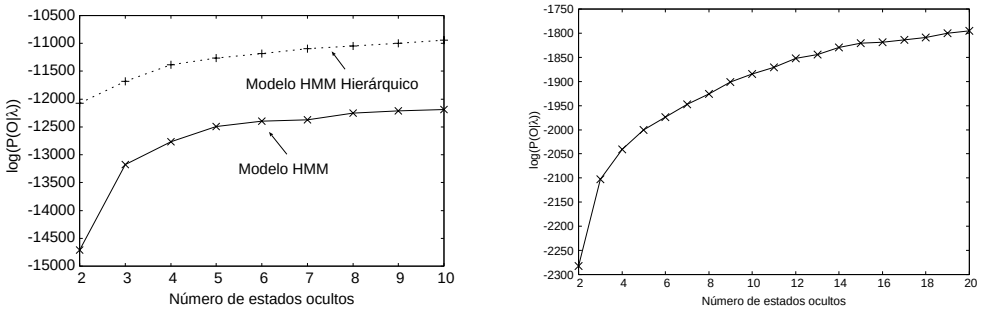
Nesta seção apresentamos a validação do modelo HMM hierárquico proposto. Primeiramente realizamos uma comparação entre nosso modelo e uma variação do modelo HMM proposto em [Tomimura et al. 2006]. Em seguida, avaliamos a acurácia do modelo quando este é utilizado para dimensionar um servidor que implementa técnicas de compartilhamento de banda. Para parametrizar o modelo utilizamos *logs* reais de aulas do curso de graduação de Tecnologia da Computação do CEDERJ, como comentado na seção 2. Neste trabalho não comparamos nosso modelos com aqueles baseados em cadeias de Markov. É sabido que cadeias de Markov, diferentemente de HMM, tem memória finita e portanto HMM podem capturar com mais precisão dependências temporais. O modelo de [Ji et al. 2001] é uma HMM, porém usado para modelar uma versão do sistema MANIC com apenas áudio sincronizado com transparências (não há vídeo).

O modelo originalmente proposto em [Tomimura et al. 2006] 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 22% 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*.

Na primeira parte da validação separamos os *logs* disponíveis do CEDERJ em 2 conjuntos. O primeiro foi utilizado para treinar os modelos, composto de 1332 *logs* e o outro, com os 308 restantes, para calcular a probabilidade dos mesmos terem sido gerados pelos modelos previamente treinados. Realizamos 20 treinamentos independentes 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 2(a) mostra a comparação entre os modelos. Claramente o modelo HMM hierárquico tem melhor desempenho, mesmo para poucos estados ocultos.

A segunda parte da validação do modelo consiste em usar *logs* sintéticos (gerados a partir do modelo) e *logs* reais como carga para um simulador de um servidor multimídia desenvolvido em [da Silva Rodrigues and Leão 2006], e comparar os resultados obtidos com ambas as cargas. O modelo de simulação foi criado usando a ferramenta Tangram-II [de Souza e Silva et al. 2006] e consiste de diversos clientes acessando um objeto de um servidor de vídeo que implementa uma técnica para compartilhar o canal de transmissão do servidor, denominada PIE (*Patching Interativo Eficiente*) [da Silva Rodrigues and Leão 2006]. 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.

Para realizar a simulação escolhemos dentre os *logs* disponíveis do CEDERJ, a aula mais popular, com 48 acessos e 7606 segundos de duração. Treinamos o modelo HMM hierárquico com este conjunto de *logs* variando o número de estados ocultos na faixa de valores entre 2 e 20. Plotamos o logaritmo da medida de verossimilhança para



(a) (b)

Figura 2. (a) Probabilidade das observações terem sido geradas pelo modelo para cada número de estados ocultos (b) Logaritmo da verossimilhança para cada número de estados ocultos

cada quantidade de estados ocultos, que pode ser verificado na Figura 2(b). Realizamos simulações com diferentes valores para o número de estados ocultos com a finalidade de avaliar o ganho ao utilizar mais estados ocultos com relação ao aumento em complexidade do modelo.

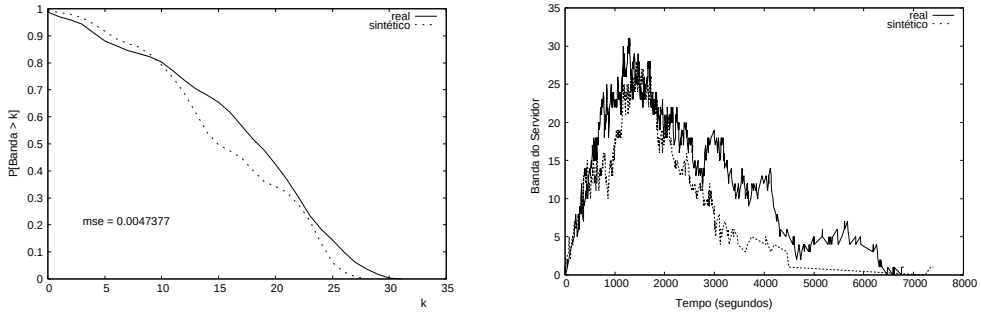
Com o modelo parametrizado, geramos seqüências de ações interativas compostas pelos símbolos *play*, pausa, salto para frente, salto para trás, próximo *slide* e saída da sessão. Mas ainda é necessário analisar a carga real para inserir as informações referentes ao instante e a posição no vídeo associados a cada uma das ações interativas. Na Tabela 1 listamos as medidas de interesse e as distribuições encontradas, com seus respectivos parâmetros, para o conjunto de *logs* reais escolhido para a simulação.

Tabela 1. Caracterização da carga real

Métrica (em segundos)	**Distribuição**	**Parâmetros**
tempo em pausa	gamma	$\alpha = 0.88$ e $\theta = 132.64$
tempo em *play*	lognormal	$\mu = 2.29$ e $\sigma = 2.16$
tamanho do salto para frente (mais de um *slide*)	lognormal	$\mu = 5.24$ e $\sigma = 1.39$
tamanho do salto para frente (dentro de um *slide*)	exponencial	$\mu = 90.80$
tamanho do salto para trás (mais de um *slide*)	lognormal	$\mu = 5.74$ e $\sigma = 1.36$
tamanho do salto para trás (dentro de um *slide*)	lognormal	$\mu = 3.12$ e $\sigma = 1.36$

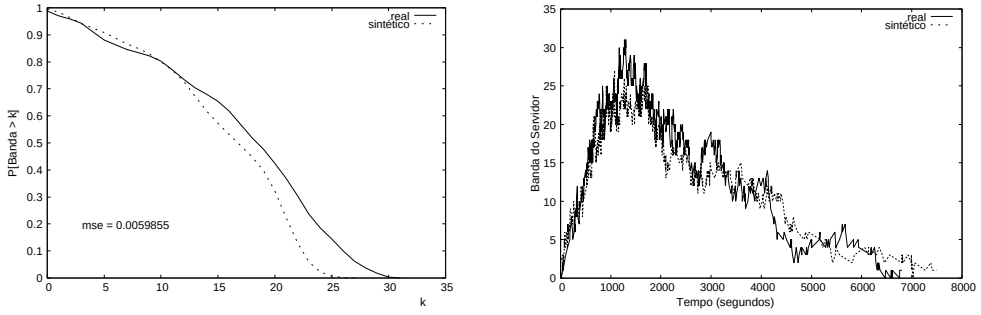
Na seção 2 comentamos que existe um problema a ser contornado na geração dos *logs* sintéticos. Aquele que diz respeito a ultrapassar os limites do vídeo no momento em que inserimos as informações de tempo e posição no vídeo, para cada ação interativa. Geramos *logs* sintéticos adotando cada uma das 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. A Tabela 2 mostra uma comparação entre métricas da carga real e das cargas sintéticas considerando as diferentes abordagens, para conjuntos de 48 *logs* sintéticos gerados pelo modelo HMM hierárquico com 4 estados ocultos. Pelos dados da Tabela 2, as abordagens de truncar e reestimar são aquelas que tem métricas que mais se aproximam às da carga real.

Realizamos simulações com cada um dos três conjuntos de cargas sintéticas, além



(a) (b)

Figura 3. Descarta (a) Distribuição Complementar da banda (b) Distribuição da banda ao longo do tempo



(a) (b)

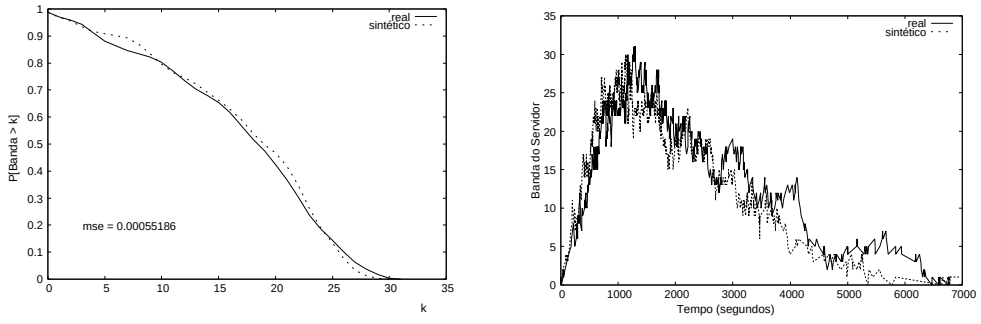
Figura 4. Trunca (a) Distribuição Complementar da banda (b) Distribuição da banda ao longo do tempo

da carga real. Cada simulação foi feita com 48 clientes (cada *log* gerado representa um cliente) e as chegadas eram determinadas por um processo de Poisson com taxa de aproximadamente 3 sessões por minuto. Para comparar os resultados, plotamos a distribuição da banda ao longo do tempo e a distribuição complementar da banda. A banda é medida em número de canais de transmissão de dados simultâneos em uso no sistema. Também calculamos o MSE entre as curvas da distribuição complementar da banda das cargas real e sintética. As Figuras 3, 4 e 5 ilustram os resultados para as cargas geradas pelo modelo HMM hierárquico com 4 estados ocultos com as diferentes abordagens. A abordagem de reestimar foi a mais satisfatória dentre as três sugeridas, se aproximando bastante da curva da distribuição da banda da carga real.

Analisando a curva da Figura 2(b) pode-se notar que o logaritmo da medida de verossimilhança variando o número de estados ocultos é crescente, estabilizando-se ape-

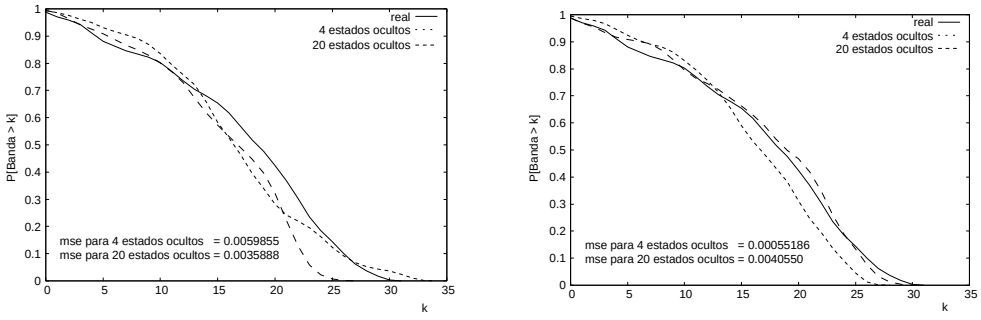
Tabela 2. Características das cargas

	real	**sintética**		
		descarta	**trunca**	**reestima**
número médio de requisições	20.25	13.33	20.10	20.96
tamanho médio do segmento (em segundos)	97.83	108.01	93.743	80.50
desvio padrão do tamanho do segmento	337.47	348.94	346.18	269.24



(a) (b)

Figura 5. Reestima (a) Distribuição Complementar da banda (b) Distribuição da banda ao longo do tempo

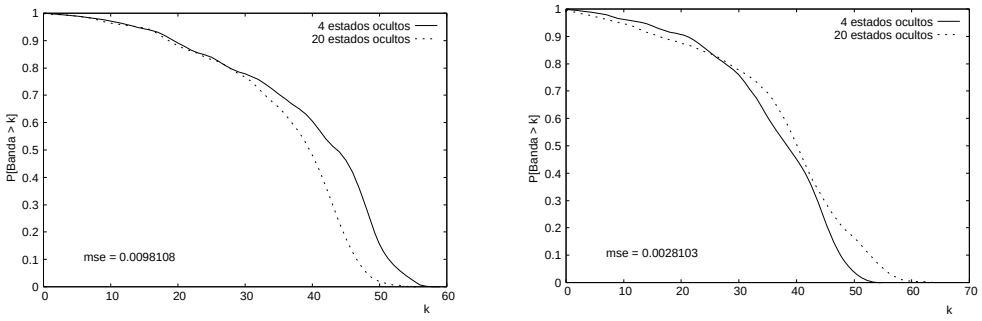


(a) (b)

Figura 6. Distribuição Complementar da banda para 48 *logs* sintéticos para a abordagem de (a) truncar (b) reestimar

nas após 15 estados ocultos. Realizamos simulações com o modelo HMM hierárquico com 20 estados ocultos para comparar com os resultados com 4 estados ocultos, para as abordagens de truncar e reestimar. Plotamos a distribuição complementar da banda para as cargas real e sintéticas com 4 e 20 estados, que pode ser observada nas Figuras 6(a) e 6(b). Para a abordagem de truncar houve melhora na aproximação da curva da distribuição da banda para o modelo com 20 estados ocultos, enquanto que para a abordagem de reestimar não. Podemos observar que a diferença entre os dois modelos (4 e 20 estados) é pequena.

Realizamos outra análise com a finalidade de verificar o comportamento do modelo com o aumento de clientes acessando um mesmo objeto no servidor. Geramos 200 *logs* com o modelo HMM hierárquico com 4 e 20 estados ocultos, para as abordagens de truncar e reestimar que foram as que apresentaram melhor resultado no exemplo anterior. Alimentamos o modelo de simulação do servidor de vídeo com estes *logs* e plotamos a distribuição complementar da banda, ilustrada nas Figuras 7(a) e 7(b). A partir das figuras pode-se observar que os modelos de 4 e 20 estados apresentam resultados similares como para o caso anterior de um sistema com menos clientes. Sendo assim, dependendo da aplicação do modelo, o ganho em precisão pode não compensar o aumento de complexidade dado pelo acréscimo de estados ocultos. Como resultado da validação, temos que a carga sintética gerada pelo nosso modelo apresentou impacto na distribuição da banda do servidor similar ao da carga real, mesmo para um modelo com poucos estados ocultos.



(a) (b)

Figura 7. Distribuição Complementar da banda para 200 *logs* sintéticos para a abordagem de (a) truncar (b) reestimar

Enfatizamos que a primeira etapa da validação é a mais importante, pois indica que o nosso modelo é estatisticamente semelhante a 308 *logs* reais quando treinado com um outro conjunto de 1332 *logs*. A segunda etapa é apenas um exemplo de aplicabilidade do modelo, onde escolhemos mostrar que o nosso modelo pode prever a distribuição de banda relativa a uma aula com um bom número de acessos, no caso 48, para obter estatísticas confiáveis.

4. Conclusões e trabalhos futuros

Neste trabalho propomos um novo modelo para geração de carga sintética de usuários interativos acessando um servidor multimídia com conteúdo educacional. O modelo proposto consiste em um HMM hierárquico, onde dependências de curto prazo são representadas dentro de um estado oculto e dependências de longo prazo são representadas pela cadeia oculta.

Para parametrizar o modelo utilizamos um conjunto bastante grande de *logs* reais de aulas do curso do CEDERJ. Nosso modelo mostrou-se bastante acurado quando usado para dimensionar um servidor multimídia, mesmo considerando um número reduzido de estados.

Sugerimos como trabalhos futuros usar o modelo proposto em outros tipos de aplicações multimídia, como por exemplo mídia para entretenimento, e estudar os resultados da utilização da carga sintética em um emulador de um cliente em um servidor de vídeo real.

Referências

Alves, B. C. B. (2006). Caracterizando variáveis de interatividade dos alunos do curso de ciência da computação do CEDERJ baseado no servidor multimídia RIO. Technical report, COPPE/UFRJ.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

Costa, C., Cunha, I., Borges, A., Ramos, C., Rocha, M., Almeida, J. M., and Ribeiro-Neto, B. (2004). Analyzing client interactivity in streaming media. In *Proc. 13th ACM Int'l World Wide Web Conference*, pages 534–543.

da Silva Rodrigues, C. K. and Leão, R. M. M. (2006). Técnicas para Sistemas de Vídeo sob Demanda Escaláveis. In *XXIV Simpósio Brasileiro de Redes de Computadores (SBRC 2006)*.

de Souza e Silva, E., Leão, R. M., da Silva, A. P. C., de A. Rocha, A. A., Duarte, F. P., Filho, F. J. S., Jaime, G. D., and Muntz, R. R. (2006). Modeling, Analysis, Measurement and Experimentation with the Tangram-II Integrated. In *International Conference on Performance Evaluation Methodologies and Tools*.

Ji, P., Kurose, J., and Woolf, B. (2001). Student behavioral model based prefetching in online tutoring system. Technical Report CMPSCI-TR-01-27, University of Massachusetts at Amherst.

Levinson, S. E., Rabiner, L. R., and Sondhi, M. M. (1983). An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell Systems Technical Journal*, 62(4):1035–1074.

Muntz, R., Santos, J., and Berson, S. (1998). A Parallel Disk Storage System for Realtime Multimedia Applicatons. In *International Journal of Intelligent Systems, Special Issue on Multimedia Computing System*, volume 13, pages 1137–1174.

Netto, B. C. M., Azevedo, J. A., e Silva, E. A. S., and Leão, R. M. M. (2005). Servidor Multimídia RIO em Ensino a Distância. In *6th International Free Software Forum*.

Padhye, J. and Kurose, J. (1997). An empirical study of client interactions with a continuous-media courseware server. Technical Report UM-CS-1997-056, University of Massachusetts at Amherst.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285.

Rocha, M., Maia, M., Cunha, I., Almeida, J., and Campos, S. (2005). Scalable Media Streaming to Interactive Users. In *MULTIMEDIA'05: Proceedings of the 13th annual ACM international conference on Multimedia*, Singapore.

Ross, S. M. (1990). *A Course in Simulation*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Silveira, F. and de Souza e Silva, E. (2006). Modeling the short-term dynamics of packet losses. *ACM Performance Evaluation Review*.

Tomimura, D., Leão, R. M. M., de Souza e Silva, E., and Filho, F. S. (2006). Caracterização do comportamento de usuários acessando um vídeo de ensino à distância. In *SBC2006 - V Workshop em Desempenho de Sistemas Computacionais e de Comunicação*, pages 34–53.

Trivedi, K. S. (1982). *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. Prentice Hall PTR, Upper Saddle River, NJ, USA.