

Tese de Doutorado

Inferência em Filas Markovianas Finitas e Infinitas

por

Márcio Augusto da Cruz Almeida

Orientador:

Frederico R. B. Cruz

Junho de 2016

Márcio Augusto da Cruz Almeida

Inferência em Filas Markovianas

Finitas e Infinitas

Tese de Doutorado apresentada ao Departamento de Estatística do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Doutor em Estatística.

Orientador: Frederico R. B. Cruz

Universidade Federal de Minas Gerais
Belo Horizonte, junho de 2016



ATA DA DEFESA DE TESE DO ALUNO MÁRCIO AUGUSTO DA CRUZ ALMEIDA

Realizou-se, no dia 10 de junho de 2016, às 14:00 horas, 2077 - ICEX, da Universidade Federal de Minas Gerais, a 39ª defesa de tese, intitulada "*Inferência em Filas Markovianas Finitas e Infinitas*", apresentada por MÁRCIO AUGUSTO DA CRUZ ALMEIDA, número de registro 2013655384, graduado no curso de ESTATÍSTICA, como requisito parcial para a obtenção do grau de Doutor em ESTATÍSTICA, à seguinte Comissão Examinadora: Prof. Frederico Rodrigues Borges da Cruz - Orientador (DEST/UFMG), Prof. Luiz Henrique Duczmal (DEST/UFMG), Prof. Roberto da Costa Quinino (DEST/UFMG), Prof. Fernando Luiz Pereira de Oliveira (DEST/UFOP), Prof. Marcos Flávio Silveira Vasconcelos D'Angelo (DCC/UNIMONTES).

A Comissão considerou a tese:

Aprovada

Reprovada

Finalizados os trabalhos, lavrei a presente ata que, lida e aprovada, vai assinada por mim e pelos membros da Comissão.

Belo Horizonte, 10 de junho de 2016.

Prof. Frederico Rodrigues Borges da Cruz (Doutor)

Prof. Luiz Henrique Duczmal (Doutor)

Prof. Roberto da Costa Quinino (Doutor)

Prof. Fernando Luiz Pereira de Oliveira (Doutor)

Prof. Marcos Flávio Silveira Vasconcelos D'Angelo (Doutor)



UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

UFMG

FOLHA DE APROVAÇÃO

"Inferência em Filas Markovianas Finitas e Infinitas"

MÁRCIO AUGUSTO DA CRUZ ALMEIDA

Tese submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ESTATÍSTICA, como requisito para obtenção do grau de Doutor em ESTATÍSTICA, área de concentração ESTATÍSTICA E PROBABILIDADE.

Aprovada em 10 de junho de 2016, pela banca constituída pelos membros:

Prof. Frederico Rodrigues Borges da Cruz - Orientador
DEST/UFMG

Prof. Luiz Henrique Duczmal
DEST/UFMG

Prof. Roberto da Costa Quinino
DEST/UFMG

Prof. Fernando Luiz Pereira de Oliveira
DEST/UFOP

Prof. Marcos Flávio Silveira Vasconcelos D'Angelo
DCC/UNIMONTES

Belo Horizonte, 10 de junho de 2016.

Agradecimentos

Agradeço a Deus por ter colocado pessoas na minha vida que foram importantes para a conclusão deste trabalho, como:

A minha esposa e meu filho Guilherme, por estarem juntos em todos os momentos desta caminhada, amo vocês;

Meu pai, por participar de todos os momentos marcantes na carreira acadêmica;

Ao amigos conquistados durante o curso, como Francisco, Nívea, Paulo e Silvana, que foram fundamentais ao longo do curso;

Ao Prof. Frederico, meu orientador, pelo seu profissionalismo e dedicação, que me fez crescer e aprender;

Ao professores que durante o curso ministraram disciplinas importante para meu aprendizado;

Ao meu amigo Alex e família, pelo acolhimento em sua residência no primeiro ano de curso;

Aos amigos e irmãos da Igreja Videira, que me ajudaram e orações, fortalecendo meu espírito e a minha fé;

A UFPA, pelo financiamento ao longo deste curso.

Resumo

Em teoria de filas, estuda-se seu comportamento, seu processo de formação e analisam-se características de desempenho, tais como o número esperado de clientes no sistema e na fila, a intensidade do tráfego, definida como sendo a razão entre a taxa de chegada e a taxa de atendimento, entre outros. No entanto, sabe-se que na maioria das vezes os parâmetros envolvidos no processo não são conhecidos e precisam ser estimados através de algum método estatístico. Esta tese visa obter estimativas para algumas medidas de desempenho em filas markovianas finitas e infinitas com um único servidor, denominadas respectivamente por $M/M/1/K$ e $M/M/1$, na notação de Kendall. Uma metodologia sobre o enfoque frequentista e bayesiano é apresentada. Do ponto de vista clássico, foi constatada a presença de vício no estimador de máxima verossimilhança para ρ e investigada a utilização do conhecido método *bootstrap* não paramétrico para sua correção. Sob o enfoque bayesiano, foram obtidas distribuições *a posteriori* e preditivas para parâmetros de interesse. Amostras foram obtidas através de simulação. Os resultados apontaram para uma correção efetiva do vício, possibilitando redução no tamanho de amostra, com conseqüente diminuição nos custos e tempo para a obtenção das medidas de desempenho. Quanto ao método bayesiano, observou-se pelo fator de Bayes que várias são as distribuições *a priori* que poderiam ser empregadas, entre as informativas e não informativas. Também, para filas $M/M/1/K$, os estimadores bayesianos para ρ apresentaram a menor variabilidade entre todos, embora não

tenham sido aqueles com os menores erros de estimação médios.

Palavras-chaves: Filas markovianas; predição bayesiana; medidas de desempenho; intensidade de tráfego.

Abstract

In queueing theory, the behavior and formation process of queues are studied and their performance measures are analyzed, including the number of customer in the system and in the queue, the traffic intensity, which is defined as the ratio between the arrival rate and the service rate, among others. However, the parameters of the process are usually unknown and must be estimated using some statistical inferential method. This thesis aims at obtaining estimates for some performance measures of $M/M/1/K$ and $M/M/1$ queues, which, in Kendall notation, stands for Markovian single-server finite and infinity queues, respectively. A methodology based on the classical frequentist and Bayesian approaches is presented. From the classical point of view, the maximum likelihood estimator for ρ was shown to be biased and a correction based on the well-known nonparametric bootstrap method for correction was proposed and investigated. Under the Bayesian approach *a posteriori* and predictive distributions were obtained for the parameters of interest. Samples were obtained through simulations. The results have shown an effective correction of bias, enabling reduction in the sample size, with consequent reduction in costs and time to obtain the performance measures. Concerning the Bayesian method, it was observed from the Bayes factor that several *a priori* distributions could be used, among informative and non-informative ones. Also, for $M/M/1/K$ queues, Bayesian estimators for ρ presented the lowest variability although they were not the ones with the lowest bias in average.

Keywords: Markovian queues; Bayesian prediction; performance measures; traffic intensity.

Glossário

Algumas definições de símbolos e abreviações, usadas frequentemente ao longo deste texto, são apresentadas neste glossário. As definições estão listadas em ordem alfabética, com as letras gregas inseridas de acordo com seu nome em português.

$A/B/X/Y/Z$: Notação usual para descrever um modelo de filas, em que A representa o padrão de chegada, B , o de serviço, X , o número de canais de serviço, Y , a capacidade total do sistema (incluindo os itens em serviço) e Z , a disciplina da fila;

A: Matriz geradora, para o processo de nascimento-e-morte de um sistema de filas;

B_{jk} : Fator de Bayes para comparar dois modelos M_j e M_k ;

$E(M|x)$: Representa a esperança preditiva *a posteriori* do número de clientes (M);

EMV: Estimador de máxima verossimilhança;

EPM: Erro padrão da média;

FIFO: Disciplina de fila ‘primeiro a chegar, primeiro a ser atendido’ (do inglês ‘*First-In, First-Out*’);

$f(t)$: Densidade de probabilidade dos tempos entre chegadas;

$g(t)$: Densidade de probabilidade dos tempos de serviço;

$I(\theta)$: Informação de Fisher para θ ;

K : Capacidade total (incluindo os itens em serviço) de um sistema de filas;

L : Número esperado de clientes no sistema;

LIFO: Disciplina de fila ‘último a chegar, primeiro a ser atendido’ (do inglês ‘*Last-In, First-Out*’);

L_q : Tamanho médio da fila;

λ : Taxa de chegada na fila;

M : Número de clientes no sistema no momento da partida;

M : Processo markoviano (quando inserido na notação $A/B/X/Y/Z$, para descrever um modelo de filas);

μ : Taxa de serviço na fila;

p_n : Probabilidade estacionária de haver n clientes no sistema de filas;

$P_n(t)$: Probabilidade estacionária de haver n usuários no sistema de filas em tempo contínuo;

$p(\theta)$: Distribuição de probabilidade *a priori*;

$p(x|\theta)$ ou $L(\theta; x)$: Função de verossimilhança;

$P(\theta|x)$: Distribuição de probabilidade *a posteriori*;

ρ : Intensidade de tráfego, definida como $\rho \equiv \lambda/\mu$;

θ : Parâmetro desconhecido, sobre o qual se deseja realizar inferência;

$x = \{x_1, x_2, \dots, x_n\}$: Representa uma amostra aleatória do número de clientes que ficaram no sistema em n momentos de partidas de usuários;

W : Tempo médio de espera no sistema;

W_q : Tempo médio de espera na fila.

Índice

Resumo	6
Abstract	8
Glossário	10
Lista de Figuras	16
Lista de Tabelas	17
1 Introdução	1
1.1 Motivação	2
1.2 Contribuições	5
1.3 Organização	7
2 Itens Fundamentais	8
2.1 Introdução à Teoria de Filas	8
2.1.1 Padrão de Chegada dos Clientes	9
2.1.2 Padrão de Serviço	10
2.1.3 Número de Servidores	10
2.1.4 Capacidade do Sistema e Estágios de Serviço	11
2.1.5 População de Usuários	13

2.1.6	Disciplina da Fila	13
2.2	Processos de Nascimento-e-Morte	13
2.3	Inferência Clássica	18
2.4	Correção <i>Bootstrap</i> de Vício	18
2.5	Inferência Bayesiana	19
2.6	Observações Finais	21
3	Metodologia Inferencial	22
3.1	Filas $M/M/1$	22
3.1.1	Introdução	22
3.1.2	Inferência Clássica	25
3.1.3	Correção de Vício via <i>Bootstrap</i>	27
3.1.4	Inferência Bayesiana	27
3.2	Filas $M/M/1/K$	37
3.2.1	Introdução	37
3.2.2	Inferência Clássica	37
3.2.3	Correção <i>Bootstrap</i>	39
3.2.4	Inferência Bayesiana	40
3.2.5	Simulação	42
3.3	Observações Finais	44
4	Resultados Computacionais e Discussão	45
4.1	Filas $M/M/1$	45
4.1.1	Inferência Clássica	46
4.1.2	Inferência Bayesiana	54
4.2	Filas $M/M/1/K$	59
4.3	Observações Finais	67

5	Conclusões e Observações Finais	68
5.1	Sumário	68
5.2	Propostas de Continuidade	70
	Referências Bibliográficas	71
A	Listagem dos Programas em R para Filas $M/M/1$	77
B	Listagem dos Programas em R para Filas $M/M/1/K$	83

Lista de Figuras

1.1	Diagrama morfológico das abordagens inferenciais para filas markovianas	6
2.1	Sistema com canal simples com fila única	11
2.2	Sistema multicanal com fila única	11
2.3	Sistema multicanal com filas em paralelo	12
2.4	Sistema com retroalimentação (<i>feedback</i>)	12
4.1	Estimativas clássicas médias de ρ e EPM (entre parênteses)	48
4.2	Desempenho dos estimadores para ρ	49
4.3	Estimativas clássicas médias de L e EPM (entre parênteses)	50
4.4	Desempenho dos estimadores para L	51
4.5	Estimativas clássicas médias de L_q e EPM (entre parênteses)	52
4.6	Desempenho dos estimadores para L_q	53
4.7	Distribuições <i>a priori</i> beta e uniforme truncada para ρ	54
4.8	Desempenho dos estimadores para ρ para $K = 5$	64
4.9	Desempenho dos estimadores para ρ para $K = 20$	65
4.10	Desempenho dos estimadores para ρ para $K = 80$	66

Lista de Tabelas

4.1	Estimativas bayesianas médias de ρ e EPM (entre parênteses) . . .	56
4.2	Valores exatos e estimativas médias para L e L_q	57
4.3	Probabilidade preditiva <i>a posteriori</i> do número de clientes no sistema e o fator de Bayes	58
4.4	Estimativas médias de ρ e EPM (entre parênteses) para $K = 5$. . .	61
4.5	Estimativas médias de ρ e EPM (entre parênteses) para $K = 20$. .	62
4.6	Estimativas médias de ρ e EPM (entre parênteses) para $K = 80$. .	63

Capítulo 1

Introdução

Teoria de filas é um ramo da matemática que analisa a formação e o comportamento das filas. É uma importante área de pesquisa e atrai interesse dos pesquisadores. Isso porque as filas fazem parte do nosso dia-a-dia. No supermercado, no banco, nos centros lotéricos, nos postos de gasolina, enfim, em praticamente qualquer lugar podemos nos deparar com um sistema de filas. Sabe-se que as filas sempre ocorrem quando a procura por um determinado serviço apresenta estocasticidade.

Diversos são os modelos de filas. Na notação de [Kendall \(1953\)](#), um sistema de filas é descrito por uma série de símbolos e barras, tais como $A/B/X/Y/Z$, em que A indica de alguma forma a distribuição do tempo entre chegadas (por exemplo, M de markoviano, para chegadas Poisson), B , o padrão de serviço, descrito por uma distribuição de probabilidade para o tempo de serviço (por exemplo, M de markoviano, para tempos de serviço exponenciais), X , o número de canais de serviço em paralelo, Y , a capacidade máxima do sistema, incluindo os itens em atendimento (que, quando omitida, significa filas com capacidade infinita), e Z , a disciplina da fila (por exemplo, disciplina FIFO, ou ‘primeiro a chegar, primeiro a ser atendido’, do inglês *First-In, First-Out*, que, quando omitida, é a disciplina

subentendida).

O grande interesse é a possibilidade de otimizar os sistemas de filas, garantindo assim uma redução dos seus custos operacionais e melhorando seu desempenho. Logo, para o gestor, é essencial o conhecimento de algumas das características das filas tais como a taxa de chegada, λ , a taxa de serviço, μ , a intensidade do tráfego¹, ρ , definida como sendo a razão entre λ e μ , número esperado de clientes no sistema, L , e o tamanho médio da fila, L_q , entre outras. Tais medidas podem ser quantificadas e avaliadas através de funções advindas da teoria de filas (Gross et al., 2009).

1.1 Motivação

Conforme já ressaltamos, é de importância para o gestor saber o comportamento do seu sistema baseado em filas, de modo que ele possa ser otimizado, reduzindo seus custos operacionais e tenha melhorado seu desempenho. Assim, a motivação para o desenvolvimento desta tese é responder questões tais como: Para qual tipo de problema certo modelo de filas é útil? Qual metodologia inferencial estatística podemos utilizar? Como obter as estimativas para as medidas de desempenho? Que tipo de dados podem ser usados para obter tais estimativas? Como podem ser coletados? Como avaliar a significância dos resultados? Estes e outros aspectos do processo inferencial em modelos de filas markovianas serão abordados nesta tese.

A literatura apresenta uma grande quantidade de artigos que descrevem estudos sobre estimação em modelos de filas, tanto por métodos inferenciais clássicos,

¹Embora neste trabalho seja dada preferência para a denominação *intensidade de tráfego*, note-se que às vezes ρ é chamado de *fator de utilização da estação de serviço*.

quanto por métodos bayesianos. Entre os métodos clássicos, pode-se citar por exemplo o trabalho de [Clarke \(1957\)](#), que apresenta estimadores de máxima verossimilhança e aproximações para estimar a taxa de chegada λ e o tempo de serviço μ . No artigo de [Aigner \(1974\)](#) é apresentada uma comparação de alguns estimadores assintóticos para os parâmetros do modelo $M/M/1$, isto é, filas markovianas de servidor único. Eles apresentam também variâncias assintóticas para os parâmetros do modelo. [Schruben & Kulkarni \(1982\)](#) mostram que estimar taxas de chegada e taxas de serviço resulta em uma diferença notável entre a distribuição do estado do modelo (parâmetros estimados) e a distribuição do estado do sistema real (parâmetros conhecidos). [Schruben & Kulkarni \(1982\)](#) também mostram problemas nos estimadores quando $\rho \geq 1$. [Zheng & Seila \(2000\)](#) exploram alguns dos problemas encontrados por [Schruben & Kulkarni \(1982\)](#) e apresentam algumas propostas para estimar o tempo de espera na fila, W_q , número de clientes na fila, L_q , tempo médio de espera no sistema, W , e o número médio de clientes no sistema, L , para filas $M/M/1$. [Kannan & Jabarali \(2014\)](#) descrevem uma aplicação de filas $M/M/1$ e apresentam estimativas de máxima verossimilhança para filas formadas durante dois períodos, de demanda normal e em período de férias. [Pereira et al. \(2015\)](#) mostram via simulações Monte Carlo que o estimador de máxima verossimilhança para ρ em filas $M/M/1$ é viciado, e propõem um estimador *bootstrap* para corrigir este vício, melhorando consequentemente estimativas de outras medidas de desempenho, tais como L e L_q . Por fim, ainda no contexto clássico, para aplicação em filas markovianas finitas, modelo $M/M/1/K$, tem-se o trabalho de [Alouf et al. \(2001\)](#), em que são apresentadas inferências para K (capacidade do sistema) e λ . Dois métodos são apresentados para tais estimativas.

Sob o enfoque bayesiano, um dos primeiros trabalhos foi desenvolvido por [Muddapur \(1972\)](#), que estendeu a metodologia desenvolvida por [Clarke \(1957\)](#) atribuindo distribuições de probabilidade para λ e μ . Combinando com a função

de verossimilhança conjunta para (λ, μ) , realizou o processo inferencial para filas $M/M/1$ e $M/M/\infty$, através do estimador de Bayes para ρ , λ e μ . Mais tarde, [Mcgrath et al. \(1987a;b\)](#) apresentam aproximações em um contexto mais subjetivo para filas $M/M/1$ e $M/M/1/K$. Eles apresentam também uma comparação entre os estimadores bayesianos e os estimadores clássicos. Ao final, [Mcgrath et al. \(1987b\)](#) utilizam uma medida de informação de Shannon para avaliar diferentes tipos de dados a partir do processo de filas.

[Armero \(1994\)](#) apresenta uma análise bayesiana e verifica a condição de ergodicidade para alguns modelos de filas. [Armero \(1994\)](#) também apresenta os resultados do estudo para filas $M/M/c$. [Armero & Bayarri \(1994a;b\)](#) estudam distribuições preditivas para estimar algumas medidas de desempenho usuais em filas $M/M/1$ e mostram também algumas distribuições *a priori* que podem ser utilizadas para realizar inferência para $\theta = (\lambda, \mu, \rho)$. Ao final, [Armero & Bayarri \(1994b\)](#) utilizam a metodologia em uma aplicação para dados reais e apresentam estimativas para L e W .

Tais trabalhos visaram apresentar uma metodologia inferencial que especifica distribuições para λ e μ , para encontrar assim um estimador para a intensidade do tráfego, dado por $\hat{\rho} = \hat{\lambda}/\hat{\mu}$, reforçando a necessidade de se conhecer ambas as taxas de chegada e de serviço dos clientes. Outra estratégia de coleta de dados, entretanto, é proposta por [Choudhury & Borthakur \(2008\)](#), em que são apresentados momentos da distribuição *a posteriori* e intervalos de credibilidade para ρ para filas $M/M/1$. O importante no trabalho de [Choudhury & Borthakur \(2008\)](#) é que ficou demonstrado que não é necessário conhecer as taxas λ e μ para inferência sobre algumas medidas de desempenho importantes. De fato, [Choudhury & Borthakur \(2008\)](#) mostram que, a partir do conhecimento da distribuição do número de clientes no sistema nos momentos de partida de um usuário, é possível obter várias medidas de desempenho. Um diagrama morfológico das abordagens mencionadas

é apresentado na Figura 1.1.

A proposta desta tese é desenvolver uma abordagem clássica e bayesiana como métodos de aproximação dos parâmetros de interesse em modelagem de fila. Os modelos considerados podem representar, dentre outros aspectos, sistemas de interesse prático. Na abordagem bayesiana e por meio de uma distribuição *a priori* sobre os parâmetros a serem estimados, inferências podem ser feitas, o que torna o modelo um pouco mais robusto e flexível.

1.2 Contribuições

Apresentar ao gestor um modelo que seja ao mesmo tempo flexível e robusto na hora de tomar decisão é de interesse prático. Sendo assim, as principais contribuições desta tese incluem:

1. Apresentação de um resumo sobre teoria de filas, assim como sobre os modelos mais usuais;
2. Demonstração de como a inferência clássica e bayesiana pode ser aplicada aos modelos de filas markovianas e ao processo de estimação dos parâmetros envolvidos;
3. Extensão do processo inferencial obtido por Choudhury & Borthakur (2008), utilizando outras distribuições *a priori*, assim como para modelos de filas $M/M/1/K$;
4. Desenvolvimento de um estudo de simulação a fim de verificar o comportamento das estimativas dos parâmetros e das medidas de desempenho da fila, a partir da distribuição *a priori*.

Neste aspecto, demonstra-se ao gestor como o conhecimento *a priori* pode

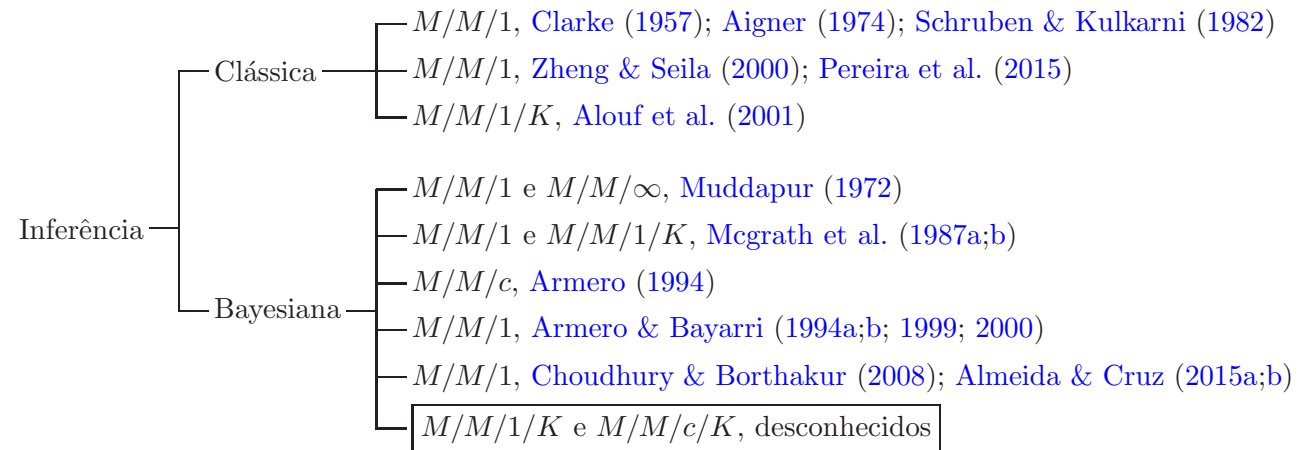


Figura 1.1: Diagrama morfológico das abordagens inferenciais para filas markovianas

contribuir para inferir sobre a intensidade do tráfego, assim como para as outras medidas de desempenho.

1.3 Organização

Esta tese encontra-se assim organizada. No Capítulo 2 apresentamos itens fundamentais, com uma revisão da literatura da área e as metodologia inferenciais clássica e bayesiana, para filas markovianas. No Capítulo 3 está descrita a metodologia desenvolvida, com o detalhamento dos estimadores de máxima verossimilhança, das distribuições *a posteriori* e do método para correção de vício via *bootstrap* de estimadores para ρ , L e L_q . Os resultados de extensivos experimentos computacionais são apresentados e comentados no Capítulo 4. O Capítulo 5 encerra esta tese, com as principais conclusões obtidas e a discussão de tópicos para futuras pesquisas na área de inferência em filas.

Capítulo 2

Itens Fundamentais

2.1 Introdução à Teoria de Filas

Segundo [Medhi \(2003\)](#), a teoria de filas surgiu a partir de 1909 quando Agner Krarup Erlang publicou um artigo sobre o congestionamento no tráfego telefônico. Desde então, a literatura aborda diversas metodologias sobre o assunto. Do ponto de vista teórico, sabe-se que é um ramo da matemática que analisa o comportamento das filas.

Entretanto, por que as filas se formam? A resposta está no fato de existir uma demanda maior pelo serviço do que recursos para realiza-los. Isso ocorre porque existem poucos servidores. Isto é, há indisponibilidade em oferecer um serviço de tal nível que evitasse a formação de filas. Há situações em que há limitação do espaço físico, tanto para alocar mais servidores, quanto para comportar uma quantidade maior de clientes. Há ainda questões financeiras, tais como restrições orçamentárias, que impedem investimentos em espaço físico, infraestrutura ou em pessoal.

Do ponto de vista prático, quando um cliente procura por um determinado serviço, há então um sistema formado por cliente e servidor. Como exemplos, há

clientes que vão até ao banco, processos que chegam para serem analisados ou então chamadas telefônicas que necessitam ser completadas. Assim, quando a demanda para executar o serviço excede a capacidade de o sistema atender ao cliente até um determinado tempo, há um processo de formação de uma fila de espera. Assim, um sistema de filas é composto por clientes (ou unidades), e servidor (também chamado de canal de serviço).

Como forma de aferir o comportamento do sistema de filas, associam-se medidas de desempenho, tais como o tempo médio de espera dos clientes na fila, tempo médio entre chegada de clientes, a probabilidade de encontrar o sistema vazio, entre muitas outras. Dessa forma, a teoria de filas procura encontrar, através de análises matemáticas detalhadas, um ponto de equilíbrio que satisfaça ao cliente (ou linha de produção) e seja viável economicamente para o provedor do serviço. Conhecer a terminologia empregada nos estudos de sistemas de filas é o primeiro passo no estudo dessa que é uma das áreas da pesquisa operacional.

Um sistema de fila é formado basicamente por seis características, quais sejam: (i) o processo de chegada, (ii) a distribuição do tempo de serviço, (iii) o número de servidores, (iv) a capacidade do sistema, (v) a população de usuários e (vi) a disciplina de atendimento. Faremos uma descrição de cada uma delas, em que os três primeiros são obrigatórios e os três últimos, se não informados, são considerados conhecidos.

2.1.1 Padrão de Chegada dos Clientes

O padrão de chegada no sistema de fila é, às vezes, definido em termos do número médio de chegadas por unidade de tempo (taxa média de chegada). Assim, o modelo de chegada é usualmente especificado pelo tempo entre as chegadas dos usuários. Pode ser determinístico, isto é, as chegadas ocorrem em intervalos de tempo exatamente iguais (tempo entre as chegadas é constante), ou ser uma variá-

vel aleatória, quando o tempo entre as chegadas é variável e segue uma distribuição de probabilidades presumivelmente conhecida. Além de sabermos se o modelo de chegada é determinístico ou é uma variável aleatória, precisamos também saber a taxa de chegada λ . A constante λ é a taxa média de chegadas dos usuários por unidade de tempo.

2.1.2 Padrão de Serviço

O padrão de serviço (ou padrão de atendimento aos usuários) é normalmente especificado pelo tempo de serviço, isto é, o tempo requerido pelo servidor para concluir o atendimento. Da mesma forma que o padrão de chegada, pode ser determinístico (constante) ou uma variável aleatória (quando o tempo de atendimento é variável e segue uma distribuição de probabilidades presumivelmente conhecida). Neste último caso, valem as mesmas considerações feitas à distribuição de probabilidades associada ao padrão de chegada dos clientes do serviço. A constante μ é a taxa média de atendimentos por unidade de tempo, por servidor.

2.1.3 Número de Servidores

O número de servidores (ou número de canais de serviço) é o número de atendentes disponíveis no sistema e que podem atender simultaneamente aos clientes. A Figura 2.1 ilustra um sistema de fila única e canal simples, enquanto as Figuras 2.2 e 2.3 mostram duas variações dos sistemas multicanais. Os dois sistemas multicanais diferem pelo fato de que o primeiro possui uma única fila, enquanto o segundo possui uma fila para cada canal. Uma barbearia com várias cadeiras é um exemplo do primeiro tipo de multicanal, assumindo que não exista um estilo particular de corte de cabelo. Por outro lado, um supermercado e um restaurante *fast-food* preenche a segunda categoria de sistema multicanal. É geralmente assumido que os mecanismos de canais paralelos operam independentemente um do

outro.

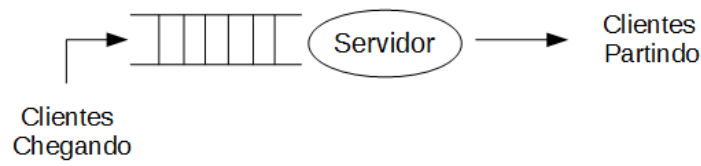


Figura 2.1: Sistema com canal simples com fila única

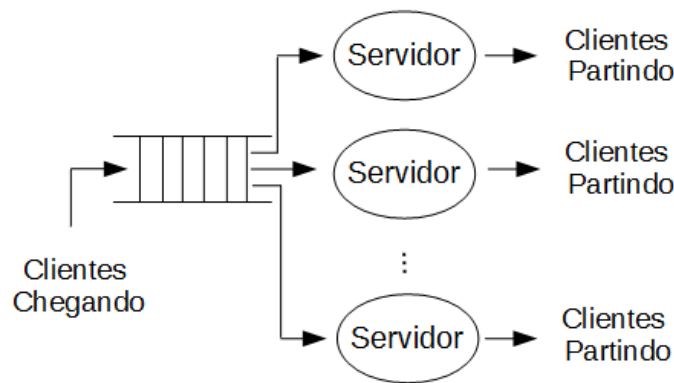


Figura 2.2: Sistema multicanal com fila única

2.1.4 Capacidade do Sistema e Estágios de Serviço

Em alguns processos de filas existe uma limitação física da quantidade de espaço na fila, de modo que, se as filas alcançarem um certo comprimento, nenhum novo cliente poderá entrar no sistema até que um espaço seja obtido com o atendimento de um cliente e a conseqüente diminuição do tamanho da fila. Estas situações são referidas como sistemas de filas finitos, ou seja, existe um limite finito no tamanho do sistema.

Um sistema de filas também pode ter um único estágio de serviço, como no caso da barbearia, ou pode ter vários estágios. Um sistema de multi-estágio pode

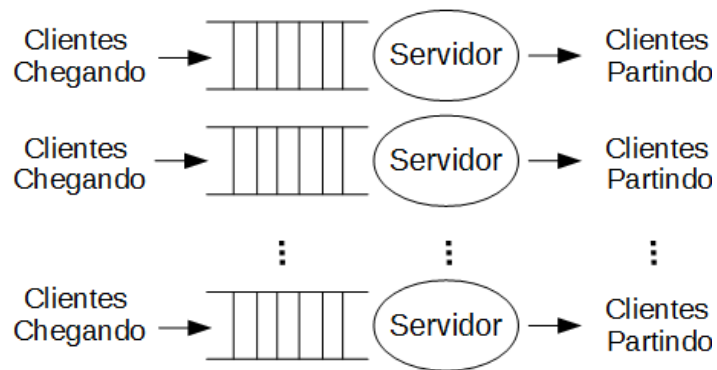


Figura 2.3: Sistema multicanal com filas em paralelo

ser exemplificado como um procedimento de exame físico, em que cada paciente passa por diversos exames, tais como, sangue, oftalmológico, urina *etc.* Em alguns sistemas multi-estágio, reciclagem (ou retorno) podem ocorrer. Reciclagem é comum em processos de manufatura, em que inspeções de controle de qualidade são realizadas, sendo que alguma peça que não se adeque ele deve ser reprocessada. Um exemplo de sistema multiestágio com retorno é mostrado na Figura 2.4.

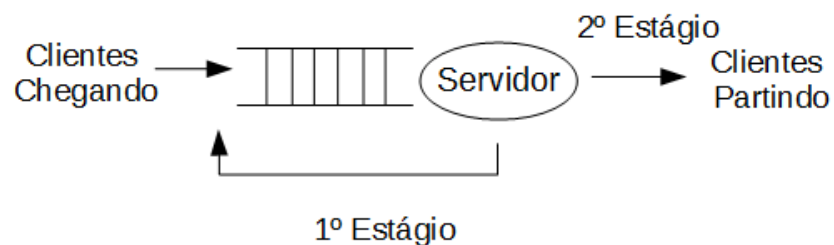


Figura 2.4: Sistema com retroalimentação (*feedback*)

2.1.5 População de Usuários

Aqui retrata-se o número potencial de clientes que podem chegar a um sistema, que pode ser finito ou infinito.

2.1.6 Disciplina da Fila

A disciplina de atendimento refere-se à maneira pela qual os clientes são selecionados para o serviço quando a fila está formada. A disciplina mais comum que tem sido observada é ‘primeiro a chegar, primeiro a ser servido’ (ou disciplina FIFO, do inglês ‘*First-In, First-Out*’). Outra disciplina comum de atendimento é ‘último a chegar, primeiro a ser servido’ (ou disciplina LIFO, do inglês ‘*Last-In, First-Out*’), que se aplica a muitos sistemas de estoque, quando não há obsolescência de unidades armazenadas, uma vez que é mais fácil atingir os itens mais próximos, no caso o último que chegou.

2.2 Processos de Nascimento-e-Morte

Antes de introduzirmos os modelos de filas, é necessário fazer uma breve introdução sobre um processo interessante que ocorre em uma fila. É o comportamento do número de clientes no sistema. Sob o ponto de vista de um processo de Markov, é conhecido como ‘processo de nascimento-e-morte’. Corresponde ao fato de que, em sistema de filas, eventos, como chegadas e partidas, estão acontecendo.

Os termos ‘nascimento’ e ‘morte’ representam respectivamente a chegada do cliente no sistema e sua saída. Assim, se a população é n , então λ_n e μ_n representam a transição desse processo. Assim, quando a população é o número de clientes no sistema, λ_n e μ_n indicam as taxas de chegada e de serviço, que dependem do número de clientes no sistema. Baseado nas propriedades do processo de Poisson, ou seja, quando as chegadas são um processo de Poisson e o tempo de serviço é exponencial,

é possível fazer as seguintes afirmações (Bhat, 2008), sobre as probabilidades de transição durante o tempo $(t, t + \Delta t]$:

- Nascimento ($n \geq 0$):

$$P(\text{um nascimento}) = \lambda_n \Delta t + o(\Delta t),$$

$$P(\text{nenhum nascimento}) = 1 - \lambda_n \Delta t + o(\Delta t),$$

$$P(\text{mais que um nascimento}) = o(\Delta t);$$

- Morte ($n \geq 0$):

$$P(\text{um nascimento}) = \mu_n \Delta t + o(\Delta t),$$

$$P(\text{nenhum nascimento}) = 1 - \mu_n \Delta t + o(\Delta t),$$

$$P(\text{mais que um nascimento}) = o(\Delta t),$$

em que $o(\Delta t)$ é tal que $o(\Delta t)/\Delta t \rightarrow 0$, quando $\Delta t \rightarrow 0$, em que em cada um dos dois casos, o termo $o(\Delta t)$ deve somar 0, para que a probabilidade total dos três eventos seja igual a 1.

Seja $Q(t)$ o número de clientes no sistema até o tempo t . Assim define-se:

$$P_{in}(t) = P[Q(t) = n | Q(0) = i].$$

Incorporando as probabilidades para as transições durante $(t, t + \Delta t)$, como mencionado acima, obtém-se:

$$P_{n,n+1}(\Delta t) = \lambda_n \Delta t + o(\Delta t), \quad n = 0, 1, 2, \dots,$$

$$P_{n,n-1}(\Delta t) = \mu_n \Delta t + o(\Delta t), \quad n = 1, 2, 3, \dots,$$

$$P_{n,n}(\Delta t) = 1 - \lambda_n \Delta t - \mu_n \Delta t + o(\Delta t), \quad n = 1, 2, 3, \dots,$$

$$P_{n,j}(\Delta t) = o(\Delta t), \quad j \neq n - 1, n, n + 1.$$

Assim, derivando-se os termos no lado direito dessas equações, é possível encontrar as taxas de transição infinitesimais, o que leva à seguinte matriz geradora, para o processo de nascimento-e-morte do modelo do sistema de filas:

$$\mathbf{A} = \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & \dots \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & 0 & \dots \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & \dots \\ 0 & 0 & \mu_3 & -(\lambda_3 + \mu_3) & \dots \\ 0 & 0 & 0 & \mu_4 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (2.1)$$

A matriz geradora \mathbf{A} conduz às equações de Kolmogorov para $P_{in}(t)$ (veja [Bhat, 2008](#), para mais detalhes). Pode-se então rescrever $P_{in}(t) \equiv P_n(t)$ e inserir o estado inicial i somente quando for necessário. Assim, reescrevem-se as seguintes equações:

$$P'_0(t) = -\lambda_0 P_0(t) + \mu_1 P_1(t), \quad (2.2)$$

$$P'_n(t) = -(\lambda_n + \mu_n)P_n(t) + \lambda_{n-1}P_{n-1}(t) + \mu_{n+1}P_{n+1}(t), \quad n = 1, 2, \dots \quad (2.3)$$

Para determinar $P_n(t)$, deve ser resolvido o sistema das Equações (2.2) e (2.3), juntamente com a condição inicial $P_i(0) = 1$, $P_n(0) = 0$, para $n \neq i$. O mais importante aqui é analisar os limites das Equações (2.2) e (2.3), quando $t \rightarrow \infty$. Um resultado geral em processos de Markov é (Bhat, 2008):

Teorema 2.1 1. *Se o processo de Markov é irredutível (todos os estados), então a distribuição limite $\lim_{t \rightarrow \infty} P_n(t) = p_n$ existe e é independente das condições iniciais do processo. Os limites $\{p_n, n \in \mathbf{S}\}$ são tais que ambos desaparecem de forma idêntica (isto é, $p_n = 0$, para todo $n \in \mathbf{S}$) ou são todos positivos e formam uma distribuição de probabilidade (isto é, $p_n > 0$ para todo $n \in \mathbf{S}$, e $\sum_{n \in \mathbf{S}} p_n = 1$).*

2. *A distribuição limite $\{p_n, n \in \mathbf{S}\}$ de um processo de Markov recorrente irredutível é dado por uma única solução da equação $\mathbf{pA} = 0$ e $\sum_{j \in \mathbf{S}} p_j = 1$, em que $\mathbf{p} = (p_0, p_1, p_2, \dots)$.*

Os resultados apresentados no Teorema 2.1 são essenciais para o estado de equilíbrio no processo estocástico. No estado de equilíbrio, também conhecido como *estado estacionário*, o comportamento do processo é independente do tempo e do valor inicial, isto é,

$$\lim_{t \rightarrow \infty} P_{in}(t) = p_n, \quad n = 0, 1, 2, \dots$$

e, conseqüentemente, temos que $P'_n(t) \rightarrow 0$ quando $t \rightarrow \infty$.

Com isso, usando este resultado na Eq. (2.3), tem-se:

$$0 = -\lambda_0 p_0 + \mu_1 p_1, \quad (2.4)$$

$$0 = -(\lambda_n + \mu_n)p_n + \lambda_{n-1}p_{n-1} + \mu_{n+1}p_{n+1}, \quad n = 1, 2, \dots \quad (2.5)$$

Estas equações podem ser facilmente resolvidas através do método recursivo. Rearranjando o primeiro termo na Equação (2.4), tem-se:

$$p_1 = \frac{\lambda_0}{\mu_1}p_0. \quad (2.6)$$

Para $n = 1$, a segunda equação é dada por

$$(\lambda_1 + \mu_1)p_1 = \lambda_0p_0 + \mu_2p_2 \Rightarrow p_2 = \frac{\lambda_1\lambda_0}{\mu_2\mu_1}p_0. \quad (2.7)$$

Assim, continuando o método recursivo para $n = 2, 3, \dots$, tem-se:

$$\mu_n p_n = \lambda_{n-1} p_{n-1} \Rightarrow p_n = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_0 \mu_1 \dots \mu_n} p_0. \quad (2.8)$$

Logo, pelo Teorema 2.1, a condição de normalização é aplicada, $\sum_{n \in \mathbf{S}} p_n = 1$, encontrando-se assim:

$$p_0 = \left[1 + \sum_{n=1}^{\infty} \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_0 \mu_1 \dots \mu_n} \right]^{-1}. \quad (2.9)$$

Com isso, a distribuição limite do estado de um modelo de fila considerando um processo de nascimento-e-morte $\{p_n, n = 0, 1, 2, \dots\}$ é dada pelas Equações (2.8) e (2.9). Por fim, o Teorema 2.2, apresentando em Bhat (2008), garante umas das propriedades da distribuição limite do processo de Markov, que trata sobre a estacionariedade do processo.

Teorema 2.2 *A distribuição limite de um processo de Markov irredutível recorrente positivo é também estacionário.*

Com isso, um processo é dito ser *estacionário* se a distribuição de estado é independente do tempo. Portanto, se $P_n(0) = p_n, n = 0, 1, 2, \dots$, então tem-se $P_n(t) = p_n, \forall t$.

2.3 Inferência Clássica

A inferência clássica apresenta várias maneiras de realizar estimação pontual. Dentre elas, podemos citar o método de estimação através da função de verossimilhança dos dados, também chamado de estimador da máxima verossimilhança (EMV). Esta maneira geralmente é preferida em relação a outras porque apresenta melhores propriedades de eficiência. No entanto, há situações em que o EMV não pode ser encontrado de maneira analítica.

Esta técnica foi desenvolvida pelos anos de 1920 pelo estatístico Sir Fisher, sendo um dos melhores métodos para obter um estimador de um parâmetro. Aqui o estimador será o valor do parâmetro que maximize a função de verossimilhança (Montgomery & Runger, 2012).

Suponha que X seja uma variável aleatória que tenha distribuição de probabilidade dada por $f(x, \theta)$, em que θ é um parâmetro desconhecido. Se $x = (x_1, x_2, \dots, x_n)^T$, são os n valores amostrais observados, a função verossimilhança desta amostra é dada por

$$L(\theta; x) = f(x_1; \theta) \times f(x_2; \theta) \times \dots \times f(x_n; \theta). \quad (2.10)$$

A função de verossimilhança $L(\theta; x)$ é agora uma função somente do parâmetro desconhecido θ . Logo, o EMV para θ é um valor $\hat{\theta}$ que maximiza esta função de verossimilhança $L(\theta; x)$.

2.4 Correção *Bootstrap* de Vício

Um método utilizado para correção de vício em estimadores é o *bootstrap* (Efron & Tibshirani, 1993). Na sua versão não paramétrica, para estimar o vício de um parâmetro de interesse, θ , diversas reamostragens (com reposição) são realizadas e o parâmetro é reestimado para cada reamostragem. Tira-se então a média

de tais estimativas, $\hat{\theta}_{(\bullet)}$. O vício pode então ser estimado da seguinte forma:

$$\text{vício} = \hat{\theta}_{(\bullet)} - \hat{\theta}. \quad (2.11)$$

Uma versão corrigida do estimador, pelo método *bootstrap* não paramétrico, pode ser obtida por:

$$\tilde{\theta}_B = 2\hat{\theta} - \hat{\theta}_{(\bullet)}. \quad (2.12)$$

Este método foi usado por diversos pesquisadores no passado, com resultados bastante satisfatórios em correção de vício (Cruz et al., 2004), construção de intervalos de confiança (Domingues et al., 2015), entre outros.

2.5 Inferência Bayesiana

A metodologia bayesiana constitui uma outra forma para se realizar inferência. Uma das principais diferenças em relação à metodologia clássica é que na inferência bayesiana é permitida a incorporação da informação *a priori* sobre os parâmetros desconhecidos do modelo. Além disso, ao contrário dos métodos clássicos, os métodos bayesianos consideram os parâmetros como variáveis aleatórias, associando a eles na maioria das vezes uma distribuição de probabilidade. Assim o conhecimento que o gestor tem sobre um determinado parâmetro desconhecido pode ser levado em consideração.

Suponha que se deseja estimar o valor do parâmetro desconhecido. Segundo Ehlers (2003), a informação *a priori* que se tem sobre este parâmetro é de fundamental importância. Do ponto de vista bayesiano a incerteza sobre o valor desconhecido pode ser quantificada a partir de uma distribuição de probabilidade definida em um espaço paramétrico, representado por Θ . Esta informação *a priori* é representada na maioria das vezes por uma distribuição de probabilidade

chamada de distribuição *a priori*.

Conforme lembra [Paulino et al. \(2003\)](#), a informação inicial depende apenas do conhecimento do pesquisador sobre o problema estudado. Logo, pode-se ter diferentes distribuições (ou modelos) *a priori* associadas ao parâmetro θ , representadas por $p(\theta)$, e definidas em Θ . Assim, coleta-se uma amostra aleatória da variável de interesse X , em que a distribuição amostral pode ser representada por $p(x|\theta)$ ou $L(\theta; x)$, denominada de função de verossimilhança. Combina-se então a distribuição *a priori* com a função de verossimilhança, a partir do Teorema de Bayes, a fim de encontrar uma distribuição *a posteriori* $p(\theta|x)$ para θ , encontrada por:

$$p(\theta|x) = \frac{p(\theta, x)}{p(x)} = \frac{p(x|\theta)p(\theta)}{p(x)}, \quad (2.13)$$

em que $1/p(x)$ é uma constante normalizadora da distribuição *a posteriori* $p(\theta|x)$.

Assim, a distribuição *a posteriori*, na Equação (2.13), representa um modelo com informação sobre θ , atualizado pelos dados. A distribuição *a posteriori* é de suma importância na abordagem bayesiana, pois é ela que será utilizada para a realização de todas as inferências, como estimação pontual, construção de regiões de credibilidade, ou intervalos HPD (*Highest Posterior Density*), e teste de hipóteses. Diferentemente da abordagem inferencial clássica, na abordagem bayesiana considera-se que θ é uma variável aleatória e a observação amostral é fixa.

Com o conhecimento prévio sobre o parâmetro θ , pode-se construir uma família paramétrica de densidades, de modo que as distribuições *a priori* e *a posteriori* pertençam à mesma classe de distribuições, isto é, sejam conjugadas. Neste caso, a distribuição *a posteriori* tem a mesma distribuição *a priori*, com os hiperparâmetros da distribuição atualizados pela informação que a amostra (função de verossimilhança) contém. Dentre algumas distribuições que apresentam conjugação, tem-se, por exemplo, a distribuição beta, que é conjugada natural da família

Bernoulli. Para maiores detalhes sobre esta e outras classes de distribuições *a priori*, ver [Paulino et al. \(2003\)](#).

Em algumas situações, o pesquisador tem pouca informação sobre o parâmetro desconhecido, ou até mesmo nenhuma informação, tanto para o parâmetro θ , quanto para a forma da distribuição. Nestes casos, é comum utilizar as chamadas distribuições *a priori* nulas ou não-informativas. Assim, espera-se que a informação contida na amostra seja dominante no processo de inferência. Para esses casos, representa-se a distribuição *a priori* por uma constante. Alguns métodos foram desenvolvidos de modo a representar o máximo possível a informação *a priori*, por mais que ela seja vaga ou nula, dentre eles pode-se citar o método de Bayes-Laplace, Box-Tiao e *a priori* não informativa de Jeffreys (ver [Paulino et al., 2003](#)).

2.6 Observações Finais

Neste capítulo vimos alguns itens que são fundamentais para o entendimento de um sistema de filas e para inferências sobre seus parâmetros. É de suma importância o conhecimento do processo, uma vez que o padrão de tal processo irá determinar qual modelo deve ser utilizado. Também vimos fundamentos de inferência clássica e bayesiana, bem como de correção de vício via *bootstrap*. No próximo capítulo veremos a metodologia inferencial proposta para estimação de ρ , L e L_q em modelos de filas markovianas de servidor único finitas e infinitas.

Capítulo 3

Metodologia Inferencial

3.1 Filas $M/M/1$

3.1.1 Introdução

O modelo de filas markovianas de servidor único, também conhecido como $M/M/1$ na notação de [Kendall \(1953\)](#), baseia-se nas características dos processos de chegada e de serviço (atendimento) assumidos markovianos. Assim, assume-se que o número de chegadas na unidade de tempo segue uma distribuição de Poisson, com taxa λ , e o tempo de serviço segue uma distribuição exponencial, com taxa μ . Em outras palavras, são filas com:

- densidade para os tempos entre chegadas: $f(t) = \lambda e^{-\lambda t}$, $t \geq 0$,
 - densidade para os tempos de serviço: $g(t) = \mu e^{-\mu t}$, $t \geq 0$,
- (3.1)

em que:

$$\begin{aligned} E[\text{tempo entre as chegadas}] &= \frac{1}{\lambda} = \frac{1}{\text{taxa de chegada}}. \\ E[\text{tempo de serviço}] &= \frac{1}{\mu} = \frac{1}{\text{taxa de serviço}}. \end{aligned} \tag{3.2}$$

A razão entre as taxas de chegada e de serviço, também conhecida por *intensidade de tráfego*, é uma importante medida de desempenho da fila e é definida como:

$$\rho \equiv \frac{\text{taxa de chegada}}{\text{taxa de serviço}} = \frac{\lambda}{\mu}. \quad (3.3)$$

Observa-se que as filas $M/M/1$ são um caso especial de um modelo geral de nascimento-e-morte em que $\lambda_n = \lambda$ e $\mu_n = \mu$. Assim, rearranjando-se a Equação (2.5), tem-se que:

$$\lambda p_0 = \mu p_1, \quad (3.4)$$

$$(\lambda + \mu)p_n = \lambda p_{n-1} + \mu p_{n+1}, \quad n = 1, 2, \dots \quad (3.5)$$

Resolvendo-se este sistema de equações e usando-se o fato de que $\sum_0^\infty p_n = 1$, encontra-se:

$$p_n = \rho^n(1 - \rho), \quad n = 0, 1, \dots, \quad \text{para } \rho = \lambda/\mu < 1. \quad (3.6)$$

Note-se que a Equação (3.6) caracteriza uma distribuição de probabilidade geométrica, que depende somente do quociente da taxa de chegada e de atendimento, λ/μ , isto é, da intensidade de tráfego, ρ . A probabilidade de o servidor estar ocupado é também uma medida importante no sistema de filas, que pode ser calculada por $\rho = 1 - p_0$. Com relação a ρ , Wagner (1986) lembra a importância de que se verifique a relação $\rho < 1$ (ou, equivalentemente, $\lambda < \mu$), para garantir-se que o sistema atinja o estado estacionário. Caso contrário, a fila só aumentaria.

Associada a esta intensidade de tráfego, podem-se encontrar algumas outras importantes medidas de desempenho para o modelo $M/M/1$, dado que a Equação (3.6) caracteriza a distribuição do número de clientes no sistema no momento

da partida, como reforçado por [Choudhury & Borthakur \(2008\)](#). Assim, lembrando que $Q(t)$ é o número de clientes no sistema, fazendo $Q(\infty) = Q$ e Q_q o número de clientes na fila, excluindo-se o que está em serviço, tem-se que $L = E(Q)$ é o número esperado de clientes no sistema, $L_q = E(Q_q)$ é o número esperado de clientes na fila e $V(Q)$ é a variância do número de clientes no sistema. Logo, a partir da Eq. (3.6), tem-se respectivamente que:

$$\begin{aligned} L &= \sum_{n=1}^{\infty} n(1-\rho)\rho_n \\ &= \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1 - \rho}, \end{aligned} \tag{3.7}$$

$$\begin{aligned} L_q &= \sum_{n=1}^{\infty} (n-1)p_n \Rightarrow \\ &= \sum_{n=1}^{\infty} np_n - \sum_{n=1}^{\infty} p_n \Rightarrow \\ &= \frac{\lambda}{\mu - \lambda} - \frac{\lambda}{\mu} \Rightarrow \\ &= \frac{\lambda^2}{\mu(\mu - \lambda)} \\ &= \frac{\rho^2}{1 - \rho}, \end{aligned} \tag{3.8}$$

e

$$\begin{aligned} V(Q) &= \frac{\lambda\mu}{(\mu - \lambda)^2} \\ &= \frac{\rho}{(1 - \rho)^2}. \end{aligned} \tag{3.9}$$

3.1.2 Inferência Clássica

A metodologia apresentada a seguir é descrita por [Almeida & Cruz \(2015a\)](#), que estende o trabalho de [Choudhury & Borthakur \(2008\)](#). Considerando-se a suposição de estado estacionário, ou seja, que $\lambda < \mu$, um resultado apresentado em teoria de fila é a fórmula de Pollaczek-Khinchine, conforme pode ser encontrado no artigo de [Medhi \(2003\)](#). Esta fórmula garante que, na Equação (3.6), tem-se uma variável aleatória não negativa que representa o número de clientes no sistema no momento da partida de um usuário do sistema, que segue uma distribuição geométrica. Assim, após o equilíbrio, o número de clientes no sistema no momento da partida, M , é dado por:

$$P(M = m) = \begin{cases} \rho^m(1 - \rho), & m = 0, 1, 2, \dots, \\ 0, & \text{caso contrário.} \end{cases} \quad (3.10)$$

Portanto, para se obter dados desta natureza, faz-se necessário observar o sistema no momento das partidas dos usuários. Considere-se então uma amostra aleatória, $x = (x_1, x_2, \dots, x_n)^T$, que representa o número de clientes que ficaram no sistema em n partidas de usuários, em momentos aleatórios. Um fato importante descrito por [Choudhury & Borthakur \(2008\)](#) é que se X_m representa o número de clientes até a partida do m -ésimo usuário, então $p_{ij}^{(k)} = P(X_{m+k} = j | X_m = i)$. A propriedade da ergodicidade garante os limites das probabilidades dadas por $v_j = \lim_{k \rightarrow \infty} p_{ij}^{(k)}$, $j = 0, 1, \dots$. Assim, v_j , $j = 0, 1, \dots$, existem e são independentes do estado inicial i , desde que o sistema esteja no estado estacionário. Com isso, a função de verossimilhança pode ser escrita como:

$$L(\rho; x) = \rho^y(1 - \rho)^n, \quad (3.11)$$

em que $y = \sum_{i=1}^n x_i$. Este modo de geração de dados garante a independência das observações amostrais, levando em conta a propriedade de ergodicidade da cadeia

de Markov.

Derivando-se a Eq. (3.11) e igualando a zero, encontra-se o estimador de máxima verossimilhança (EMV) para ρ :

$$\hat{\rho} = \frac{y}{n+y} = \frac{1}{n/y+1}. \quad (3.12)$$

Uma das características mais atrativas do método de máxima verossimilhança é sua *invariância* a transformações (Mukhopadhyay, 2000). Ou seja, se $\hat{\theta}$ é o EMV de θ e $\alpha = h(\theta)$ é uma função de θ , então $\hat{\alpha} = h(\hat{\theta})$ é o EMV de α . Desta forma, são os seguintes os EMV para o número esperado de clientes e tamanho médio da fila, respectivamente:

$$\hat{L} = \frac{\hat{\rho}}{1-\hat{\rho}} = \frac{1}{1/\hat{\rho}-1}, \quad (3.13)$$

e

$$\hat{L}_q = \frac{\hat{\rho}^2}{1-\hat{\rho}}. \quad (3.14)$$

As implementações dos estimadores para ρ , L e L_q , em R (R Core Team, 2013), podem ser visualizadas na Listagem 3.1.

```

1 EMVRo<-function(amostra) {
  return(1/(1+length(amostra)/sum(amostra)))
3 }
EMVI<-function(amostra) {
5   return(1/(1/EMVRo(amostra)-1))
  }
7 EMVLq<-function(amostra) {
  roHat<-EMVRo(amostra)
9   return(roHat^2/(1-roHat))
  }

```

Listagem 3.1: Implementação do EMV para ρ , L e L_q

3.1.3 Correção de Vício via *Bootstrap*

A implementação em R (R Core Team, 2013) do EMV corrigido via *bootstrap* pode ser visualizada na Listagem 3.2.

```
ECBoot<-function(amostra, fEst) {  
2   B<-10  
   soma<-0  
4   for (i in 1:B) {  
       reamostra<-sample(amostra, replace=T)  
6       estm<-fEst(reamostra)  
       soma<-soma+estm}  
8   estmStar=soma/B  
   return(2*fEst(amostra)-estmStar)  
10 }
```

Listagem 3.2: Implementação do EMV corrigido via *bootstrap*

3.1.4 Inferência Bayesiana

Para fazer inferência sobre ρ , partindo da Equação (3.11), vários tipos de informações *a priori* podem ser consideradas, em termos de distribuições. Consideraremos aqui uma que é conjugada natural e duas outras não-informativas (Zacharias et al., 2004). Maiores detalhes sobre as distribuições *a priori* não-informativas podem ser encontrados facilmente na literatura da área (consultar, por exemplo, Paulino et al., 2003, ou referências por eles citadas).

Priori Conjugada Beta

Esta opção garante que distribuições *a priori* e *a posteriori* pertençam à mesma família paramétrica. Assim, a distribuição beta com parâmetros a e b pode ser utilizada (Choudhury & Borthakur, 2008), ou seja, $\rho \sim \text{beta}(a, b)$. Esta é uma

distribuição flexível e que pode assumir diversas formas, além de ser uma conjugada natural da Equação (3.11). Em outras palavras, tem-se que:

$$p_1(\rho) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \rho^{a-1}(1-\rho)^{b-1} = \frac{\rho^{a-1}(1-\rho)^{b-1}}{B(a,b)}, \quad (3.15)$$

para $0 < \rho < 1$, $a > 0$, $b > 0$, em que $B(a,b) \equiv \int_0^1 \rho^{a-1}(1-\rho)^{b-1} d\rho$ é a função beta (Abramowitz & Stegun, 1972).

Assim, é possível encontrar a distribuição *a posteriori* de ρ combinando-se as Equações (2.13), (3.11), e (3.15):

$$\begin{aligned} p_1(\rho|x) &= \frac{L(\rho; x) \times p_1(\rho)}{\int_0^1 L(\rho; x) \times p_1(\rho) d\rho} \\ &= \frac{\rho^y(1-\rho)^n \times \left[\frac{\rho^{a-1}(1-\rho)^{b-1}}{B(a,b)} \right]}{\int_0^1 \rho^y(1-\rho)^n \times \left[\frac{\rho^{a-1}(1-\rho)^{b-1}}{B(a,b)} \right] d\rho}, \end{aligned}$$

para $0 < \rho < 1$, $a > 0$, $b > 0$, com $y = \sum_{i=1}^n x_i$.

Assim, após cancelar a constante $B(a,b)$ e rearranjar os termos, tem-se que:

$$p_1(\rho|x) = \begin{cases} \frac{\rho^{y+a-1}(1-\rho)^{n+b-1}}{B(y+a, n+b)}, & 0 < \rho < 1, \\ 0, & \text{caso contrário,} \end{cases} \quad (3.16)$$

lembrando-se que $B(y+a, n+b) = \int_0^1 \rho^{(y+a)-1}(1-\rho)^{(n+b)-1} d\rho$ é a definição da função beta, com parâmetros $y+a$ e $n+b$, resultando portanto numa distribuição *a posteriori* que é também uma beta, com parâmetros $y+a$ e $n+b$, $\rho|x \sim \text{beta}(y+a, n+b)$, a partir da qual todas as estimativas podem ser encontradas (Choudhury & Borthakur, 2008).

De fato, assumindo-se uma função de perda quadrática para o erro e tomando-se a média da função de densidade de probabilidade $\text{beta}(y+a, n+b)$, Equ-

ção (3.16), é possível deduzir os estimadores pontuais bayesianos para ρ , L e L_q , dados respectivamente por:

$$\hat{\rho}_1 = \frac{y + a}{y + a + n + b}, \quad (3.17)$$

$$\hat{L}_1 = \frac{y + a}{n + b}, \quad (3.18)$$

e

$$\hat{L}_{q1} = \frac{(y + a + 1)(y + a)}{(n + b - 1)(y + a + n + b)}. \quad (3.19)$$

Se, após observar a amostra x , existir um interesse em prever o número (M) de clientes no sistema no momento da partida, também relacionada com ρ e descrito probabilisticamente por $P(M = m|\rho)$, então define-se a distribuição preditiva *a posteriori* $P(M = m|x)$ (Ehlers, 2003). Pode-se então analisar a distribuição preditiva para o número de clientes no sistema no momento da partida, em estado estacionário, dada por:

$$P_1(M = m|x) = \frac{B(y + a + m, n + b + 1)}{B(y + a, n + b)}, \text{ para } m = 1, 2, \dots, \quad (3.20)$$

A implementação em R (R Core Team, 2013) dos estimadores bayesianos, para ρ , L e L_q , e da distribuição preditiva, para a *priori* beta, pode ser vista na Listagem 3.3.

***Priori* Uniforme**

Se não existe uma informação palpável *a priori* a respeito de ρ , uma distribuição uniforme poderia ser utilizada. Como temos que ρ pertence ao intervalo $(0, 1)$, a uniforme poderia ser definida nestes limites, mas em muitos sistemas os gerentes

```

EPRo1<-function(amostra , a , b) {
2  n<-length(amostra)
  y<-sum(amostra)
4  return((y+a)/(y+a+n+b))
  }
6 EPL1<-function(amostra , a , b) {
  n<-length(amostra)
8  y<-sum(amostra)
  return((y+a)/(n+b))
10 }
EPLq1<-function(amostra , a , b) {
12  n<-length(amostra)
  y<-sum(amostra)
14  return(((y+a+1)*(y+a)/((n+b-1)*(y+a+n+b))))
  }
16 PPl<-function(amostra , a , b , m) {
  n<-length(amostra)
18  y<-sum(amostra)
  return(beta(y+a+m, n+b+1)/beta(y+a , n+b))
20 }

```

Listagem 3.3: Implementação dos estimadores bayesianos e da preditiva para a *priori* beta

operacionais podem oferecer limites mais justos (Choudhury & Borthakur, 2008), de tal forma que $0 < c < \rho < d < 1$. Ou seja, apesar de ρ poder estar entre 0 e 1, sabe-se que são incomuns na prática valores tais que $\rho < c$ ou $\rho > d$. Logo, pode-se assumir uma distribuição uniforme truncada para ρ , dada por:

$$p_2(\rho) = \frac{1}{d-c}, \quad c < \rho < d. \quad (3.21)$$

De forma análoga ao caso anteriormente apresentado, combinado-se as Equações (2.13), (3.11), e (3.21), obtém-se:

$$\begin{aligned}
p_2(\rho|x) &= \frac{L(\rho; x) \times p_2(\rho)}{\int_c^d L(\rho; x) \times p_2(\rho) d\rho} \\
&= \frac{\rho^y(1-\rho)^n \times \left(\frac{1}{d-c}\right)}{\int_c^d \rho^y(1-\rho)^n \times \left(\frac{1}{d-c}\right) d\rho},
\end{aligned}$$

para $c < \rho < d$.

Logo, após cancelar as constantes e rearranjar os termos, tem-se que:

$$p_2(\rho|x) = \begin{cases} \frac{\rho^{y+1-1}(1-\rho)^{n+1-1}}{B_{(c,d)}(y+1, n+1)}, & c < \rho < d, \\ 0, & \text{caso contrário,} \end{cases} \quad (3.22)$$

em que $B_{(c,d)}(y+1, n+1) = \int_c^d \rho^{(y+1)-1}(1-\rho)^{(n+1)-1} d\rho$ é a definição da função beta incompleta generalizada ([Wolfram Research Inc., 2015](#)), com parâmetros c , d , $y+1$ e $n+1$, o que resulta numa distribuição *a posteriori* beta generalizada, com estes mesmos parâmetros, $\rho|x \sim \text{beta}(c, d, y+1, n+1)$, a partir da qual as estimativas podem ser encontradas ([Choudhury & Borthakur, 2008](#)).

De fato, assumindo-se uma função de perda quadrática para o erro e tomando-se a média da função de densidade de probabilidade $\text{beta}(c, d, y+1, n+1)$, Equação (3.22), deduzem-se os seguintes estimadores pontuais bayesianos para ρ , L e L_q :

$$\hat{\rho}_2 = \frac{B_{(c,d)}(y+2, n+1)}{B_{(c,d)}(y+1, n+1)}, \quad (3.23)$$

$$\hat{L}_2 = \frac{B_{(c,d)}(y+2, n)}{B_{(c,d)}(y+1, n+1)}, \quad (3.24)$$

e

$$\hat{L}_{q^2} = \frac{B_{(c,d)}(y+3, n)}{B_{(c,d)}(y+1, n+1)}. \quad (3.25)$$

A distribuição preditiva *a posteriori* é dada por:

$$P_2(M = m|x) = \frac{B_{(c,d)}(y+1+m, n+2)}{B_{(c,d)}(y+1, n+1)}, \text{ para } m = 1, 2, \dots, \quad (3.26)$$

A implementação em R (R Core Team, 2013) dos estimadores bayesianos, para ρ , L e L_q , e a distribuição preditiva, para a *priori* uniforme, pode ser vista na Listagem 3.4. As avaliações da função beta incompleta, `Ibeta`, são feitas por meio do pacote `zipfR` (Evert & Baroni, 2006; 2007) do R.

***Priori* de Jeffreys**

Esta seção refere-se ao uso da especificação de distribuições *a priori* quando o interesse é que a informação dos dados seja dominante, no sentido de que a nossa informação *a priori* é vaga. Por outro lado, reconhece-se a necessidade de alguma forma de análise que, em algum sentido, consiga captar esta noção de uma *priori* que tenha um efeito mínimo, relativamente aos dados, na inferência final.

Tal análise pode ser pensada como um ponto de partida quando não se consegue fazer uma elicitacão detalhada do ‘verdadeiro’ conhecimento *a priori*. Assim, um outro tipo de informação *a priori* também pode ser utilizado para ρ , que é utilizar uma classe de *prioris* não informativas proposta por Jeffreys (ver Paulino et al., 2003), que é invariante a transformações 1 a 1. Embora em geral seja imprópria, em alguns casos a distribuição *a posteriori* é garantida ser própria. Baseada na informação de Fisher, que é dada conforme se segue:

$$I(\theta) = E \left[-\frac{\partial^2 \log p(X|\theta)}{\partial \theta^2} \right], \quad (3.27)$$

```

require(zipfR)
2 EPRo2<-function(amostra ,c,d) {
    n<-length(amostra)
4   y<-sum(amostra)
    return((Ibeta(c,y+2,n+1)-Ibeta(d,y+2,n+1))/
6         (Ibeta(c,y+1,n+1)-Ibeta(d,y+1,n+1)))
    }
8 EPL2<-function(amostra ,c,d) {
    n<-length(amostra)
10  y<-sum(amostra)
    return((Ibeta(c,y+2,n)-Ibeta(d,y+2,n))/
12        (Ibeta(c,y+1,n+1)-Ibeta(d,y+1,n+1)))
    }
14 EPLq2<-function(amostra ,c,d) {
    n<-length(amostra)
16  y<-sum(amostra)
    return((Ibeta(c,y+3,n)-Ibeta(d,y+3,n))/
18        (Ibeta(c,y+1,n+1)-Ibeta(d,y+1,n+1)))
    }
20 PP2<-function(amostra ,c,d,m) {
    n<-length(amostra)
22  y<-sum(amostra)
    return((Ibeta(c,y+1+m,n+2)-Ibeta(d,y+1+m,n+2))/
24        (Ibeta(c,y+1,n+1)-Ibeta(d,y+1,n+1)))
    }

```

Listagem 3.4: Implementação dos estimadores bayesianos e a preditiva para a *priori* uniforme

a seguinte distribuição *a priori* para o parâmetro θ pode então ser obtida:

$$p(\theta) \propto [I(\theta)]^{1/2}. \quad (3.28)$$

Com isso, a partir da Equação (3.6), a $p(\rho)$ pode ser encontrada por:

$$I(\rho) = E \left[-\frac{\partial^2 \log p(X|\rho)}{\partial \rho^2} \right]. \quad (3.29)$$

O logaritmo da distribuição $p(X|\rho)$, dada pela Equação (3.10), é dado por

$$\log p(X|\rho) = \log [\rho^X(1 - \rho)] = \log(1 - \rho) + X \log \rho. \quad (3.30)$$

A primeira e segunda derivadas são dadas por:

$$\frac{\partial \log p(X|\rho)}{\partial \rho} = \frac{1}{1 - \rho} + \frac{X}{\rho} \Rightarrow \frac{\partial^2 \log p(X|\rho)}{\partial \rho^2} = -\frac{1}{(1 - \rho)^2} - \frac{X}{\rho^2}. \quad (3.31)$$

Como $E[X|\rho] = \frac{\rho}{1 - \rho}$, temos que:

$$I(\rho) = \frac{1}{(1 - \rho)^2} + \frac{\frac{\rho}{1 - \rho}}{\rho^2} = \frac{1}{(1 - \rho)^2} + \frac{1}{(1 - \rho)\rho} = \rho^{-1}(1 - \rho)^{-2}. \quad (3.32)$$

Assim,

$$I(\rho) \propto [I(\theta)]^{1/2} = \rho^{-1/2}(1 - \rho)^{-1}. \quad (3.33)$$

Com isso, combinando-se *a priori* de Jeffreys, dada pela Equação (3.33), e a verossimilhança, Equação (3.11), é possível encontrar a seguinte distribuição *a posteriori* de ρ :

$$\begin{aligned} p_3(\rho|x) &= \frac{L(\rho; x) \times p_3(\rho)}{\int_0^1 L(\rho; x) \times p_3(\rho) d\rho} \\ &= \frac{\rho^y(1 - \rho)^n \times [\rho^{-1/2}(1 - \rho)^{-1}]}{\int_0^1 \rho^y(1 - \rho)^n \times [\rho^{-1/2}(1 - \rho)^{-1}] d\rho}, \end{aligned}$$

para $0 < \rho < 1$, com $y = \sum_{i=1}^n x_i$.

Assim, após rearranjar os termos, tem-se que:

$$p_3(\rho|x) = \begin{cases} \frac{\rho^{y+1/2-1}(1 - \rho)^{n-1}}{B(y + 1/2, n)}, & 0 < \rho < 1, \\ 0, & \text{caso contrário,} \end{cases} \quad (3.34)$$

em que $B(y + 1/2, n) = \int_0^1 \rho^{y+1/2-1}(1 - \rho)^{n-1}d\rho$ é a definição da função beta (Abramowitz & Stegun, 1972), com parâmetros $y + 1/2$ e n , o que resulta na distribuição *a posteriori* $\rho|x \sim \text{beta}(y+1/2, n)$. Note-se que, ao contrário dos casos anteriormente expostos, Equações (3.16) e (3.22), esta distribuição *a posteriori* é fixa. De fato, não é possível variar a forma da distribuição *a priori* de Jeffreys, que assume uma forma específica.

A partir desta distribuição *a posteriori*, estimativas bayesianas pontuais podem ser encontradas. Se uma função de perda quadrática é assumida, tomando-se então a média, deduzem-se os seguintes estimadores:

$$\hat{\rho}_3 = \frac{y + 1/2}{y + 1/2 + n}, \quad (3.35)$$

$$\hat{L}_3 = \frac{y + 1/2}{n}, \quad (3.36)$$

$$\hat{L}_{q3} = \frac{(y + 3/2)(y + 1/2)}{(n - 1)(y + 1/2 + n)}. \quad (3.37)$$

A distribuição preditiva *a posteriori* é dada por:

$$P_3(M = m|x) = \frac{B(y + 1/2 + m, n + 1)}{B(y + 1/2, n)}, \text{ para } m = 1, 2, \dots \quad (3.38)$$

A implementação em R (R Core Team, 2013) dos estimadores bayesianos, para ρ , L e L_q , e a distribuição preditiva, para a *priori* de Jeffreys, pode ser vista na Listagem 3.5.

Observações Finais

Como observação final, é importante ressaltar que as densidades preditivas desempenham um papel importante na seleção do modelo bayesiano e na sua comparação (Gamerman & Lopes, 2006). As chances *a posteriori* do modelo M_j relativo

```

1 EPRo3<-function(amostra) {
    n<-length(amostra)
3  y<-sum(amostra)
    return((1/2+y)/(y+1/2+n))
5 }
EPL3<-function(amostra) {
7  n<-length(amostra)
    y<-sum(amostra)
9  return((1/2+y)/n)
    }
11 EPLq3<-function(amostra) {
    n<-length(amostra)
13  y<-sum(amostra)
    return((y+3/2)*(y+1/2)/((n-1)*(y+1/2+n)))
15 }
PP3<-function(amostra ,m) {
17  n<-length(amostra)
    y<-sum(amostra)
19  return(beta(y+1/2+m, n+1)/beta(y+1/2, n))
    }

```

Listagem 3.5: Implementação dos estimadores bayesianos e a preditiva para a *priori* de Jeffreys

a um modelo M_k são definidas como $P(M_j = m|x)/P(M_k = m|x)$, as quais são a razão das chances *a priori*, $P(M_j)/P(M_k)$, do modelo M_j relativo ao modelo M_k , chamado de fator de Bayes, dado por:

$$B_{jk} = \frac{f(M|M_j)}{f(M|M_k)} \Rightarrow B_{12} = \frac{P_1(M = m|x)}{P_2(M = m|x)}, \quad m = 0, 1, 2, \dots \quad (3.39)$$

Para decidir entre os modelos M_j e M_k , é recomendada a utilização da seguinte regra (Jeffreys, 1998): considera-se que há provas (i) decisivas, (ii) fortes ou (iii) substanciais contra M_k quando B_{jk} é (i) superior a 100, (ii) está entre 10 e 100 ou (iii) está entre 3 e 10, respectivamente.

3.2 Filas $M/M/1/K$

3.2.1 Introdução

Quando se tem servidores simples em um espaço limitado para a espera, chamamos de filas $M/M/1/K$, em que K representa o limite de clientes que poderão ficar simultaneamente no sistema de filas. Quando K é muito grande ($\rightarrow \infty$), tem-se então uma fila $M/M/1$. A solução da Equação (2.5), para $n = 0, 1, \dots, K$, é (Wagner, 1986):

$$p_n = \begin{cases} \frac{\rho^n(1-\rho)}{1-\rho^{K+1}}, & \text{para } \rho \neq 1 \\ \frac{1}{K+1}, & \text{para } \rho = 1. \end{cases} \quad (3.40)$$

Percebe-se que quando $\rho < 1$ e $K \rightarrow \infty$, tem-se que a Equação (3.40) coincide com a Equação (3.6). O número esperado de clientes $E[N]$ e o tempo médio no sistema $E[W]$, para $\rho \neq 1$, são dados respectivamente por:

$$E[N] = \frac{\rho}{1-\rho} - \frac{\rho^{K+1}(K+1)}{1-\rho^{K+1}} \quad (3.41)$$

$$E[W] = \frac{1}{\mu(1-\rho)} - \frac{K\rho^K}{\mu(1-\rho^K)} \quad (3.42)$$

3.2.2 Inferência Clássica

Alguns trabalhos têm sido desenvolvidos para filas markovianas finitas de servidor único, modelo $M/M/1/K$. Dentre eles pode-se citar Kuumola et al. (2002), que estudam o tamanho médio da fila, L_q , por meio de um sistema de equações ordinárias diferenciais e métodos numéricos para aproximar a distribuição estacionária.

Uma metodologia semelhante àquela proposta na Seção 3.1 foi desenvolvida para filas $M/M/1/K$. Baseado no processo markoviano com um servidor simples e

considerando a distribuição de probabilidade estacionária apresentada na Equação (3.40), considere agora que $x = (x_1, x_2, \dots, x_n)^T$ seja uma amostra aleatória que representa o número de clientes que ficaram no sistema em n partidas de usuários. Neste caso apenas é permitido que no máximo K clientes podem permanecer no sistema. Assim, a função de verossimilhança pode ser dada como

$$\begin{aligned} L(\rho; x) &= \frac{\rho^{x_1}(1-\rho)}{1-\rho^{K+1}} \times \dots \times \frac{\rho^{x_n}(1-\rho)}{1-\rho^{K+1}} \\ &= \frac{\rho^{\sum_{i=1}^n x_i} \times (1-\rho)^n}{(1-\rho^{K+1})^n}. \end{aligned} \quad (3.43)$$

Após a obtenção de uma estimativa para ρ , medidas de desempenho de interesse podem ser estimadas, tais como L e L_q , dadas por (Gross et al., 2009):

$$\hat{L}_q = \frac{\hat{\rho}}{1-\hat{\rho}} - \frac{\hat{\rho}(K\hat{\rho}^K + 1)}{1-\hat{\rho}^{K+1}}, \quad (3.44)$$

$$\hat{L} = \hat{L}_q + \left(1 - \frac{1-\hat{\rho}}{1-\hat{\rho}^{K+1}}\right), \quad (3.45)$$

em que $\hat{\rho}$ é uma estimativa para ρ .

Na inferência clássica, temos o método dos momentos e da máxima verossimilhança, dentre os mais populares métodos de estimação pontual. Nesta tese utilizaremos este que, desenvolvido por Fisher, na década de 20, é considerado um dos melhores. Conforme pode ser inferido pelo nome do método, o estimador será aquele valor do parâmetro que maximize a função de verossimilhança, Equação (3.43).

Poderíamos tentar obter o máximo da Equação (3.43) utilizando-nos dos recursos do cálculo diferencial, isto é, possivelmente tirando o logaritmo da verossimilhança, para transformar os produtos em somas, para buscar o ponto $\hat{\rho}$ que anula a derivada $d \log L(\rho; x)/d\rho$ e determinar o vício e variabilidade do EMV de forma

analítica. Entretanto, por conveniência optaremos pela maximização numérica e simulação Monte Carlo.

A implementação do EMV utilizada nesta tese, em R (R Core Team, 2013), pode ser visualizada na Listagem 3.6. Note-se que, por conveniência, foi considerado o logaritmo da função de verossimilhança, com a facilidade de transformar produtos em somas. A maximização da log-verossimilhança é feita numericamente, por meio de função interna do R. Testes (não apresentados) foram realizados com a função de verossimilhança original e indicaram que os resultados não mudaram significativamente, em termos de precisão ou de esforço computacional.

```

EMVRoMMIK<-function(K, amostra) {
2  loglike.f<-function(rho,K,n,sumxi) {
      sumxi*log(rho) + n*log(1-rho) - n*log(1-rho^(K+1))
4  }
  Eps.EMV<-1e-06
6  n<-length(amostra)
  sumxi<-sum(amostra)
8  res<-optimize(loglike.f,c(Eps.EMV,1-Eps.EMV),K,n,sumxi,maximum=TRUE,tol=Eps.EMV)
  return(res$maximum)
10 }

```

Listagem 3.6: Implementação do EMV para ρ

3.2.3 Correção *Bootstrap*

Foi desenvolvida uma versão do EMV com correção via *bootstrap*, conforme descrito na Seção 2.4. O código, em R (R Core Team, 2013), utilizado nesta tese é mostrado na Listagem 3.7. O parâmetro K é o número máximo de usuários simultaneamente (em serviço e na fila de espera) na fila markoviana finita, $M/M/1/K$. Note-se que foram utilizadas 1.000 replicações *bootstrap* e que a correção dá-se sobre o EMV (ver Listagem 3.6).

```

EBoRoMM1K<-function(K, amostra) {
2   B<-1000
   soma<-0
4   for (i in 1:B) {
       reamostra<-sample(amostra, replace=T)
6       estm<-EMVRoMM1K(K, reamostra)
       soma<-soma+estm}
8   estmStar=soma/B
   return(2*EMVRoMM1K(K, amostra)-estmStar)
10 }

```

Listagem 3.7: Implementação do EMV corrigido via *bootstrap*

3.2.4 Inferência Bayesiana

Neste caso, o processo inferencial para estimar ρ a partir da Equação (3.43) será o mesmo adotado para filas $M/M/1$, assumindo uma distribuição *a priori* beta, com parâmetros a e b , Equação (3.15), ou seja, $p(\rho) \sim \text{beta}(a, b)$. Assim, a distribuição *a posteriori* para ρ é dada por:

$$\begin{aligned}
p(\rho|x) &\propto L(\rho; x) \times p(\rho) \\
&\propto \frac{\rho^{\sum_{i=1}^n x_i} (1-\rho)^n}{(1-\rho^{K+1})^n} \times \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \rho^{(a-1)} (1-\rho)^{(b-1)} \\
&\propto \frac{\rho^{\sum_{i=1}^n x_i + a - 1} (1-\rho)^{n+b-1}}{(1-\rho^{K+1})^n}.
\end{aligned} \tag{3.46}$$

Ao contrário do caso das filas $M/M/1$, Seção 3.1.4, a Eq. (3.46) não é uma distribuição padrão. Portanto, a obtenção exata de medidas de posição e de dispersão, ou mesmo a obtenção de uma amostra, pode se tornar uma tarefa bastante árdua. Para resolver o problema, justifica-se então a utilização de métodos Monte Carlo via cadeias de Markov (MCMC), que têm sido fundamentais para o desenvolvimento e difusão da inferência bayesiana. Dentre os métodos MCMC, encontram-se os algoritmos de [Metropolis et al. \(1953\)](#) e [Hastings \(1970\)](#) e seus

derivados. Em particular, nesta tese será utilizado o método *adaptive rejection Metropolis sampling* (ARMS, de [Gilks et al., 1995](#)), que torna possível a iteração da cadeia mesmo sob distribuições não-padrão, como é o caso aqui.

Após encontrar uma amostra da distribuição *a posteriori*, faz-se necessário encontrar a distribuição preditiva, que é uma abordagem centrada nas predições. Ou seja, as quantidades desconhecidas de interesse não são os parâmetros, mas sim as variáveis aleatórias futuras ([Paulino et al., 2003](#)). Assim, a definição de densidade preditiva é fundamental para a inferência preditiva. Desta forma, Berger (1985) comenta uma típica situação que envolve a predição de uma variável aleatória, com densidade $g(z|\theta)$, com θ desconhecido, em que existem dados disponíveis, x , com densidade $p(x|\theta)$. No nosso caso, x representa dados de clientes no sistema, e deseja-se prever a futura variável resposta. Assim, pode ser definida como a distribuição conjunta de z e θ dado x , e integrando com relação a θ teremos a distribuição de z dado x :

$$p(z|x) = \int g(z|\theta)p(\theta|x)d\theta = \int g(z|\theta)p(\theta)p(x|\theta)d\theta, \quad (3.47)$$

em que $p(z|x)$ é a densidade preditiva da variável aleatória Z , $g(z|\theta)$ é a densidade de z dado θ , e $p(\theta|x)$ é a densidade *a posteriori* de θ .

Após obter uma amostra da distribuição *a posteriori*, as medidas de desempenho L_q e L podem ser estimadas, em que $\hat{\rho}$ é a média obtida a partir de $p(\rho|x)$.

A implementação do estimador bayesiano utilizada nesta tese, no ambiente R ([R Core Team, 2013](#)), pode ser visualizada na Listagem 3.8. A extração da amostra da distribuição *a posteriori* é feita por meio da função *arms* (utilizado o tamanho 5.000), disponível no pacote HI ([Petris & Tardella, 2013](#)) do R. A distribuição *a posteriori* é representada por meio do logaritmo da sua função densidade (a menos da constante de normalização). Como foi considerada a função de perda quadrática, o estimador pontual de Bayes é a média aritmética simples da amostra.

```

require(HI)
2 EBaRoMMIK<-function(K, amostra , a , b) {
  logpost . f<-function(rho ,K, n, sumxi , a , b) {
4     (sumxi+a-1)*log(rho) + (n+b-1)*log(1-rho) - n*log(1-rho^(K+1))
  }
6   sSize<-5000
  n<-length(amostra)
8   sumxi<-sum(amostra)
  res<-arms(runif(1) , logpost . f , function(x,K, n, sumxi , a , b)((x>0)*(x<1)) , sSize ,K, n,
    sumxi , a , b)
10  return(mean(res))
}

```

Listagem 3.8: Implementação do estimador bayesiano

3.2.5 Simulação

Em filas $M/M/1/K$, ao contrário do modelo de filas $M/M/1$ que segue uma distribuição conhecida (geométrica), é necessário utilizar algum método de geração de variáveis aleatórias para distribuição discreta geral. Usaremos um análogo discreto do método da transformação inversa, em que, basicamente, é necessário gerar números $R \sim \text{unif}(0, 1)$, isto é, de uma distribuição uniforme entre 0 e 1, e conhecer as probabilidades de interesse, $P\{N = n_j\} = p_j, \forall j$. Assim, para simular uma variável aleatória discreta N com função de probabilidade

$$P\{N = n_j\} = p_j, j = 0, 1, \dots, \sum_j p_j = 1,$$

deve-se fazer:

$$N = \begin{cases} n_0, & \text{se } R \leq P_0, \\ n_1, & \text{se } P_0 < R \leq P_0 + P_1, \\ \vdots & \\ n_j, & \text{se } \sum_{i=0}^{j-1} P_i < R \leq \sum_{i=0}^j P_i, \\ \vdots & \end{cases}$$

Como

$$P\{N = n_j\} = P\left\{\sum_{i=0}^{j-1} p_i < R \leq \sum_{i=0}^j p_i\right\} = p_j,$$

então N segue a distribuição desejada.

No caso do modelo de filas markovianas finitas, $M/M/1/K$, tem-se da Eq. (3.40) que $p_n \equiv P\{N = n\} = \frac{\rho^n(1-\rho)}{1-\rho^{K+1}}$, $n = 0, 1, \dots, K$, para $\rho \neq 1$. Segue então que:

$$\sum_{i=0}^n p_i = \frac{1 - \rho^{n+1}}{1 - \rho^{K+1}}, \quad n = 0, 1, \dots, K.$$

Fazendo $c = 1 - \rho^{K+1}$, segue que:

$$\begin{aligned} \sum_{i=0}^{N-1} p_i < R &\leq \sum_{i=0}^N p_i \Rightarrow \\ \frac{1 - \rho^N}{c} < R &\leq \frac{1 - \rho^{N+1}}{c} \Rightarrow \\ 1 - \rho^N < Rc &\leq 1 - \rho^{N+1} \Rightarrow \\ -1 + \rho^{N+1} \leq -Rc &< -1 + \rho^N \Rightarrow \\ \rho^{N+1} \leq 1 - Rc &< \rho^N \Rightarrow \\ (N+1)\log(\rho) \leq \log(1 - Rc) &< N\log(\rho) \Rightarrow \\ N < \frac{\log(1 - Rc)}{\log(\rho)} &\leq N+1 \Rightarrow \\ \frac{\log(1 - Rc)}{\log(\rho)} - 1 \leq N &< \frac{\log(1 - Rc)}{\log(\rho)}. \end{aligned}$$

Portanto,

$$N = \left\lceil \frac{\log(1 - Rc)}{\log(\rho)} - 1 \right\rceil, \quad (3.48)$$

em que $\lceil x \rceil$ é o menor inteiro não inferior a x .

A Listagem 3.9 apresenta a implementação utilizada nesta tese, codificada em R (R Core Team, 2013), e um exemplo de chamada.

```
1 rmm1k<-function(tam, rho, K){ # simulate
  u<-runif(tam, 0, 1)
3  x<-log(1-u*(1-rho^(K+1)))/log(rho)
  return(x)
5 }

7 > set.seed(13579)
  > rmm1k(tam=10, rho=0.20, K=5)
9 [1] 0 1 0 1 0 0 0 2 1 0
```

Listagem 3.9: Implementação da geração de amostras de uma fila $M/M/1/K$

3.3 Observações Finais

Neste seção foram apresentadas as metodologias inferenciais clássica e bayesiana, para modelos de filas markovianas de servidor único infinitas e finitas, respectivamente, para os modelos $M/M/1$ e $M/M/1/K$, bem como um estimador corrigido via *bootstrap* baseado no estimador de máxima verossimilhança para ρ . No próximo capítulo serão apresentados e discutidos os principais resultados computacionais que foram obtidos.

Capítulo 4

Resultados Computacionais e Discussão

Neste capítulo serão apresentados os resultados computacionais obtidos, a partir de códigos escritos em R (R Core Team, 2013), disponíveis nos Apêndices A e B, ou diretamente com o autor.

4.1 Filas $M/M/1$

A implementação para a geração de amostras para o número de clientes no sistema, em R (R Core Team, 2013), pode ser visualizada na Listagem 4.1.

```
1 gerAmosRo<-function(tam, rho) { # simulate
  return(rgeom(tam, 1-rho))
3 }
```

Listagem 4.1: Geração das amostras

Na Listagem 4.2 é apresentada a implementação em R (R Core Team, 2013) de uma função para a análise de desempenho dos estimadores via simulação Monte Carlo, que determina média e erro padrão da média (EPM), para 10.000 repli-

cações¹. A função `fEst(K, amostra, ...)` pode ser qualquer uma daqueles estimadores, para ρ , L ou L_q (EMV, EMV corrigido via *bootstrap* e bayesianos). O estado da semente aleatória `.GlobalEnv$.Random.seed` é armazenado imediatamente antes da chamada à função `fEst(K, amostra, ...)`, que pode ser estocástica (p.e., nos casos *bootstrap* e bayesianos), para garantir que as amostras geradas para estimação serão as mesmas para todos os estimadores.

```

1 MtCa<-function(tam, rho, fEst, ...) {
  rep<-10000
3  amostra<-numeric(tam)
  est<-numeric(rep)
5  for (i in 1:rep) {
    # gerar dados
7    amostra<-gerAmosRo(tam, rho)
    oldseed<-GlobalEnv$.Random.seed
9    # ajustar modelo e guardar ajuste
    est[i]<-fEst(amostra, ...)
11   GlobalEnv$.Random.seed<-oldseed
  return(c(mean(est), sqrt(var(est)/rep)))
13 }

```

Listagem 4.2: Implementação do Monte Carlo para avaliação dos estimadores

4.1.1 Inferência Clássica

Todos os cálculos foram realizados com as implementações apresentadas no Capítulo 3, desenvolvidas para o ambiente R (R Core Team, 2013). Foram realizadas simulações de amostras com tamanhos $n \in \{10; 20; 50; 100; 200\}$ e intensidades de tráfego $\rho \in \{0, 01; 0, 10; 0, 20; 0, 50; 0, 90; 0, 99\}$. Foram calculadas estimativas médias de ρ pelo EMV, Equação (3.12), e pela sua versão corrigida via *bootstrap*, Equação (2.12), com os respectivos erros-padrão da média (EPM), com 10.000 re-

¹Observe-se que é também bastante comum em estudos de simulação a apresentação da raiz do erro quadrático médio (ou o RMSE, do inglês ‘*root mean square error*’), definido como

$$\text{RMSE}(\hat{\theta}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{\theta}_i - \theta)^2} = \sqrt{\text{Vício}(\hat{\theta})^2 + \text{Var}(\hat{\theta})}, \text{ que agrega a informação de média e erro.}$$

plicações Monte Carlo, para todos os cenários. Os resultados podem ser vistos na Tabela 4.1, que se encontram sintetizados na Figura 4.2.

Podemos confirmar o vício do EMV para ρ , para amostras pequenas e moderadas e para intensidades de tráfego $\rho \approx 0,50$. Este comportamento é conhecido, uma vez que a distribuição envolvida é a geométrica clássica. Para valores extremos (i.e., $\rho \approx 0,99$ e $\rho \approx 0,01$), as estimativas são bastante satisfatórias, conforme visto nas Figuras 4.2-(a) e -(b), com pequenos erros de estimação e baixo erro-padrão da média, o que sugere que estes são os valores mais fáceis de estimar na prática. Também para amostras grandes ($n \geq 100$), o desempenho é melhor, conforme visto nas Figuras 4.2-(c) e -(d), que mostra pequenos valores para os erros de estimação e erro-padrão da média. Portanto, as medidas de desempenho da fila que são estimadas a partir de ρ também deverão apresentar vício, o que deve ser uma preocupação para o pesquisador. Sobre o estimador corrigido via *bootstrap* é suficiente dizer que ele apresenta sempre menores erros médios de estimação, sem inflação da variância, conforme constatado pela semelhança entre os erros-padrão da média dos dois estimadores, EMV e *bootstrap*, Figuras 4.2-(b) e -(d).

Também foram realizados experimentos computacionais para os estimadores de L , Equação (3.13), e L_q , Equação (3.14), bem como com suas versões corrigidas via *bootstrap*. São apresentados nas Tabelas 4.3 e 4.5 e nas Figuras 4.4 e 4.6 os resultados obtidos para L e $L_q \in \{0,5; 1; 2; 5; 10; 20\}$, para tamanhos de amostras $n \in \{10; 20; 50; 100; 200\}$ e 10.000 replicações Monte Carlo. É possível notar pelos resultados que, com o esforço computacional extra do método *bootstrap*, o pesquisador pode alcançar com amostras de tamanho $n = 10$ estimativas para o L e L_q com o mesmo erro médio que obteria com o EMV, para amostras de tamanho $n = 200$. Esta redução pode ser bastante importante do ponto de vista prático, pela diminuição que pode acarretar nos tempos para obtenção das amostras e nos custos de estimação.

Figura 4.1: Estimativas clássicas médias de ρ e EPM (entre parênteses)

Estimador	ρ	n				
		10	20	50	100	200
EMV	0,01	0,00910 (0,00029)	0,00968 (0,00021)	0,00987 (0,00014)	0,00994 (0,00010)	0,01000 (0,00007)
	0,10	0,09279 (0,00084)	0,09668 (0,00062)	0,09820 (0,00039)	0,09940 (0,00028)	0,09953 (0,00020)
	0,20	0,18718 (0,00109)	0,19372 (0,00079)	0,19710 (0,00050)	0,19892 (0,00035)	0,19923 (0,00025)
	0,50	0,47665 (0,00116)	0,48838 (0,00081)	0,49509 (0,00050)	0,49801 (0,00035)	0,49868 (0,00025)
	0,90	0,88979 (0,00036)	0,89500 (0,00023)	0,89801 (0,00014)	0,89895 (0,00010)	0,89940 (0,00007)
	0,99	0,98887 (0,00004)	0,98945 (0,00002)	0,98978 (0,00001)	0,98990 (0,00001)	0,98994 (0,00001)
<i>Bootstrap</i>	0,01	0,00992 (0,00031)	0,00970 (0,00022)	0,00993 (0,00014)	0,01011 (0,00010)	0,01005 (0,00007)
	0,10	0,09749 (0,00089)	0,09828 (0,00063)	0,09931 (0,00040)	0,09991 (0,00029)	0,09972 (0,00020)
	0,20	0,19527 (0,00114)	0,19688 (0,00081)	0,19912 (0,00051)	0,19956 (0,00036)	0,19981 (0,00026)
	0,50	0,49470 (0,00119)	0,49725 (0,00081)	0,49900 (0,00051)	0,49951 (0,00035)	0,49995 (0,00025)
	0,90	0,89909 (0,00034)	0,89959 (0,00022)	0,89994 (0,00014)	0,90007 (0,00010)	0,90003 (0,00007)
	0,99	0,98997 (0,00004)	0,99000 (0,00002)	0,99001 (0,00001)	0,99000 (0,00001)	0,99001 (0,00001)

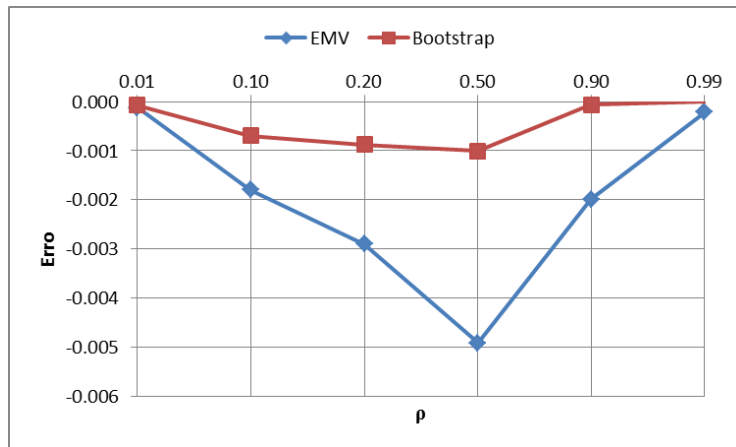
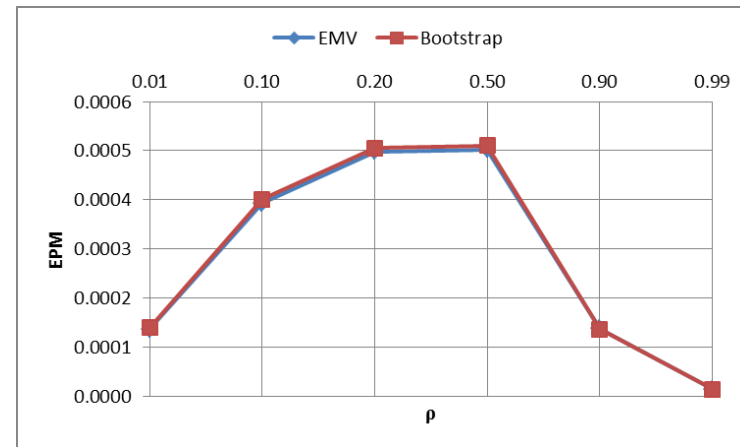
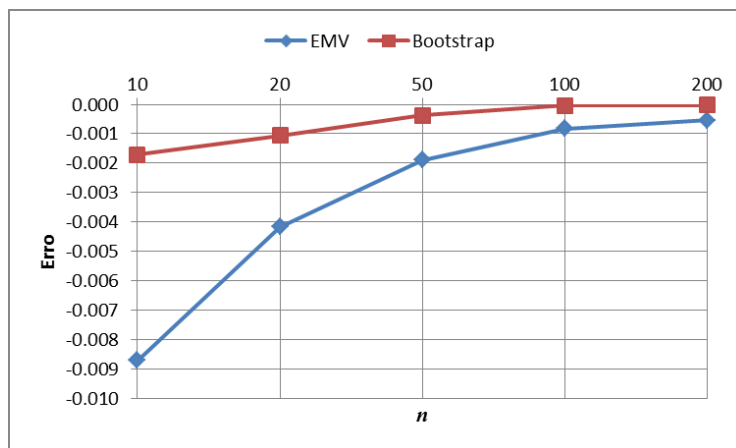
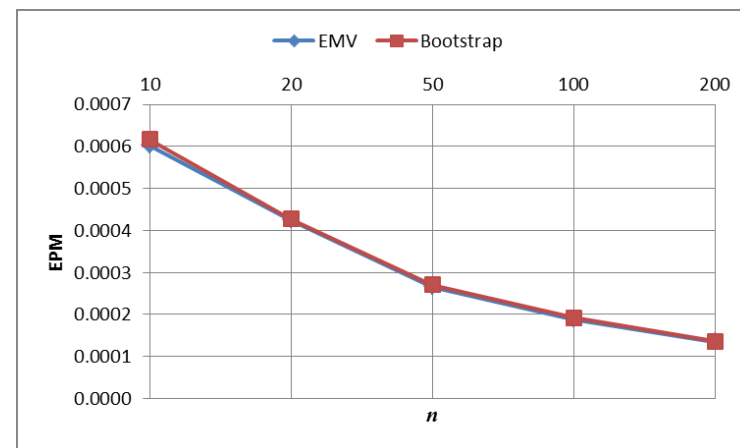
(a) erro médio em função de ρ (b) EPM médio em função de ρ (c) erro médio em função de n (d) EPM médio em função de n Figura 4.2: Desempenho dos estimadores para ρ

Figura 4.3: Estimativas clássicas médias de L e EPM (entre parênteses)

Estimador	L	n				
		10	20	50	100	200
EMV	0,5	0,45492 (0,00121)	0,47629 (0,00087)	0,48913 (0,00054)	0,49547 (0,00038)	0,49696 (0,00027)
	1	0,91076 (0,00116)	0,95457 (0,00081)	0,98055 (0,00050)	0,99207 (0,00035)	0,99473 (0,00025)
	2	1,81358 (0,00094)	1,90545 (0,00064)	1,95981 (0,00039)	1,98233 (0,00027)	1,98902 (0,00019)
	5	4,52533 (0,00056)	4,76186 (0,00036)	4,89951 (0,00022)	4,94713 (0,00015)	4,97105 (0,00011)
	10	9,00473 (0,00033)	9,49177 (0,00021)	9,78876 (0,00013)	9,89238 (0,00009)	9,93663 (0,00006)
	20	18,07304 (0,00018)	19,01846 (0,00011)	19,60429 (0,00007)	19,78629 (0,00005)	19,89911 (0,00003)
<i>Bootstrap</i>	0,5	0,48653 (0,00126)	0,49397 (0,00087)	0,49726 (0,00055)	0,49942 (0,00039)	0,49932 (0,00027)
	1	0,97901 (0,00119)	0,98906 (0,00081)	0,99599 (0,00051)	0,99805 (0,00035)	0,99980 (0,00025)
	2	1,98602 (0,00094)	1,98738 (0,00064)	1,99717 (0,00039)	1,99813 (0,00028)	2,00185 (0,00020)
	5	4,93839 (0,00055)	4,99765 (0,00036)	4,99523 (0,00022)	4,99988 (0,00015)	5,00232 (0,00011)
	10	9,98817 (0,00031)	9,98713 (0,00020)	9,99158 (0,00013)	9,99431 (0,00009)	10,00380 (0,00006)
	20	19,89938 (0,00016)	20,11764 (0,00011)	19,96857 (0,00007)	20,00419 (0,00005)	20,01488 (0,00003)

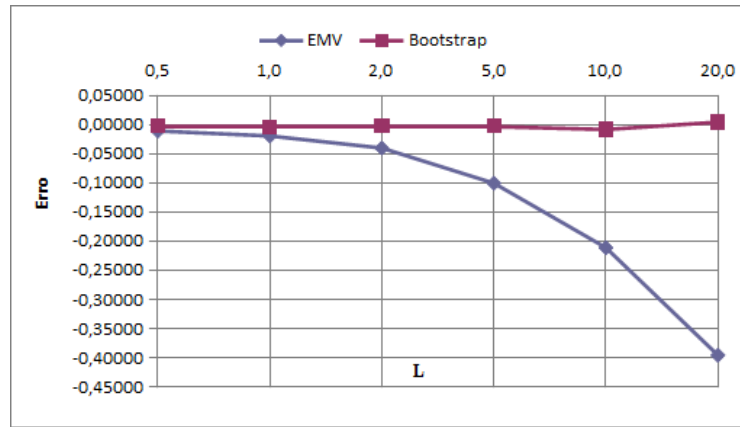
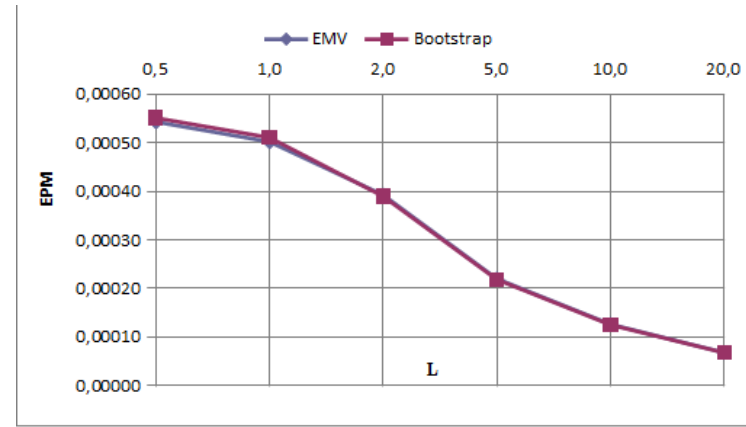
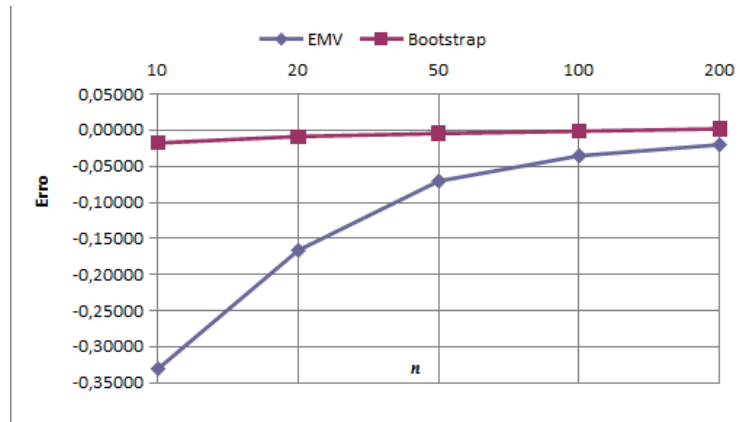
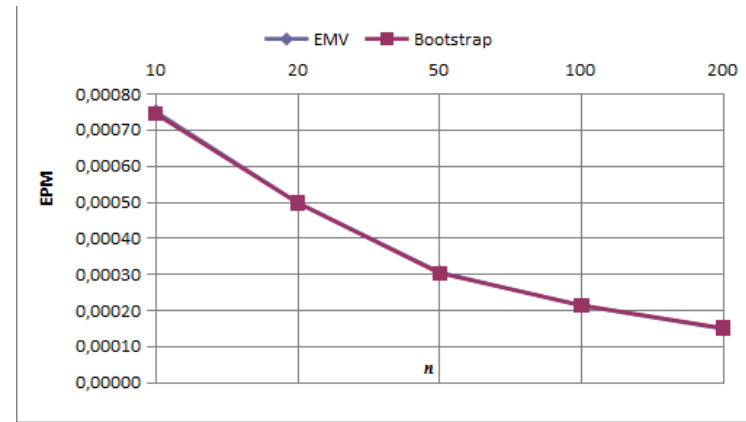
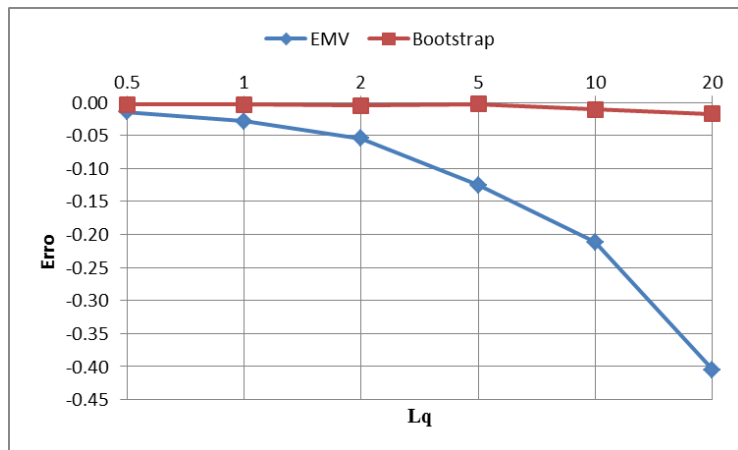
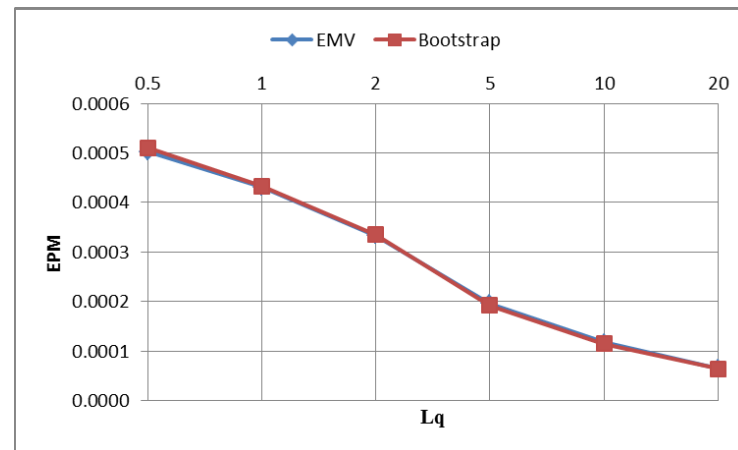
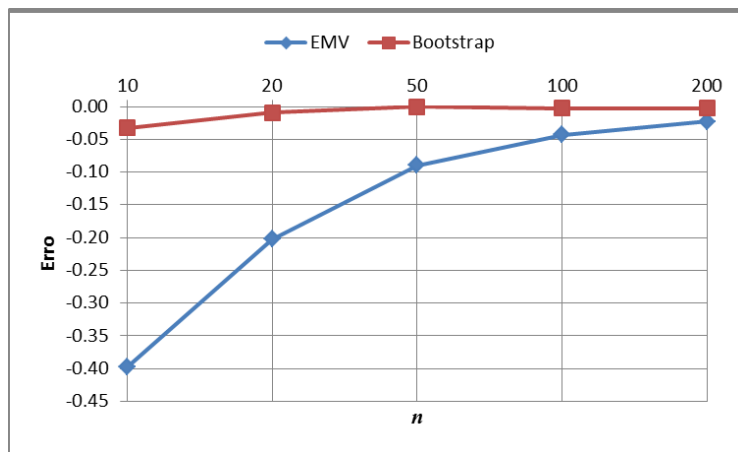
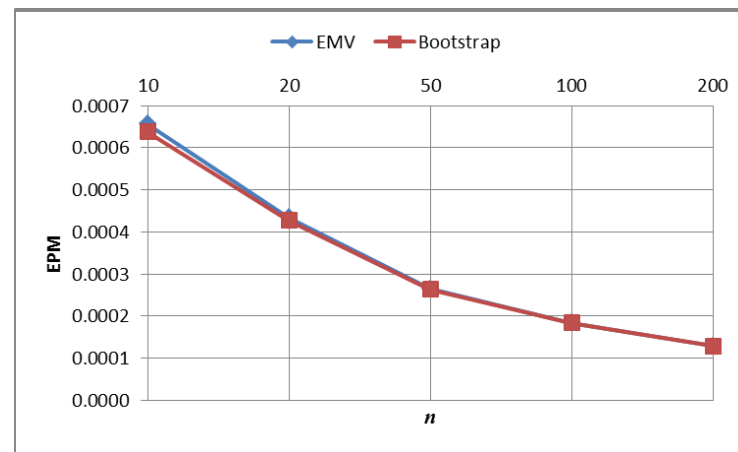
(a) erro médio em função de L (b) EPM médio em função de L (c) erro médio em função de n (d) EPM médio em função de n Figura 4.4: Desempenho dos estimadores para L

Figura 4.5: Estimativas clássicas médias de L_q e EPM (entre parênteses)

Estimador	L_q	n				
		10	20	50	100	200
EMV	0,5	0,43411 (0,00116)	0,46619 (0,00081)	0,48546 (0,00050)	0,49406 (0,00035)	0,49605 (0,00025)
	1	0,87197 (0,00102)	0,93441 (0,00070)	0,97168 (0,00043)	0,98821 (0,00030)	0,99246 (0,00021)
	2	1,75143 (0,00082)	1,87568 (0,00055)	1,94550 (0,00033)	1,97572 (0,00023)	1,98579 (0,00016)
	5	4,45302 (0,00049)	4,71935 (0,00032)	4,87472 (0,00020)	4,93802 (0,00014)	4,96822 (0,00009)
	10	8,92303 (0,00031)	9,45755 (0,00019)	9,78838 (0,00012)	9,88343 (0,00008)	9,93151 (0,00006)
	20	17,96563 (0,00017)	18,96589 (0,00011)	19,59511 (0,00006)	19,78940 (0,00005)	19,90578 (0,00003)
<i>Bootstrap</i>	0,5	0,48431 (0,00119)	0,49181 (0,00081)	0,49700 (0,00051)	0,49854 (0,00035)	0,49985 (0,00025)
	1	0,97472 (0,00103)	0,98885 (0,00070)	0,99747 (0,00043)	0,99701 (0,00030)	1,00073 (0,00021)
	2	1,96776 (0,00080)	1,99504 (0,00054)	1,99985 (0,00033)	2,00386 (0,00023)	1,99483 (0,00016)
	5	4,96641 (0,00048)	4,97319 (0,00032)	4,99995 (0,00019)	4,99733 (0,00014)	5,00092 (0,00010)
	10	9,93206 (0,00029)	9,98385 (0,00019)	10,02029 (0,00011)	9,99415 (0,00008)	9,98929 (0,00006)
	20	19,84954 (0,00016)	20,03573 (0,00010)	20,00104 (0,00006)	19,97931 (0,00005)	19,98261 (0,00003)

(a) erro médio em função de L_q (b) EPM médio em função de L_q (c) erro médio em função de n (d) EPM médio em função de n Figura 4.6: Desempenho dos estimadores para L_q

4.1.2 Inferência Bayesiana

Para a metodologia bayesiana proposta, também foram geradas amostras do número de clientes no sistema no momento da partida, a partir da Equação (3.6), utilizando amostras de tamanho $n \in \{10; 20; 50; 100; 200\}$ para $\rho \in \{0, 2; 0, 5; 0, 9\}$. O procedimento foi realizado 10.000 vezes. Quatro tipos de distribuições *a priori* foram utilizadas, duas distribuições *a priori* beta, $\rho \sim \text{beta}(1, 5; 2, 5)$ e $\rho \sim \text{beta}(2, 5; 1, 5)$, e uma distribuição uniforme truncada, $\rho \sim \text{unif}(0, 05; 0, 95)$, conforme mostrado na Figura 4.7, e a distribuição *a priori* de Jeffreys.

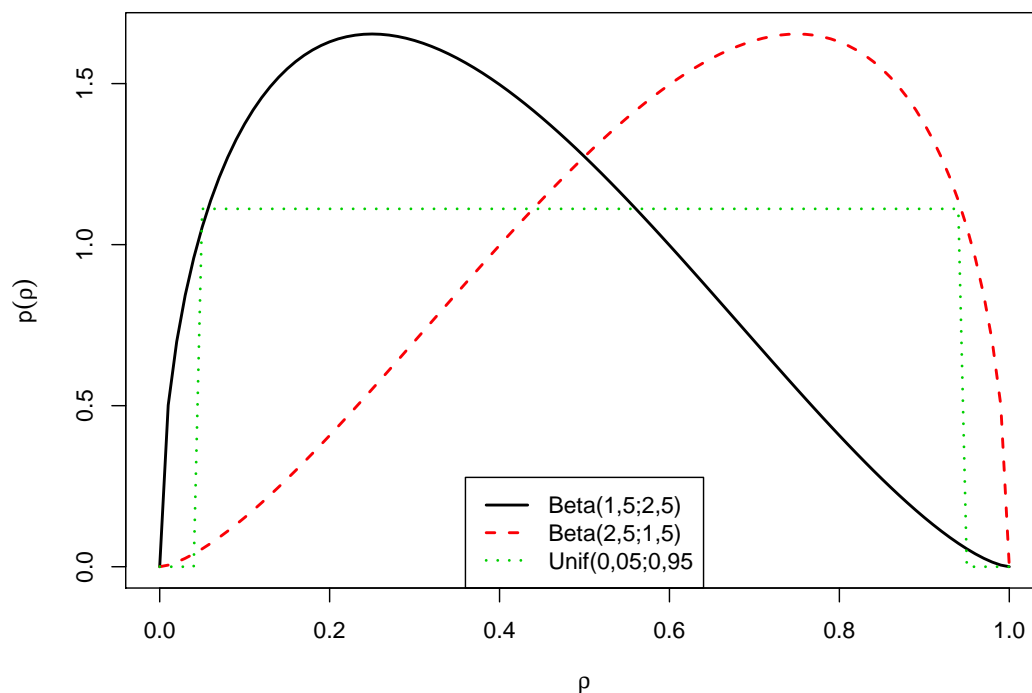


Figura 4.7: Distribuições *a priori* beta e uniforme truncada para ρ

A Tabela 4.1 mostra as estimativas para cada tamanho de amostra e valor de ρ . Observa-se que com o aumento do tamanho amostral, as estimativas se

aproximam do verdadeiro valor de ρ utilizado, sendo que as estimativas utilizando a distribuição *a priori* de Jeffreys parecem estar mais próximas para ρ de 0,2 e 0,5. Por outro lado, quando o valor de ρ é 0,5, os resultados são melhores quando utilizamos a distribuição $\rho \sim \text{beta}(2, 5; 1, 5)$. Outro fato importante é que quando utilizamos distribuições *a priori* beta próximas do verdadeiro valor, as estimativas para ρ são melhores. Com isso, podemos dizer que, para o gestor, quanto melhor ele poder quantificar a sua informação inicial sobre ρ , os resultados tendem a ser mais precisos.

A Tabela 4.2 mostra algumas características de operação da fila. Observa-se que com o aumento do tamanho amostral as estimativas se aproximam dos valores exatos para o número de clientes no sistema e o tamanho da fila. As estimativas baseadas na distribuição *a priori* de Jeffreys parece ter um desempenho um pouco melhor. Quando o ρ é 0,9, as estimativas para o número de clientes no sistema e na fila são melhores ao utilizar a distribuição *a priori* uniforme, para um tamanho amostral maior que 100.

Para determinar qual modelo melhor se ajustou aos dados (simulados), fixou-se um tamanho amostral de 200 e obteve-se a distribuição preditiva do número de clientes no sistema no momento de partida, para o número de clientes de 0 a 5, e calculou-se o fator de Bayes para cada um deles. A partir da Tabela 4.3 observa-se que o fator de Bayes ficou próximo de 1 para todos os valores de ρ , não evidenciando para este parâmetro que algum dos modelos utilizados seja superior ao outro. Utilizando *a priori* de Jeffreys, o fator de Bayes, em comparação com as outras distribuições utilizada, também apresentou um valor próximo de 1.

Tabela 4.1: Estimativas bayesianas médias de ρ e EPM (entre parênteses)

ρ	Distribuição	n				
		10	20	50	100	200
0,2	Beta(1,5;2,5)	0,2358 (0,0008)	0,2198 (0,0007)	0,2080 (0,0005)	0,2044 (0,0003)	0,2020 (0,0002)
	Beta(2,5;1,5)	0,2970 (0,0007)	0,2544 (0,0006)	0,2231 (0,0005)	0,2122 (0,0003)	0,2060 (0,0002)
	Uniforme	0,2420 (0,0008)	0,2205 (0,0007)	0,2069 (0,0005)	0,2037 (0,0003)	0,2016 (0,0002)
	Jeffreys	0,2195 (0,0010)	0,2098 (0,0007)	0,2035 (0,0005)	0,2021 (0,0003)	0,2008 (0,0002)
0,5	Beta(1,5;2,5)	0,4628 (0,0010)	0,4790 (0,0007)	0,4906 (0,0005)	0,4956 (0,0003)	0,4975 (0,0002)
	Beta(2,5;1,5)	0,5057 (0,0009)	0,5021 (0,0007)	0,5003 (0,0005)	0,5006 (0,0003)	0,4999 (0,0002)
	Uniforme	0,4813 (0,0008)	0,4895 (0,0007)	0,4953 (0,0004)	0,4981 (0,0003)	0,4987 (0,0002)
	Jeffreys	0,4906 (0,0011)	0,4950 (0,0008)	0,4976 (0,0005)	0,4993 (0,0003)	0,4993 (0,0002)
0,9	Beta(1,5;2,5)	0,8686 (0,0004)	0,8844 (0,0002)	0,8938 (0,0001)	0,8968 (0,0001)	0,8983 (0,0001)
	Beta(2,5;1,5)	0,8791 (0,0004)	0,8896 (0,0002)	0,8958 (0,0001)	0,8978 (0,0001)	0,8988 (0,0001)
	Uniforme	0,8803 (0,0004)	0,8907 (0,0002)	0,8964 (0,0001)	0,8981 (0,0001)	0,8990 (0,0001)
	Jeffreys	0,8905 (0,0004)	0,8953 (0,0002)	0,8981 (0,0001)	0,8990 (0,0001)	0,8994 (0,0001)

Obs.: Melhores valores em negrito

Tabela 4.2: Valores exatos e estimativas médias para L e L_q

L	L_q	Distribuição	n									
			10		20		50		100		200	
			\hat{L}	\hat{L}_q	\hat{L}	\hat{L}_q	\hat{L}	\hat{L}_q	\hat{L}	\hat{L}_q	\hat{L}	\hat{L}_q
0,25	0,05	Beta(1, 5; 2, 5)	0,3232	0,1155	0,2912	0,0849	0,2670	0,0642	0,2593	0,0574	0,2543	0,0536
		Beta(2, 5; 1, 5)	0,4383	0,1831	0,3512	0,1139	0,2916	0,0743	0,2717	0,0622	0,2606	0,0559
		Uniforme	0,3667	0,1246	0,3068	0,0863	0,2707	0,0638	0,2608	0,0571	0,2550	0,0534
		Jeffreys	0,3040	0,1184	0,2776	0,0824	0,2603	0,0621	0,2558	0,0562	0,2525	0,0530
1,00	0,50	Beta(1, 5; 2, 5)	0,9249	0,5425	0,9591	0,5247	0,9811	0,5095	0,9920	0,5062	0,9947	0,5022
		Beta(2, 5; 1, 5)	1,0923	0,6905	1,0502	0,5993	1,0196	0,5394	1,0116	0,5212	1,0046	0,5097
		Uniforme	1,1064	0,6251	1,0540	0,5644	1,0202	0,5249	1,0118	0,5138	1,0047	0,5060
		Jeffreys	1,0561	0,6828	1,0290	0,5881	1,0102	0,5331	1,0068	0,5177	1,0022	0,5079
9,00	8,10	Beta(1, 5; 2, 5)	7,2939	7,0595	8,0376	7,5270	8,5815	7,8543	8,7799	7,9696	8,8808	8,0265
		Beta(2, 5; 1, 5)	8,0151	7,8993	8,4580	7,9810	8,7675	8,0453	8,8763	8,0667	8,9298	8,0755
		Uniforme	8,6913	7,8111	8,9495	8,0588	8,9996	8,1032	8,9944	8,0963	8,9893	8,0903
		Jeffreys	9,0173	9,1288	8,9923	8,5703	8,9906	8,2759	8,9894	8,1812	8,9868	8,1325

Obs.: Melhores valores em negrito

Tabela 4.3: Probabilidade preditiva *a posteriori* do número de clientes no sistema e o fator de Bayes

ρ	Nº de Clientes	Beta(1, 5; 2, 5)	Uniforme	Jeffreys	B_{12}	B_{13}	B_{23}
0,2	0	0,7980	0,7984	0,7992	1,0005	1,0015	1,0010
	1	0,1600	0,1597	0,1592	0,9981	0,9950	0,9969
	2	0,0330	0,0329	0,0327	0,9970	0,9909	0,9939
	3	0,0070	0,0070	0,0069	1,0000	0,9857	0,9857
	4	0,0015	0,0015	0,0015	1,0000	1,0000	1,0000
	5	0,0003	0,0003	0,0003	1,0000	1,0000	1,0000
0,5	0	0,5025	0,5013	0,5007	0,9976	0,9964	0,9988
	1	0,2488	0,2488	0,2488	1,0000	1,0000	1,0000
	2	0,1238	0,1241	0,1242	1,0024	1,0032	1,0008
	3	0,0619	0,0622	0,0623	1,0048	1,0065	1,0016
	4	0,0311	0,0313	0,0314	1,0064	1,0096	1,0032
	5	0,0157	0,0158	0,0159	1,0064	1,0127	1,0063
0,9	0	0,1017	0,1010	0,1006	0,9931	0,9892	0,9960
	1	0,0912	0,0907	0,0904	0,9945	0,9912	0,9967
	2	0,0819	0,0815	0,0812	0,9951	0,9915	0,9963
	3	0,0735	0,0732	0,0730	0,9959	0,9932	0,9973
	4	0,0660	0,0658	0,0656	0,9970	0,9939	0,9970
	5	0,0592	0,0591	0,0590	0,9983	0,9966	0,9983

4.2 Filas M/M/1/K

Para análise de desempenho dos estimadores, 10.000 replicações Monte Carlo foram realizadas, de acordo com o código, em R (R Core Team, 2013), apresentado na Listagem 4.3. Note-se que a função `fEst(K, amostra, ...)` pode ser qualquer uma das que implementam estimadores para ρ (EMV, bayesiano e EMV corrigido via *bootstrap*) e também que é armazenado o estado da semente aleatória `.GlobalEnv$.Random.seed`, imediatamente antes da chamada a esta função, que pode ser estocástica, para garantir que as amostras geradas para estimação serão as mesmas para todos os estimadores.

```
1 MtCaRoMM1K<-function(tam, rho, K, fEst, ...) {
  rep<-10000
3  amostra<-numeric(tam)
  est<-numeric(rep)
5  for (i in 1:rep) {
    # gerar dados
7    amostra<-rmm1k(tam, rho, K)
    oldseed<-GlobalEnv$.Random.seed
9    # ajustar modelo e guardar ajuste
    est[i]<-fEst(K, amostra, ...)
11   GlobalEnv$.Random.seed<-oldseed
  }
13  c(mean(est), sqrt(var(est)/rep))
}
```

Listagem 4.3: Implementação da simulação Monte Carlo

Foram realizadas simulações de amostras, a partir da Eq. (3.48) e código da Listagem 3.9, para os tamanhos $n \in \{10; 20; 50; 100; 200\}$ e as intensidades de tráfego $\rho \in \{0,01; 0,10; 0,20; 0,50; 0,90; 0,99\}$. Para todos esses cenários, foram calculadas médias, para 10.000 replicações Monte Carlo (via código da Listagem 4.3), das estimativas pontuais de ρ (i) pelo EMV, via maximização numérica da verossimilhança, Eq. (3.43) e código da Listagem 3.6; (ii) pelo método bayesiano,

com *priori* $\rho \sim \text{beta}(1, 5; 2, 5)$, via médias de amostras de tamanho 5.000 da distribuição a *posteriori*, Eq. (3.46), obtidas pelo método ARMS Gilks et al. (1995) do pacote HI Petris & Tardella (2013) do R, via código da Listagem 3.8; (iii) e estimativas pontuais pelo EMV, corrigidas pelo método *bootstrap*, Eq. (2.12) e código da Listagem 3.7. Também foram calculadas os respectivos erros padrão da média (EPM). Os resultados podem ser vistos nas Tabelas 4.4, 4.5 e 4.6, que se encontram sintetizados nas Figuras 4.8, 4.9 e 4.10.

Nas Figuras 4.8, 4.9 e 4.10, encontram-se representados os erros de estimação médios e os EPM médios, em função da intensidade de tráfego ρ (média entre todos os tamanhos de amostra) e em função do tamanho da amostra n (média entre todas as intensidades de tráfego). Para as filas com capacidade $K = 5$, Figura 4.8, notamos um erro de estimação médio aproximadamente constante para o EMV e EMV corrigido pelo *bootstrap*, exceto quando $\rho \approx 1,0$ (Figura 4.8-a). O estimador bayesiano não apresentou desempenho equivalente, inclusive as estimativas tendem a superestimar o valor verdadeiro (erro positivo), quando $\rho < 0,5$, e a subestimar, caso contrário. Em relação ao tamanho da amostra n , todos os estimadores apresentaram uma redução monotônica no erro (Figura 4.8-c). Do lado do EPM, que mede a variabilidade das estimativas, o estimador bayesiano apresenta-se como a melhor alternativa, uma vez que apresenta sempre os menores valores. Entre o EMV e EMV corrigido *bootstrap*, os valores são semelhantes, com uma vantagem para o EMV original.

Aumentando-se um pouco a capacidade da fila, para $K = 20$, Figura 4.9, notamos um comportamento bastante semelhante. Entretanto, a dificuldade de estimação para intensidades de tráfego $\rho \approx 1,0$ parece ter diminuído, sendo que os maiores erros de estimação médios ocorrem com $\rho \approx 0,5$, para os estimadores EMV e EMV corrigido. Este comportamento é aquele observado para filas markovianas infinitas, $M/M/1$, anteriormente estudadas. Por seu lado, o estimador bayesiano

Tabela 4.4: Estimativas médias de ρ e EPM (entre parênteses) para $K = 5$

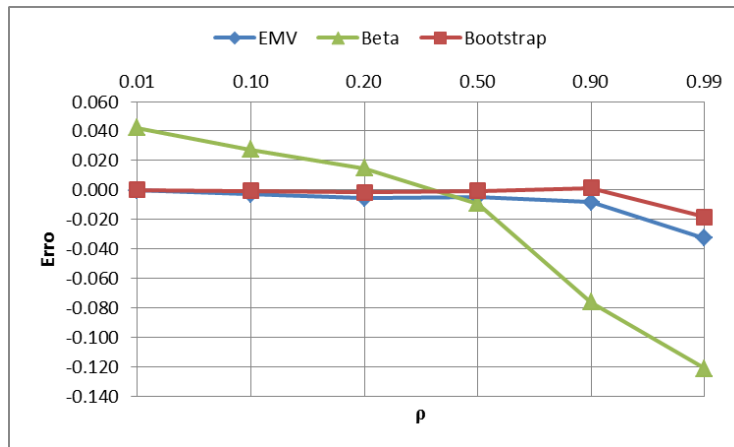
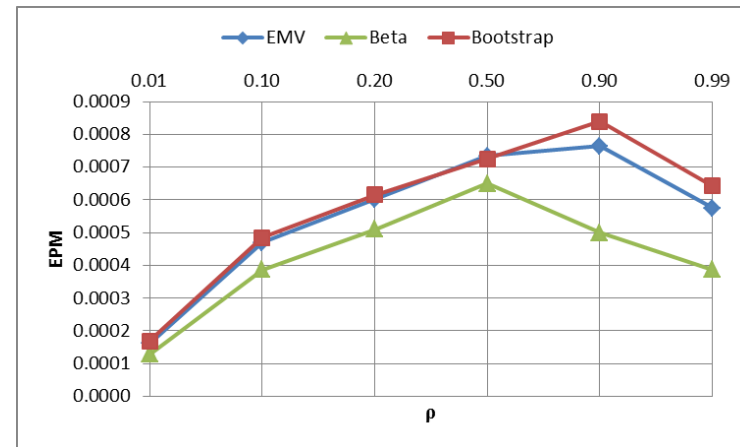
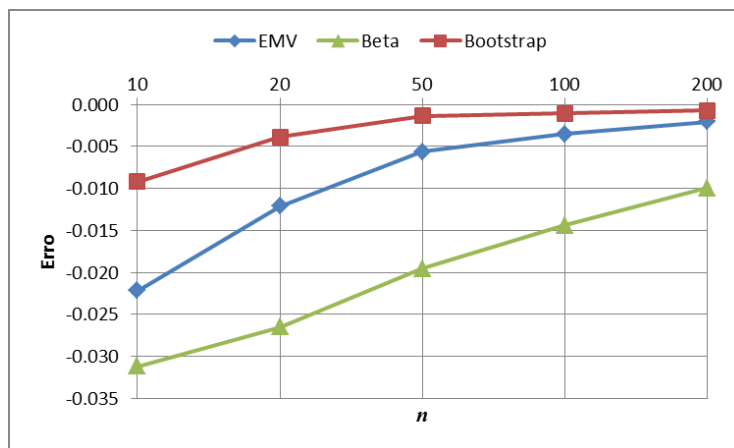
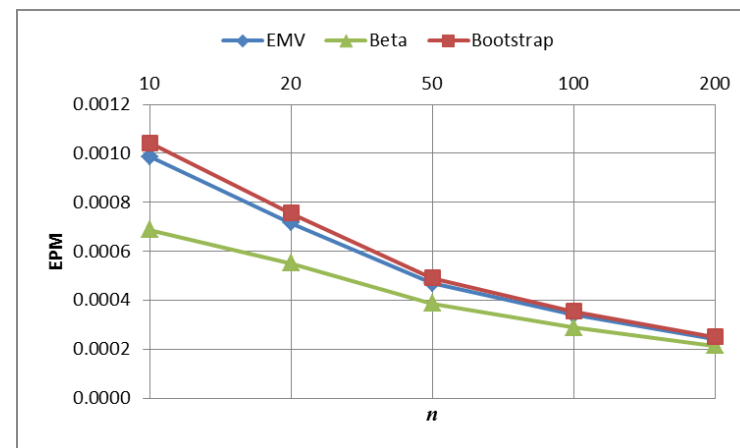
K	Estimador	ρ	n				
			10	20	50	100	200
5	EMV	0,01	0,00928 (0,00029)	0,00960 (0,00021)	0,01000 (0,00014)	0,01003 (0,00010)	0,01007 (0,00007)
		0,10	0,09172 (0,00084)	0,09614 (0,00062)	0,09845 (0,00040)	0,09907 (0,00028)	0,09964 (0,00020)
		0,20	0,18449 (0,00109)	0,19235 (0,00080)	0,19700 (0,00051)	0,19833 (0,00036)	0,19921 (0,00025)
		0,50	0,48492 (0,00139)	0,49422 (0,00096)	0,49824 (0,00060)	0,49884 (0,00042)	0,49952 (0,00030)
		0,90	0,87146 (0,00127)	0,88854 (0,00097)	0,89839 (0,00068)	0,90021 (0,00052)	0,90044 (0,00038)
		0,99	0,92503 (0,00103)	0,94638 (0,00074)	0,96396 (0,00049)	0,97225 (0,00036)	0,97854 (0,00026)
	beta(1, 5; 2, 5)	0,01	0,11359 (0,00019)	0,06994 (0,00017)	0,03646 (0,00012)	0,02366 (0,00009)	0,01704 (0,00007)
		0,10	0,17032 (0,00060)	0,13921 (0,00050)	0,11622 (0,00036)	0,10822 (0,00027)	0,10402 (0,00020)
		0,20	0,23721 (0,00081)	0,22125 (0,00067)	0,20848 (0,00047)	0,20457 (0,00035)	0,20214 (0,00025)
		0,50	0,47449 (0,00110)	0,48763 (0,00087)	0,49495 (0,00058)	0,49769 (0,00041)	0,49900 (0,00030)
		0,90	0,74065 (0,00078)	0,79179 (0,00061)	0,84140 (0,00045)	0,86479 (0,00037)	0,88099 (0,00030)
		0,99	0,77650 (0,00066)	0,83119 (0,00049)	0,88520 (0,00034)	0,91462 (0,00025)	0,93702 (0,00019)
	<i>Bootstrap</i>	0,01	0,00992 (0,00031)	0,01000 (0,00022)	0,01018 (0,00014)	0,01013 (0,00010)	0,01012 (0,00007)
		0,10	0,09746 (0,00089)	0,09970 (0,00064)	0,10009 (0,00041)	0,09993 (0,00029)	0,10007 (0,00020)
		0,20	0,19446 (0,00113)	0,19837 (0,00082)	0,19976 (0,00051)	0,19977 (0,00036)	0,19994 (0,00025)
		0,50	0,49647 (0,00137)	0,50011 (0,00095)	0,50065 (0,00059)	0,50007 (0,00042)	0,50012 (0,00030)
		0,90	0,89517 (0,00141)	0,90321 (0,00108)	0,90473 (0,00076)	0,90256 (0,00056)	0,90074 (0,00039)
		0,99	0,95119 (0,00114)	0,96522 (0,00083)	0,97635 (0,00055)	0,98100 (0,00040)	0,98475 (0,00030)

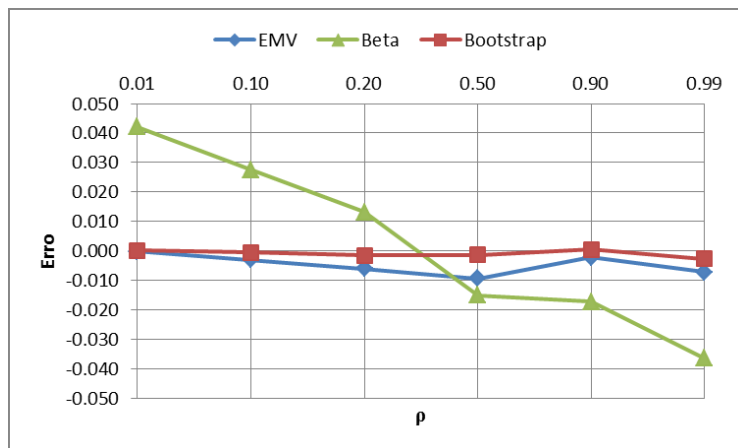
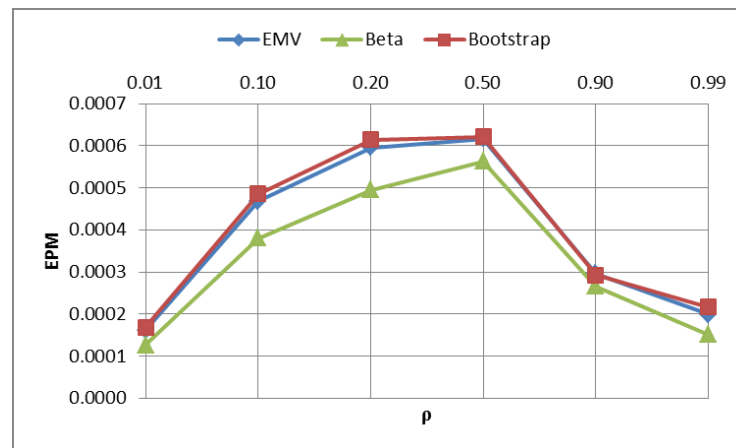
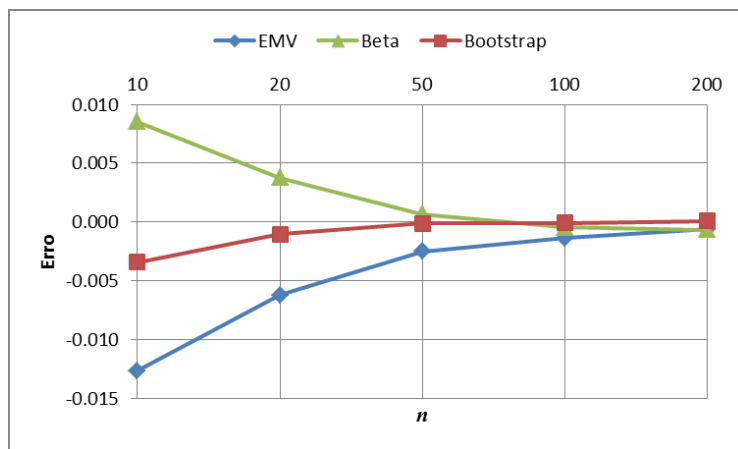
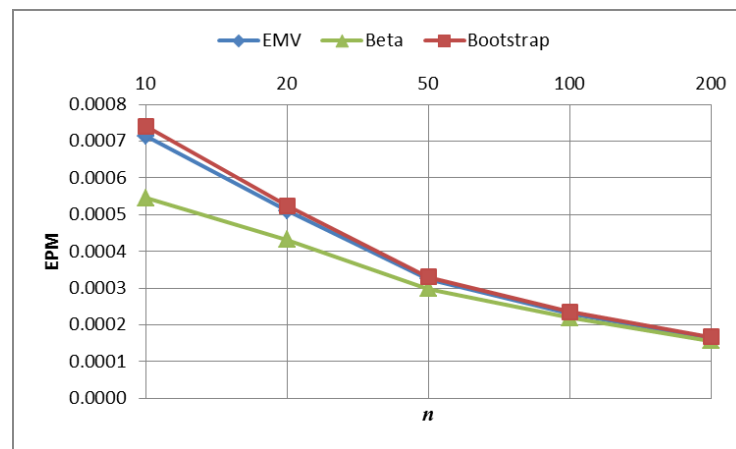
Tabela 4.5: Estimativas médias de ρ e EPM (entre parênteses) para $K = 20$

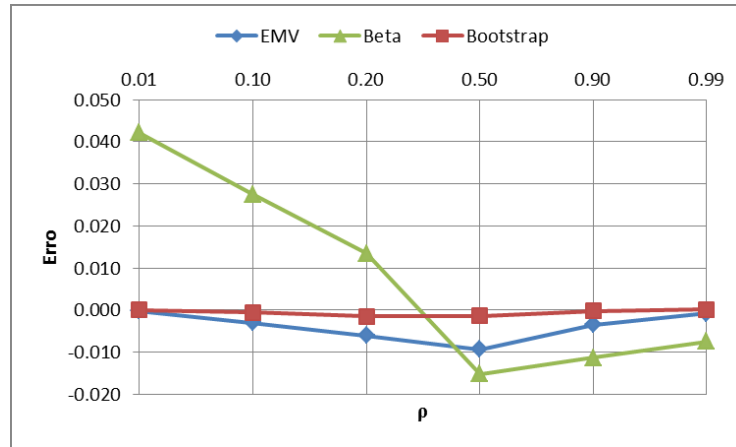
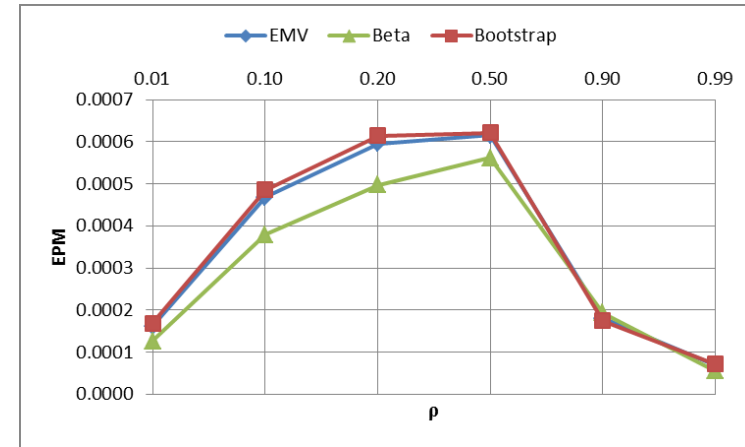
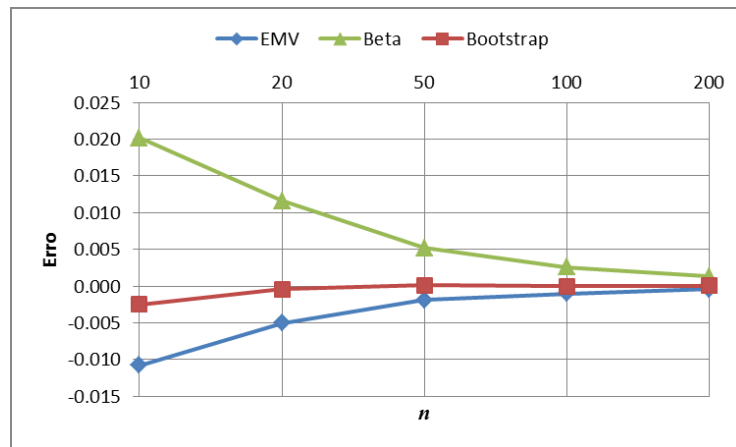
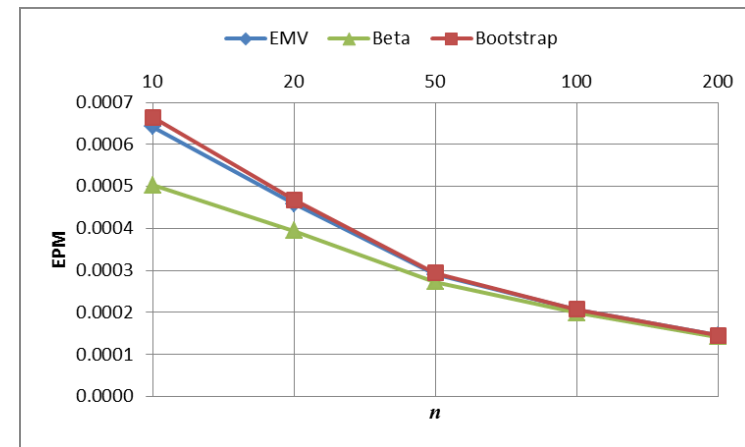
K	Estimador	ρ	n				
			10	20	50	100	200
20	EMV	0,01	0,00928 (0,00029)	0,00960 (0,00021)	0,01000 (0,00014)	0,01003 (0,00010)	0,01007 (0,00007)
		0,10	0,09158 (0,00084)	0,09609 (0,00062)	0,09844 (0,00040)	0,09906 (0,00028)	0,09964 (0,00020)
		0,20	0,18345 (0,00107)	0,19186 (0,00079)	0,19684 (0,00050)	0,19827 (0,00036)	0,19919 (0,00025)
		0,50	0,47371 (0,00116)	0,48785 (0,00081)	0,49543 (0,00050)	0,49750 (0,00036)	0,49885 (0,00025)
		0,90	0,89316 (0,00057)	0,89758 (0,00039)	0,89922 (0,00024)	0,89956 (0,00017)	0,89979 (0,00012)
		0,99	0,97283 (0,00036)	0,97979 (0,00025)	0,98507 (0,00017)	0,98728 (0,00013)	0,98877 (0,00010)
	beta(1, 5; 2, 5)	0,01	0,11295 (0,00018)	0,06982 (0,00017)	0,03639 (0,00012)	0,02396 (0,00009)	0,01704 (0,00007)
		0,10	0,16848 (0,00057)	0,13939 (0,00050)	0,11722 (0,00037)	0,10846 (0,00027)	0,10397 (0,00020)
		0,20	0,23420 (0,00077)	0,21768 (0,00066)	0,20773 (0,00046)	0,20430 (0,00034)	0,20221 (0,00025)
		0,50	0,46271 (0,00098)	0,47892 (0,00074)	0,49092 (0,00049)	0,49474 (0,00035)	0,49781 (0,00025)
		0,90	0,85476 (0,00048)	0,87577 (0,00034)	0,89079 (0,00023)	0,89523 (0,00017)	0,89754 (0,00012)
		0,99	0,91812 (0,00028)	0,94089 (0,00019)	0,96086 (0,00012)	0,97061 (0,00009)	0,97753 (0,00007)
<i>Bootstrap</i>	0,01	0,00993 (0,00031)	0,01000 (0,00022)	0,01018 (0,00014)	0,01013 (0,00010)	0,01012 (0,00007)	
	0,10	0,09763 (0,00089)	0,09976 (0,00064)	0,10010 (0,00041)	0,09994 (0,00029)	0,10007 (0,00020)	
	0,20	0,19462 (0,00113)	0,19846 (0,00081)	0,19982 (0,00051)	0,19982 (0,00036)	0,19997 (0,00025)	
	0,50	0,49385 (0,00118)	0,49896 (0,00081)	0,50019 (0,00050)	0,49995 (0,00036)	0,50008 (0,00025)	
	0,90	0,90117 (0,00056)	0,90065 (0,00038)	0,90021 (0,00024)	0,90004 (0,00017)	0,90002 (0,00012)	
	0,99	0,98215 (0,00038)	0,98590 (0,00027)	0,98872 (0,00019)	0,98963 (0,00014)	0,99023 (0,00011)	

Tabela 4.6: Estimativas médias de ρ e EPM (entre parênteses) para $K = 80$

K	Estimação	ρ	n				
			10	20	50	100	200
80	EMV	0,01	0,00928 (0,00029)	0,00960 (0,00021)	0,01000 (0,00021)	0,01003 (0,00010)	0,01007 (0,00007)
		0,10	0,09158 (0,00084)	0,09609 (0,00062)	0,09844 (0,00062)	0,09906 (0,00028)	0,09964 (0,00020)
		0,20	0,18345 (0,00107)	0,19186 (0,00079)	0,19684 (0,00079)	0,19827 (0,00036)	0,19919 (0,00025)
		0,50	0,47366 (0,00116)	0,48783 (0,00081)	0,49543 (0,00081)	0,49750 (0,00036)	0,49885 (0,00025)
		0,90	0,89001 (0,00037)	0,89551 (0,00023)	0,89835 (0,00023)	0,89915 (0,00010)	0,89960 (0,00007)
		0,99	0,98737 (0,00012)	0,98903 (0,00009)	0,98982 (0,00009)	0,98995 (0,00004)	0,98998 (0,00003)
	beta(1, 5; 2, 5)	0,01	0,11310 (0,00018)	0,06992 (0,00017)	0,03640 (0,00017)	0,02396 (0,00009)	0,01704 (0,00007)
		0,10	0,16896 (0,00057)	0,13844 (0,00050)	0,11715 (0,00050)	0,10872 (0,00027)	0,10440 (0,00019)
		0,20	0,23400 (0,00078)	0,21867 (0,00065)	0,20822 (0,00065)	0,20430 (0,00034)	0,20206 (0,00025)
		0,50	0,46323 (0,00098)	0,47879 (0,00074)	0,49021 (0,00074)	0,49475 (0,00035)	0,49723 (0,00025)
		0,90	0,86916 (0,00040)	0,88449 (0,00025)	0,89418 (0,00025)	0,89688 (0,00010)	0,89855 (0,00007)
		0,99	0,97255 (0,00010)	0,97946 (0,00007)	0,98483 (0,00007)	0,98708 (0,00003)	0,98852 (0,00003)
	<i>Bootstrap</i>	0,01	0,00993 (0,00031)	0,01000 (0,00022)	0,01018 (0,00014)	0,01013 (0,00010)	0,01012 (0,00007)
		0,10	0,09763 (0,00089)	0,09976 (0,00064)	0,10010 (0,00041)	0,09994 (0,00029)	0,10007 (0,00020)
		0,20	0,19462 (0,00113)	0,19846 (0,00081)	0,19982 (0,00051)	0,19982 (0,00036)	0,19997 (0,00025)
		0,50	0,49390 (0,00118)	0,49897 (0,00081)	0,50019 (0,00050)	0,49995 (0,00036)	0,50008 (0,00025)
		0,90	0,89910 (0,00034)	0,89987 (0,00023)	0,90008 (0,00014)	0,90002 (0,00010)	0,90003 (0,00007)
		0,99	0,99000 (0,00013)	0,99042 (0,00009)	0,99031 (0,00006)	0,99012 (0,00004)	0,99002 (0,00003)

(a) erro médio em função de ρ para $K = 5$ (b) EPM médio em função de ρ para $K = 5$ (c) erro médio em função de n para $K = 5$ (d) EPM médio em função de n para $K = 5$ Figura 4.8: Desempenho dos estimadores para ρ para $K = 5$

(a) erro médio em função de ρ para $K = 20$ (b) EPM médio em função de ρ para $K = 20$ (c) erro médio em função de n para $K = 20$ (d) EPM médio em função de n para $K = 20$ Figura 4.9: Desempenho dos estimadores para ρ para $K = 20$

(a) erro médio em função de ρ para $K = 80$ (b) EPM médio em função de ρ para $K = 80$ (c) erro médio em função de n para $K = 80$ (d) EPM médio em função de n para $K = 80$ Figura 4.10: Desempenho dos estimadores para ρ para $K = 80$

mantem seu padrão de desempenho, que é pior, em termos de erro de estimação médio. As estimativas são superestimadas para $\rho < 0,5$ e subestimadas, caso contrário. Todos os estimadores apresentam erros que convergem para zero, à medida que o tamanho da amostra cresce, com destaque para o estimador EMV corrigido, que apresenta erro de estimação médio aproximadamente zero, para amostras $n \geq 50$. Do ponto de vista do EPM, novamente os menores valores são aqueles obtidos para o estimador bayesiano, que parece apresentar a menor variabilidade, embora não tenha os menores erros de estimação médios.

Finalmente, para filas com $K = 80$, Figura 4.10, o comportamento observado pode ser considerado, para fins práticos, como sendo igual ao das filas markovianas infinitas, em termos de erros de estimação médios e EPM. Esta constatação é uma evidência da correção das nossas implementações e da qualidade dos resultados computacionais apresentados.

4.3 Observações Finais

Neste capítulo foram apresentados os principais resultados, com a metodologia proposta, utilizando a inferência clássica e bayesiana, para sistema de filas simples infinitas e finitas. No capítulo seguinte será apresentado as principais conclusões obtidas e algumas recomendações para trabalhos futuros.

Capítulo 5

Conclusões e Observações Finais

5.1 Sumário

Nesta tese, foi apresentada uma continuidade, uma extensão, do trabalho de [Choudhury & Borthakur \(2008\)](#). Foram desenvolvidas ferramentas inferenciais para filas markovianas de servidor único infinitas e finitas, ou, na notação de [Kendall \(1953\)](#), filas $M/M/1$ e $M/M/1/K$. Foram consideradas apenas amostras do número de clientes no sistema nos momentos de partidas aleatórias. Dada a importância das filas no dia a dia das pessoas, esta tese abordou o problema da estimação de algumas das suas medidas de desempenho, mais especificamente, da intensidade de tráfego, ρ , do número esperado de clientes no sistema, L , e do tamanho médio da fila, L_q . Apesar de o foco ter sido nos tipos mais simples de filas, $M/M/1$ e $M/M/1/K$, isto é, com chegadas Poisson, tempos de serviço exponenciais, servidor único e capacidades infinita ou finita, tais modelos são uma aproximação acurada para vários problemas práticos e uma base sólida para modelos mais complexos e específicos.

Foi constatada a presença de vício no estimador de máxima verossimilhança para ρ , para valores próximos de 0,5, e investigada a utilização do conhecido mé-

todo *bootstrap* não paramétrico para sua correção. Resultados computacionais apresentados apontaram para uma correção efetiva do vício pelo *bootstrap*, sobretudo em amostras pequenas, o que possibilita uma redução no seu tamanho, com conseqüente diminuição nos tempos para sua obtenção e nos custos de estimação. Sob ponto de vista bayesiano, outras distribuições *a priori* foram utilizadas. Observa-se que com o uso da distribuição *a priori* de Jeffreys, as estimativas também apresentam um vício menor, e, com o aumento do tamanho amostral, este vício tende a diminuir.

Para o modelo $M/M/1/K$, o problema de estimação da intensidade de tráfego apresenta-se como bastante desafiador. De fato, nenhum estimador foi absolutamente superior a outro, considerando os quesitos erro de estimação e EPM. A estimação pelos métodos EMV e EMV corrigido apresenta menor vício que o estimador bayesiano, mas apresenta uma maior variabilidade. Possivelmente pela assimetria da distribuição *a posteriori*, o estimador bayesiano não foi competitivo, em termos de erro de estimação. Em geral, para amostras de tamanho $n = 50$ e filas com $K \geq 20$, os erros de estimação médios foram inferiores a 0,005, superando este valor apenas para o estimador bayesiano, em filas de capacidade total $K = 5$.

No que diz respeito ao comportamento dos erros de estimação médios e EPM médios em função da intensidade de tráfego ρ , podemos confirmar os maiores erros quando as amostras são pequenas ($n \leq 20$) e as intensidades de tráfego $\rho \approx 1,0$, diferentemente do que ocorre com as filas $M/M/1$, que possuem maior vício para $\rho \approx 0,5$. Possivelmente devido ao truncamento do número de usuários no tamanho máximo da fila, K , os sistemas com altas intensidades de tráfego apresentam-se como os que exigem maior esforço computacional e que são os mais difíceis de estimar.

É importante ressaltar que, para filas com capacidade $K = 80$, o sistema comporta-se de forma similar às filas markovianas infinitas, $M/M/1$, conforme

esperado. Isto é, o erro de estimação médio é maior e o EPM é mais alto quando $\rho \approx 0,5$.

5.2 Propostas de Continuidade

Como trabalhos futuros, pode-se citar os seguintes:

- Fazer uma aplicação da metodologia apresentada a dados reais;
- Investigar as estimativas das medidas de desempenho utilizando outras formas de informação *a priori*;
- Estender o estudo de filas $M/M/1/K$ para outras medidas de desempenho, tais como o número de clientes no sistema, L , tamanho médio da fila, L_q , entre outras;
- Estender o processo inferencial para filas com servidores múltiplos, modelos $M/M/c$ e $M/M/\infty$;
- Utilizar critérios de comparação de modelos para analisar o impacto do uso da informação *a priori* nas estimativas;
- Realizar uma análise de custo para os modelos estudados.

Investigações futuras também incluem a avaliação da qualidade da correção via *bootstrap* em outras medidas de desempenho da fila, tais como a probabilidade de servidor ocioso, $P(M = 0)$, o tempo esperado no sistema, W , e o tempo médio na fila, W_q . Também outros tipos de filas devem ser investigados, tais como filas markovianas multi-servidoras infinitas, $M/M/c$, finitas, $M/M/c/K$, e assim por diante. Estes são apenas alguns tópicos para trabalhos futuros nesta instigante linha de pesquisa.

Referências Bibliográficas

- Abramowitz, M. & Stegun, I. A. (1972). *Handbook of Mathematical Functions*, Dover, New York, NY, chapter Chapter 6: Gamma and Related Functions.
- Aigner, D. J. (1974). Technical note—parameter estimation from cross-sectional observations on an elementary queuing system, *Operations Research* **22**(2): 422–428. **URL:** <http://dx.doi.org/10.1287/opre.22.2.422>
- Almeida, M. A. C. & Cruz, F. R. B. (2015a). Análise de desempenho em filas $M/M/1$ usando uma abordagem bayesiana, *Congresso de Matemática Aplicada e Computacional - III CMAC-Sudeste 2015*, n. 2 in *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, Vitória, ES, Brasil, pp. 0200571–0200577. **URL:** <http://dx.doi.org/10.5540/03.2015.003.02.0057>
- Almeida, M. A. C. & Cruz, F. R. B. (2015b). A Bayesian approach to performance analysis of $M/M/1$ queues. (em revisão).
- Alouf, S., Nain, P. & Towsley, D. (2001). Inferring network characteristics via moment-based estimators, *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 2, pp. 1045–1054 vol.2.
- Armero, C. (1994). Bayesian inference in markovian queues, *Queueing Systems* **15**(1): 419–426.

- Armero, C. & Bayarri, M. J. (1994a). Bayesian prediction in $M/M/1$ queues, *Queueing Systems* **15**(1-4): 401–417. **URL:** <http://dx.doi.org/10.1007/BF01189248>
- Armero, C. & Bayarri, M. J. (1994b). Prior assessments for prediction in queues, *Journal of the Royal Statistical Society. Series D (The Statistician)* **43**(1): 139–153.
- Armero, C. & Bayarri, M. J. (1999). Multivariate analysis, design of experiments and survey sampling, in S. Ghosh (ed.), *Dealing with uncertainties in queues and networks of queues: A Bayesian approach*, Springer Science+Business Media, Marcel Dekker, New York, NY, pp. 579–608.
- Armero, C. & Bayarri, M. J. (2000). Queues, *ISDS Discussion paper No 39*.
- Bhat, U. N. (2008). *An Introduction to Queueing Theory: Modeling and Analysis in Applications*, (Statistics for Industry and Technology), Springer, Dordrecht.
- Choudhury, A. & Borthakur, A. C. (2008). Bayesian inference and prediction in the single server Markovian queue, *Metrika* **67**(3): 371–383. **URL:** <http://dx.doi.org/10.1007/s00184-007-0138-3>
- Clarke, A. B. (1957). Maximum likelihood estimates in a simple queue, *The Annals of Mathematical Statistics* **28**(4): 1036–1040. **URL:** <http://dx.doi.org/10.1214/aoms/1177706808>
- Cruz, F. R. B., Colosimo, E. A. & Smith, J. M. (2004). Sample size corrections for the maximum partial likelihood estimator, *Communications in Statistics - Simulation & Computation* **33**(1): 35–47.
- Domingues, K. M., Oliveira, F. L. P., Cruz, F. R. B. & Bessegato, L. F. (2015). Estimação de intervalos de confiança via reamostragem bootstrap, *I Encontro Interestadual de Engenharia de Produção - EINEPro*

- 2015, *Anais [CD-ROM]*, São João da Barra, RJ, Brasil, pp. 1–12. **URL:**
<ftp://ftp.est.ufmg.br/pub/fcruz/publics/einep-icboot.pdf>
- Efron, B. & Tibshirani, R. (1993). *An Introduction to the Bootstrap*, Chapman & Hall, London, UK.
- Ehlers, R. S. (2003). Introdução à inferência bayesiana. **URL:**
<http://www.leg.ufpr.br/%7Epaulojus/CE227/ce227.pdf>
- Evert, S. & Baroni, M. (2006). The zipfR library: Words and other rare events in R, *Presentation at useR! 2006: The Second R User Conference*, Vienna, Austria. **URL:** <http://purl.org/stefan.evert/zipfR/>
- Evert, S. & Baroni, M. (2007). *zipfR*: Word frequency distributions in R, *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Posters and Demonstrations Sessions*, Prague, Czech Republic, pp. 29–32. (R package version 0.6-6 of 2012-04-03).
- Gamerman, D. & Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, 2 ed., Chapman and Hall/CRC, London, UK.
- Gilks, W. R., Best, N. G. & Tan, K. K. C. (1995). Adaptive rejection Metropolis sampling within Gibbs sampling, *Applied Statistics* pp. 455–472.
- Gross, D., Shortle, J. F., Thompson, J. M. & Harris, C. M. (2009). *Fundamentals of Queueing Theory*, 4 ed., Wiley-Interscience, New York, NY, USA.
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**(1): 97–109.
- Jeffreys, H. (1998). *The Theory of Probability*, OUP Oxford. **URL:**
<http://books.google.com.br/books?id=vh9Act9rtzQC>

- Kannan, K. S. & Jabarali, A. (2014). Parameter estimation of single server queue with working vacations, *Research & Reviews: Journal of Statistics (Special Issue on Recent Statistical Methodologies and Applications)* **2**: 94–98.
- Kendall, D. G. (1953). Stochastic processes occurring in the theory of queues and their analysis by the method of embedded Markov chains, *Annals Mathematical Statistics* **24**: 338–354.
- Kuumola, E., Resing, J. & Virtamo, J. (2002). Joint distribution of instantaneous and averaged queue length in an $M/M/1/K$ system, *Proceedings of the 15th ITC specialist seminar "Internet traffic engineering and traffic management"*, pp. 58–67.
- Mcgrath, M. F., Gross, D. & Singpurwalla, N. D. (1987a). A subjective Bayesian approach to the theory of queues I - modeling, *Queueing Systems* **1**(4): 317–333.
- Mcgrath, M. F., Gross, D. & Singpurwalla, N. D. (1987b). A subjective Bayesian approach to the theory of queues II - modeling, *Queueing Systems* **1**(4): 335–353.
- Medhi, J. (2003). *Stochastic Models in Queueing Theory*, 2 edn, Academic Press.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). Equation of state calculations by fast computing machines, *The Journal of Chemical Physics* **21**(6): 1087–1092.
- Montgomery, D. C. & Runger, G. C. (2012). *Estatística Aplicada e Probabilidade Para Engenheiros*, 5 ed., LTC - Livros Técnicos e Científicos Ltda.
- Muddapur, M. V. (1972). Bayesian estimates of parameters in some queueing models, *Annals of the Institute of Statistical Mathematics* **24**(1): 327–331. **URL:** <http://dx.doi.org/10.1007/BF02479762>

- Mukhopadhyay, N. (2000). *Probability and Statistical Inference*, Marcel Dekker, New York, NY.
- Paulino, C. D., Turkman, M. A. A. & Murteira, B. (2003). *Estatística Bayesiana*, Fundação Calouste Gulbenkian, Lisboa.
- Pereira, L. C. R., Santos, M. A. C., Cruz, F. R. B. & Oliveira, F. L. P. (2015). Inferência e correção de vício em filas markovianas de servidor Único, *II Encontro Goiano de Probabilidade e Estatística - II EnGOPE*, Goiânia, GO, Brasil, pp. 1–12. **URL:** <ftp://ftp.est.ufmg.br/pub/fcruz/publics/engope-inf.pdf>
- Petris, G. & Tardella, L. (2013). *HI: Simulation from distributions supported by nested hyperplanes*. (original C code for ARMS by Wally R. Gilks; R package version 0.4). **URL:** <http://CRAN.R-project.org/package=HI>
- R Core Team (2013). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. **URL:** <http://www.R-project.org/>
- Schruben, L. & Kulkarni, R. (1982). Some consequences of estimating parameters for the $M/M/1$ queue, *Operations Research Letters* **1**(2): 75–78.
- Wagner, H. M. (1986). *Pesquisa Operacional*, 2 ed., Prentice-Hall do Brasil Ltda., Rio de Janeiro.
- Wolfram Research Inc. (2015). Generalized incomplete beta function. Accessed: 2015-11-09. **URL:** <http://functions.wolfram.com/GammaBetaErf/Beta4/>
- Zacharias, H. P., Leite, J. G. & Diniz, C. A. R. (2004). O uso de prioris não informativas para estimação do tamanho populacional, *TEMA - Trends in Applied and Computational Mathematics* **5**(1): 165–175.

Zheng, S. & Seila, A. F. (2000). Some well-behaved estimators for the $M/M/1$ queue, *Operations Research Letters* **26**(5): 231–235.

Apêndice A

Listagem dos Programas em R para Filas $M/M/1$

MM1_160216.R

```
#####  
# functions for M/M/1 queue  
#####  
gerAmosRo<-function(tam, rho) { # simulate  
  return(rgeom(tam, 1-rho))  
#####  
# maximum likelihood estimate for rho, L, and Lq  
#####  
EMVRO<-function(amostra) {  
  return(1/(1+length(amostra)/sum(amostra)))  
}  
EMVL<-function(amostra) {  
  return(1/(1/EMVRO(amostra)-1))  
}  
EMVLq<-function(amostra) {  
  roHat<-EMVRO(amostra)  
  return(roHat^2/(1-roHat))  
}#####  
# Bootstrap-corrected estimate  
#####  
ECBoot<-function(amostra, fEst) {  
  B<-10  
  soma<-0  
  for (i in 1:B) {  
    reamostra<-sample(amostra, replace=T)  
    estm<-fEst(reamostra)  
    soma<-soma+estm}  
  estmStar=soma/B  
  return(2*fEst(amostra)-estmStar)}  
#####
```

```

# Bayesian estimate rho ~ Beta(a,b), L, and Lq
#####
EPRo1<-function(amostra , a, b) {
  n<-length(amostra)
  y<-sum(amostra)
  return((a+y)/(y+a+n+b))}
EPL1<-function(amostra , a, b) {
  n<-length(amostra)
  y<-sum(amostra)
  return((a+y)/(n+b))}
EPLq1<-function(amostra , a, b) {
  n<-length(amostra)
  y<-sum(amostra)
  return(((a+y+1)*(y+a)/((n+b-1)*(y+a+n+b))))}
#####
# Bayesian estimate rho ~ Unif(c,d), L, and Lq
#####
library(zipfR)
require(zipfR) # required package
EPRo2<-function(amostra , c, d) {
  n<-length(amostra)
  y<-sum(amostra)
  return(((Ibeta(c,y+2,n+1)-Ibeta(d,y+2,n+1))/
    (Ibeta(c,y+1,n+1)-Ibeta(d,y+1,n+1))))}
EPL2<-function(amostra , c, d) {
  n<-length(amostra)
  y<-sum(amostra)
  return(((Ibeta(c,y+2,n)-Ibeta(d,y+2,n))/
    (Ibeta(c,y+1,n+1)-Ibeta(d,y+1,n+1))))}
EPLq2<-function(amostra , c, d) {
  n<-length(amostra)
  y<-sum(amostra)
  return(((Ibeta(c,y+3,n)-Ibeta(d,y+3,n))/
    (Ibeta(c,y+1,n+1)-Ibeta(d,y+1,n+1))))}
#####
# Bayesian estimate rho ~ Jeffreys, L, and Lq
#####
EPRo3<-function(amostra) {
  n<-length(amostra)
  y<-sum(amostra)
  return((1/2+y)/(y+1/2+n))}
EPL3<-function(amostra) {
  n<-length(amostra)
  y<-sum(amostra)
  return((1/2+y)/n)}
EPLq3<-function(amostra) {
  n<-length(amostra)
  y<-sum(amostra)
  return(((y+3/2)*(y+1/2)/((n-1)*(y+1/2+n))))}
#####
# predictive probability
#####
PPL<-function(amostra , a, b, m) {
  n<-length(amostra)
  y<-sum(amostra)
  return(beta(y+a+m, n+b+1)/beta(y+a, n+b))}

```

```

PP2<-function(amostra , c , d , m) {
  n<-length(amostra)
  y<-sum(amostra)
  return((Ibeta(c,y+1+m,n+2)-Ibeta(d,y+1+m,n+2))/
    (Ibeta(c,y+1,n+1)-Ibeta(d,y+1,n+1)))}
PP3<-function(amostra , m) {
  n<-length(amostra)
  y<-sum(amostra)
  return(beta(y+1/2+m,n+1)/beta(y+1/2,n))}
#####
# monte carlo estimates
#####
MtCa<-function(tam , rho , fEst , ... ) {
  rep<-10000
  amostra<-numeric(tam)
  est<-numeric(rep)
  for ( i in 1:rep) {
    # gerar dados
    amostra<-gerAmosRo(tam , rho)
    oldseed<-GlobalEnv$.Random.seed
    # ajustar modelo e guardar ajuste
    est [ i ]<-fEst(amostra , ... )
    .GlobalEnv$.Random.seed<-oldseed}
  return(c(mean(est) , sqrt(var(est)/rep)))}
#####
# monte carlo tables
#####
TabMtCa<-function(tam , rho , fEst , ... ) {
  res<-matrix(nrow=length(rho) , ncol=2*length(tam))
  for ( i in 1:length(rho)) {
    for ( j in 1:length(tam)) {
      set.seed(13579)
      res [ i , (2*j-1):(2*j) ]<-MtCa(tam [ j ] , rho [ i ] , fEst , ... )}
  return(res)}
#####
# generate MLE tables for rho
#####
tam<-c(10,20,50,100,200)
rho=c(0.01,0.10,0.20,0.50,0.90,0.99)
r.EMVRo<-TabMtCa(tam , rho , EMVRo)
save(tam , rho , r.EMVRo , file="TabEMVRo.RData")
#load(file="TabEMVRo.RData")
#####
# generate bootstrap tables for rho
#####
tam<-c(10,20,50,100,200)
rho=c(0.01,0.10,0.20,0.50,0.90,0.99)
r.ECBootRo<-TabMtCa(tam , rho , ECBoot , EMVRo)
save(tam , rho , r.EMVRo , file="TabECBootRo.RData")
#load(file="TabECBootRo.RData")
#####
# expected number of clients L and queue size Lq
#####
espL<-function(rho) {
  return(1/(1/rho-1))}
espLq<-function(rho) {

```



```

    return(rho^2/(1-rho))}
#####
# inverse functions; rho as a function of L and Lq
#####
fRoL<-function(L) {
  return(1/(1+1/L))}
fRoLq<-function(Lq) {
  return((-Lq+sqrt(Lq^2+4*Lq))/2)}
#####
# generate MLE tables for L
#####
tam<-c(10,20,50,100,200)
L=c(0.5,1,2,5,10,20)
#r.EMVL<-TabMtCa(tam,fRoL(L),EMVL)
r.EMVL<-TabMtCa(tam,fRoL(L),EMVRO)
r.EMVL[, (1:(2*length(tam))%%2)!=0]<-espL(r.EMVL[, (1:(2*length(tam))%%2)!=0])
save(tam,L,r.EMVL, file="TabEMVL.RData")
#load(file="TabEMVL.RData")
#####
# generate bootstrap tables for L
#####
tam<-c(10,20,50,100,200)
L=c(0.5,1,2,5,10,20)
#r.ECBootL<-TabMtCa(tam,fRoL(L),ECBoot,EMVL)
r.ECBootL<-TabMtCa(tam,fRoL(L),ECBoot,EMVRO)
r.ECBootL[, (1:(2*length(tam))%%2)!=0]<-espL(r.ECBootL[, (1:(2*length(tam))%%2)!=0])
save(tam,L,r.ECBootL, file="TabECBootL.RData")
#load(file="TabECBootL.RData")
#####
# generate MLE tables for Lq
#####
tam<-c(10,20,50,100,200)
Lq=c(0.5,1,2,5,10,20)
#r.EMVLq<-TabMtCa(tam,fRoLq(Lq),EMVLq)
r.EMVLq<-TabMtCa(tam,fRoLq(Lq),EMVRO)
r.EMVLq[, (1:(2*length(tam))%%2)!=0]<-espLq(r.EMVLq[, (1:(2*length(tam))%%2)!=0])
save(tam,Lq,r.EMVLq, file="TabEMVLq.RData")
#load(file="TabEMVLq.RData")
#####
# generate bootstrap tables for Lq
#####
tam<-c(10,20,50,100,200)
Lq=c(0.5,1,2,5,10,20)
#r.ECBootLq<-TabMtCa(tam,fRoLq(Lq),ECBoot,EMVLq)
r.ECBootLq<-TabMtCa(tam,fRoLq(Lq),ECBoot,EMVRO)
r.ECBootLq[, (1:(2*length(tam))%%2)!=0]<-espLq(r.ECBootLq[, (1:(2*length(tam))%%2)!=0])
save(tam,L,r.ECBootL, file="TabECBootL.RData")
#load(file="TabECBootL.RData")
#####
# Bayesian estimates for rho, L, and Lq
#####
# rho ~ B(1.5;2.5)
#####
tam<-c(10,20,50,100,200)
rho=c(0.20,0.50,0.90)
a<-1.5

```

```

b<-2.5
r.EPRo1<-TabMtCa(tam,rho,EPRo1,a,b)
save(tam,rho,a,b,r.EPRo1,file="TabEPRo1.RData")
#load(file="TabEPRo1.RData")
r.EPL1<-TabMtCa(tam,rho,EPL1,a,b)
save(tam,rho,a,b,r.EPL1,file="TabEPL1.RData")
#load(file="TabEPL1.RData")
r.EPLq1<-TabMtCa(tam,rho,EPLq1,a,b)
save(tam,rho,a,b,r.EPLq1,file="TabEPLq1.RData")
#load(file="TabEPLq1.RData")
#####
# rho ~ B(2.5;1.5)
#####
tam<-c(10,20,50,100,200)
rho=c(0.20,0.50,0.90)
a<-2.5
b<-1.5
r.EPRo1b<-TabMtCa(tam,rho,EPRo1,a,b)
save(tam,rho,a,b,r.EPRo1b,file="TabEPRo1b.RData")
#load(file="TabEPRo1b.RData")
r.EPL1b<-TabMtCa(tam,rho,EPL1,a,b)
save(tam,rho,a,b,r.EPL1b,file="TabEPL1b.RData")
#load(file="TabEPL1b.RData")
r.EPLq1b<-TabMtCa(tam,rho,EPLq1,a,b)
save(tam,rho,a,b,r.EPLq1b,file="TabEPLq1b.RData")
#load(file="TabEPLq1b.RData")
#####
# rho ~ U(0.05;0.95)
#####
tam<-c(10,20,50,100,200)
rho=c(0.20,0.50,0.90)
c<-0.05
d<-0.95
r.EPRo2<-TabMtCa(tam,rho,EPRo2,c,d)
save(tam,rho,a,b,r.EPRo2,file="TabEPRo2.RData")
#load(file="TabEPRo2.RData")
r.EPL2<-TabMtCa(tam,rho,EPL2,c,d)
save(tam,rho,a,b,r.EPL2,file="TabEPL2.RData")
#load(file="TabEPL2.RData")
r.EPLq2<-TabMtCa(tam,rho,EPLq2,c,d)
save(tam,rho,a,b,r.EPLq2,file="TabEPLq2.RData")
#load(file="TabEPLq2.RData")
#####
# rho ~ Jeffreys
#####
tam<-c(10,20,50,100,200)
rho=c(0.20,0.50,0.90)
r.EPRo3<-TabMtCa(tam,rho,EPRo3)
save(tam,rho,a,b,r.EPRo3,file="TabEPRo3.RData")
#load(file="TabEPRo3.RData")
r.EPL3<-TabMtCa(tam,rho,EPL3)
save(tam,rho,a,b,r.EPL3,file="TabEPL3.RData")
#load(file="TabEPL3.RData")
r.EPLq3<-TabMtCa(tam,rho,EPLq3)
save(tam,rho,a,b,r.EPLq3,file="TabEPLq3.RData")
#load(file="TabEPLq3.RData")

```

```

#####
# posterior probability and Bayes factor
#####
tam<-c(200)
rho=c(0.2,0.5,0.9)
a<-1.5
b<-2.5
c<-0.05
d<-0.95
m<-seq(0,5)
dist=3
r.predit<-matrix(nrow=length(m),ncol=(dist+3)*length(rho))
for(i in m) {
  res1<-TabMtCa(tam,rho,PP1,a,b,i)
  res2<-TabMtCa(tam,rho,PP2,c,d,i)
  res3<-TabMtCa(tam,rho,PP3,i)
  for(j in 1:length(rho)) {
    r.predit[i+1,(dist+3)*j-(dist+3-1)]<-res1[j,1]
    r.predit[i+1,(dist+3)*j-(dist+3-2)]<-res2[j,1]
    r.predit[i+1,(dist+3)*j-(dist+3-3)]<-res3[j,1]
    r.predit[i+1,(dist+3)*j-(dist+3-4)]<-res2[j,1]/res1[j,1]
    r.predit[i+1,(dist+3)*j-(dist+3-5)]<-res3[j,1]/res1[j,1]
    r.predit[i+1,(dist+3)*j-(dist+3-6)]<-res3[j,1]/res2[j,1]
  }
}
save(tam,rho,a,b,c,d,r.predit,file="Tabpredict.RData")
#load(file="Tabpredict.RData")
#####

```

Apêndice B

Listagem dos Programas em R para Filas $M/M/1/K$

MM1K_160314.R

```
#####  
# functions for M/M/1/K queue  
#####  
dmmlk<-function(n, rho, K){ #probability  
  p<-NULL  
  p<-sapply(floor(n), function(n) if(n>K) 0  
    else (1-rho)*rho^(n)/(1-rho^(K+1)))  
  return(p)}  
pmmlk<-function(n, rho, K){ #CDF  
  p<-NULL  
  p<-sapply(n, function(n) if(n>K) 1  
    else (1-rho^(n+1))/(1-rho^(K+1)))  
  return(p)}  
qmmlk<-function(p, rho, K){ # quantile  
  q<-NULL  
  q<-log(1-p*(1-rho^(K+1)))/log(rho)-1  
  return(round(q))}  
rmmlk<-function(tam, rho, K){ # simulate  
  u<-runif(tam, 0, 1)  
  x<-log(1-u*(1-rho^(K+1)))/%log(rho)  
  return(x)}  
geom.trunc <- function(tam, rho, k){  
  xi<-NULL  
  u<-runif(tam, 0, 1)  
  xi<-1+(log(1-u*(1-rho^(k+1)))/%log(rho)-1)  
  return(xi)}  
#####  
# maximum likelihood estimate for rho
```

```

#####
like.f<-function(rho,K,n,sumxi) {
  (1-rho)^n*rho^(sumxi)/(1-rho^(K+1))^n}
loglike.f<-function(rho,K,n,sumxi) {
  n*log(1-rho)+sumxi*log(rho)-n*log(1-rho^(K+1))}
EMVRoMMIK<-function(K,amostra) {
  Eps.EMV<-1e-06
  n<-length(amostra)
  sumxi<-sum(amostra)
  res<-optimize(loglike.f,c(Eps.EMV,1-Eps.EMV),K,n,sumxi,maximum=TRUE,tol=Eps.EMV)
  return(res$maximum)}
#####
# Bayesian estimate rho ~ Unif(0,1)
#####
require(HI)
EBURoMMIK<-function(K,amostra) {
  sSize<-5000
  n<-length(amostra)
  sumxi<-sum(amostra)
  res<-arms(runif(1),loglike.f,function(x,K,n,sumxi)((x>0)*(x<1)),sSize,K,n,sumxi)
  return(mean(res))}
#####
# Bayesian estimate rho ~ Beta(a,b)
#####
logpost.f<-function(rho,K,n,sumxi,a,b) {
  (n+b-1)*log(1-rho)+(sumxi+a-1)*log(rho)-n*log(1-rho^(K+1))}
EBaRoMMIK<-function(K,amostra,a,b) {
  sSize<-5000
  n<-length(amostra)
  sumxi<-sum(amostra)
  res<-arms(runif(1),logpost.f,function(x,K,n,sumxi,a,b)((x>0)*(x<1)),sSize,K,n,sumxi,a,b)
  # summary(res);mean(res);sd(res)/sqrt(n)
  # hist(res,prob=TRUE,main="fPost")
  # boxplot(res)
  return(mean(res))}
#####
# Bootstrap-corrected estimate
#####
EBoRoMMIK<-function(K,amostra) {
  B<-1000
  soma<-0
  for(i in 1:B) {
    reamostra<-sample(amostra,replace=T)
    estm<-EMVRoMMIK(K,reamostra)
    soma<-soma+estm}
  estmStar=soma/B
  return(2*EMVRoMMIK(K,amostra)-estmStar)}
#####
# monte carlo estimates
#####
MtCaRoMMIK<-function(tam,rho,K,fEst,...) {
  rep<-10000
  amostra<-numeric(tam)
  est<-numeric(rep)
  for(i in 1:rep) {
    # gerar dados

```

```

    amostra<-rmmlk(tam, rho, K)
    oldseed<-GlobalEnv$.Random.seed
    # ajustar modelo e guardar ajuste
    est[i]<-fEst(K, amostra, ...)
    GlobalEnv$.Random.seed<-oldseed
  }
  c(mean(est), sqrt(var(est)/rep))}
MtCaRoMMIKComp<-function(tam, rho, K, fEst1, fEst2, fEst3) {
  rep<-100
  amostra<-numeric(tam)
  est1<-numeric(rep)
  est2<-numeric(rep)
  est3<-numeric(rep)
  a<-1.5
  b<-2.5
  for (i in 1:rep) {
    # gerar dados
    amostra<-rmmlk(tam, rho, K)
    print(amostra)
    oldseed<-GlobalEnv$.Random.seed
    # ajustar modelo e guardar ajuste
    est1[i]<-fEst1(K, amostra)
    print(est1[i])
    est2[i]<-fEst2(K, amostra)
    print(est2[i])
    GlobalEnv$.Random.seed<-oldseed
    est3[i]<-fEst3(K, amostra, a, b)
    print(est3[i])
    GlobalEnv$.Random.seed<-oldseed
  }
  c(mean(est1), sqrt(var(est1)/rep),
    mean(est2), sqrt(var(est2)/rep),
    mean(est3), sqrt(var(est3)/rep))}
#####
# compute monte carlo tables
#####
TabMtCaRoMMIK<-function(tam, rho, K, fEst, ...) {
  res<-matrix(nrow=length(rho), ncol=2*length(tam))
  for (i in 1:length(rho)) {
    for (j in 1:length(tam)) {
      set.seed(13579)
      res[i, (2*j-1):(2*j)]<-MtCaRoMMIK(tam[j], rho[i], K, fEst, ...) }
    res}
#####
# tests for random generation
#####
tam=10
rho=0.20
K=5
n=seq(0, K)
dmmlk(n, rho, K)
pmmlk(n, rho, K)
qmmlk(pmmlk(n, rho, K), rho, K)
set.seed(13579)
rmmlk(tam, rho, K)
set.seed(13579)

```

```

geom.trunc(tam, rho, K)
#####
# tests for maximum likelihood
#####
tam<-10
rho<-0.20
K<-5
set.seed(13579)
amostra<-rmmlk(tam, rho, K)
n<-length(amostra)
sumxi<-sum(amostra)
rhos<-seq(0, 1, 0.01)
plot(rhos, like.f(rhos, K, n, sumxi), type="l")
hatrho<-EMVRoMMIK(K, amostra)
lines(hatrho, like.f(hatrho, K, n, sumxi), type="h")
plot(rhos, loglike.f(rhos, K, n, sumxi), type="l")
hatrho<-EMVRoMMIK(K, amostra)
lines(hatrho, loglike.f(hatrho, K, n, sumxi), type="h")
#####
# test for Bayesian estimates
#####
tam<-10
rho<-0.20
K<-5
a<-1.5
b<-2.5
set.seed(13579)
amostra<-rmmlk(tam, rho, K)
n<-length(amostra)
sumxi<-sum(amostra)
rhos<-seq(0, 1, 0.01)
plot(rhos, logpost.f(rhos, K, n, sumxi, a, b), type="l")
set.seed(13579)
hatrho<-EBaRoMMIK(K, amostra, a, b)
lines(hatrho, logpost.f(hatrho, K, n, sumxi, a, b), type="h")
plot(rhos, loglike.f(rhos, K, n, sumxi), type="l")
set.seed(13579)
hatrho<-EBURoMMIK(K, amostra)
lines(hatrho, loglike.f(hatrho, K, n, sumxi), type="h")
#####
# compute averages
#####
tam<-10
rho<-0.20
K<-5
a<-1.5
b<-2.5
set.seed(13579)
MtCaRoMMIK(tam, rho, K, EMVRoMMIK)
set.seed(13579)
MtCaRoMMIK(tam, rho, K, EBURoMMIK)
set.seed(13579)
MtCaRoMMIK(tam, rho, K, EBaRoMMIK, a, b)
set.seed(13579)
MtCaRoMMIK(tam, rho, K, EBoRoMMIK)
set.seed(13579)

```

```

MtCaRoMM1KComp (tam , rho , K, EMVRoMM1K, EBURoMM1K, EBaRoMM1K)
#####
# generate MLE tables
#####
tam<-c(10,20,50,100,200)
rho=c(0.01,0.10,0.20,0.50,0.90,0.99)
K<-5
r.EMVRoMM105<-TabMtCaRoMM1K(tam, rho, K, EMVRoMM1K)
save(tam, rho, K, r.EMVRoMM105, file="TabEMVRoMM105.RData")
load(file="TabEMVRoMM105.RData")
K<-20
r.EMVRoMM120<-TabMtCaRoMM1K(tam, rho, K, EMVRoMM1K)
save(tam, rho, K, r.EMVRoMM120, file="TabEMVRoMM120.RData")
load(file="TabEMVRoMM120.RData")
K<-80
r.EMVRoMM180<-TabMtCaRoMM1K(tam, rho, K, EMVRoMM1K)
save(tam, rho, K, r.EMVRoMM180, file="TabEMVRoMM180.RData")
load(file="TabEMVRoMM180.RData")
#####
# generate Bayesian tables - uniform distribution
#####
tam<-c(10,20,50,100,200)
rho=c(0.01,0.10,0.20,0.50,0.90,0.99)
K<-5
r.EBURoMM105<-TabMtCaRoMM1K(tam, rho, K, EBURoMM1K)
save(tam, rho, K, r.EBURoMM105, file="TabEBURoMM105.RData")
load(file="TabEBURoMM105.RData")
K<-20
r.EBURoMM120<-TabMtCaRoMM1K(tam, rho, K, EBURoMM1K)
save(tam, rho, K, r.EBURoMM120, file="TabEBURoMM120.RData")
load(file="TabEBURoMM120.RData")
K<-80
r.EBURoMM180<-TabMtCaRoMM1K(tam, rho, K, EBURoMM1K)
save(tam, rho, K, r.EBURoMM180, file="TabEBURoMM180.RData")
load(file="TabEBURoMM180.RData")
#####
# generate Bayesian tables - beta(a,b) distribution
#####
tam<-c(10,20,50,100,200)
rho=c(0.01,0.10,0.20,0.50,0.90,0.99)
a<-1.5
b<-2.5
K<-5
r.EBaRoMM105<-TabMtCaRoMM1K(tam, rho, K, EBaRoMM1K, a, b)
save(tam, rho, K, r.EBaRoMM105, file="TabEBaRoMM105.RData")
load(file="TabEBaRoMM105.RData")
K<-20
r.EBaRoMM120<-TabMtCaRoMM1K(tam, rho, K, EBaRoMM1K, a, b)
save(tam, rho, K, r.EBaRoMM120, file="TabEBaRoMM120.RData")
load(file="TabEBaRoMM120.RData")
K<-80
r.EBaRoMM180<-TabMtCaRoMM1K(tam, rho, K, EBaRoMM1K, a, b)
save(tam, rho, K, r.EBaRoMM180, file="TabEBaRoMM180.RData")
load(file="TabEBaRoMM180.RData")
#####
# generate bootstrap tables

```



```
#####  
tam<-c(10,20,50,100,200)  
rho=c(0.01,0.10,0.20,0.50,0.90,0.99)  
K<-5  
r.EBoRoMM105<-TabMtCaRoMM1K(tam,rho,K,EBoRoMM1K)  
save(tam,rho,K,r.EBoRoMM105,file="TabEBoRoMM105.RData")  
#load(file="TabEBoRoMM105.RData")  
K<-20  
r.EBoRoMM120<-TabMtCaRoMM1K(tam,rho,K,EBoRoMM1K)  
save(tam,rho,K,r.EBoRoMM120,file="TabEBoRoMM120.RData")  
#load(file="TabEBoRoMM120.RData")  
K<-80  
r.EBoRoMM180<-TabMtCaRoMM1K(tam,rho,K,EBoRoMM1K)  
save(tam,rho,K,r.EBoRoMM180,file="TabEBoRoMM180.RData")  
#load(file="TabEBoRoMM180.RData")  
#####  
_____
```