

UNIVERSIDADE FEDERAL DE MINAS GERAIS - UFMG
INSTITUTO DE CIÊNCIAS EXATAS - ICEX
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA - PPGEST

Rodolfo Santos Nunes Rodrigues

Influência do número de partículas na estimação de parâmetros via
máxima verossimilhança em modelos de espaço de estados

Belo Horizonte - MG
Março de 2018

Rodolfo Santos Nunes Rodrigues

Influência do número de partículas na estimação de parâmetros via
máxima verossimilhança em modelos de espaço de estados

Dissertação apresentada ao
Departamento de Estatística da
UFMG como requisito parcial para
a obtenção do título de Mestre em
Estatística.

Orientador: Prof. Dr. Luiz
Henrique Duczmal
Co-orientador: Dr. Felipe Carvalho
Álvares da Silva

Belo Horizonte - MG
Março de 2018

RESUMO

Modelos de espaço de estados são muito utilizados para modelar diversos problemas nas áreas de economia e biologia, por isso a realização de inferências, como, por exemplo, a estimação de parâmetros, para essa classe de modelos é importante. Para esses casos, os algoritmos da classe dos filtros de partículas são capazes de resolver questões relacionadas aos modelos não lineares e não gaussianos. Poyiadjis et al. (2011) propõe duas versões de algoritmos dessa classe para a estimação de parâmetros em modelos de espaço de estados. Uma versão tem complexidade computacional linear no número de partículas e a variância da estimativa cresce quadraticamente com o tempo. A outra versão tem custo computacional quadrático e a variância da estimativa linear. Com base nisso, Nemeth et al. (2016) apresenta uma nova versão, com a utilização de métodos de densidade de kernel e Rao-Blackwell, em que a variância da estimativa e a complexidade computacional são lineares. Neste trabalho, portanto, analisamos a influência do número de partículas nessa última versão e obtemos um número ideal de partículas a ser utilizado para a estimação de parâmetros em modelos de espaço de estado, como, por exemplo, autorregressivo, de volatilidade estocástica e Poisson, por fim, utilizamos ainda uma versão com filtro bootstrap para comparar com a proposta apresentada por Nemeth et al. (2016).

Palavras-chave: Modelo de Espaço de Estados, Filtro de Partículas, Estimação de Parâmetros, Número de Partículas.

ABSTRACT

State space models are widely used to model various problems in the areas of economics and biology, so inferences, such as parameter estimation, for this class of models are important. For these cases, the algorithms of class of particle filters are able to solve questions related to non-linear and non-Gaussian models. Poyiadjis et al. (2011) proposes two versions of algorithms of this class for the parameter estimation in state space models. One version has linear computational complexity in the number of particles and the variance of the estimates that increases quadratically over time. The other one has a quadratic computational cost and variance of the estimates increases linear through time. Based on this results, Nemeth et al. (2016) presents a new version, using the kernel density methods and Rao-Blackwellisation, in which the variance of estimates and the computational complexity are linear. Therefore, in this paper, we analyze the influence of the number of particles in this last version and we obtain an ideal number of particles to be used for the parameter estimation in state space models, for example, autoregressive model, stochastic volatility model, and Poisson model. Finally, we use a bootstrap filter version to compare with the model shown by Nemeth et al. (2016).

Keywords: State Space Model, Particle Filters, Parameter Estimation, Number of Particles.

Conteúdo

1	Introdução	6
2	Métodos	7
2.1	Modelos de Espaço de Estados	7
2.2	Filtro de Partículas Auxiliar	8
2.3	Estimação de Parâmetros	10
3	Resultados	15
3.1	Modelo Autorregressivo - AR	16
3.2	Volatilidade Estocástica - SV	20
3.3	Modelo Poisson	26
4	Conclusão	31
	Apêndice	34
	Referências Bibliográficas	38

1 Introdução

Os modelos de espaço de estados são uma ferramenta muito utilizada para modelar problemas de séries temporais em diversas áreas, como economia e biologia. Um exemplo simples dessa classe de modelos é trazido por Cappé et al. (2005) sobre os modelos de captura e recaptura que são usados no estudo de populações com tamanhos desconhecidos. O foco é modelar o movimento de uma população de lagartos em três zonas espacialmente conectadas, denotadas 1, 2 e 3. Para um determinado lagarto, a seqüência das zonas onde ele fica pode ser modelada como uma cadeia de Markov com matriz de transição Q . Este modelo pertence aos modelos de espaço de estados, já que, em um dado momento, a maioria dos lagartos não são observados. Para fazer a inferência na matriz Q , o experimento de captura e recaptura é executado da seguinte maneira: no momento $k = 0$, um número (aleatório) de lagartos é capturado, marcado e liberado. Essa operação é repetida k vezes com $k = 1, \dots, n$. Marcando os animais recém capturados e registrando a cada captura a posição (zona) dos animais recapturados. Portanto, o modelo consiste em uma série de eventos de captura e posições (condicionais em uma captura) de $n+1$ coortes de animais marcados em tempos $k = 0, \dots, n$. Para considerar as populações abertas, como os lagartos podem morrer ou deixar a região de observação para sempre, um quarto estado é geralmente adicionado às três zonas espaciais. É denotado por \dagger , e do ponto de vista da cadeia de Markov subjacente é um estado absorvente, enquanto que do ponto de vista dos modelos de espaço de estados está sempre oculto. As observações podem assim ser resumidas pelas séries $\{Y_{km}\}_{0 \leq k \leq n}$ de histórias de captura que indicam, para cada lagarto pelo menos uma vez capturado, sendo m o índice de lagarto, em que zona estava em cada uma das vezes em que foi capturado. Podemos, por exemplo, escrever $\{Y_{km}\}_{0 \leq k \leq n} = (0, \dots, 0, 1, 1, 2, 0, 2, 0, 0, 3, 0, 0, 0, 1, 0, \dots, 0)$, onde 0 significa que o lagarto não foi capturado naquele índice de tempo específico. Para cada uma dessas seqüências observadas tem-se uma seqüência parcialmente oculta correspondente $\{X_{km}\}_{0 \leq k \leq n}$ de localizações de lagartos, por exemplo $\{X_{km}\}_{0 \leq k \leq n} = (1, \dots, 2, 1, 1, 2, 2, 2, 3, 2, 3, 3, 2, 2, 1, \dots)$. As razões em executar experimentos de captura e recaptura são geralmente duas: primeira, pode-se inferir sobre o tamanho de toda a população com base na história de recaptura, e, segundo, as características da população podem ser estimadas a partir dos animais capturados, como as probabilidades de captura e movimentação.

Neste trabalho estamos interessados na estimação dos parâmetros da classe de modelos de espaço de estados, para isso temos que maximizar a verossimilhança das observações. No entanto, quando os modelos de espaço de estados são não lineares e/ou não gaussianos, tal verossimilhança não possui forma fechada, portanto é necessário o uso de métodos computacionais como os da classe Monte Carlo sequencial, ou filtro de partículas. Essa classe de métodos nada mais é do que variações de amostragem por importância implementadas de maneira sequencial. No entanto, no início, quando proposta, apresentava um comportamento instável que era causado pela degeneração das partículas, termo que será explicado mais adiante. Buscando solucionar tal instabilidade, Gordon et al. (1993) propõem o filtro de partículas conhecido como filtro *bootstrap*, em que foi introduzido um passo intermediário de reamostragem que seria responsável por eliminar as partículas que apresentavam o problema da degeneração. E, com isso os filtros de partículas se consolidaram como um método eficiente para a

maximização da verossimilhança das observações, ou o problema da filtragem. Mais tarde, Pitt and Shephard (1999) propõem um novo filtro chamado filtro de partículas auxiliar, em que a ordem dos passos de amostragem e reamostragem é invertida com relação aos filtros anteriores. Tal inversão é realizada com a inserção de uma variável auxiliar, por isso o nome, de maneira que a informação mais recente, ou mais atualizada, é levada em consideração no passo de reamostragem. Estes são os dois filtros utilizados neste trabalho.

O filtro de partículas auxiliar é utilizado por Poyiadjis et al. (2011) na sua proposta de um método Monte Carlo sequencial para a estimação dos parâmetros de modelos de espaço de estados, a partir da aproximação do vetor score e da matriz de informação observada em dois formatos. A primeira versão tem complexidade computacional linear no número de partículas, mas a variância da estimativa cresce quadraticamente com o tempo. Na segunda versão, a variância da estimativa passa a crescer linearmente. Mas há uma perda na eficiência computacional, com a complexidade passando a ser quadrática. A partir disso, Nemeth et al. (2016) propõem uma melhoria usando métodos de densidade de kernel e Rao-Blackwell, em que tanto a variância da estimação quanto a complexidade computacional passam a ser lineares. Uma vez obtidas as estimativas para o vetor score e a matriz de informação observada podemos utilizar o resultado para obter uma estimativa dos parâmetros através do método do gradiente. Os detalhes desta última proposta, assim como as outras ferramentas técnicas necessárias para a estimativa dos parâmetros de modelos de espaço de estados, estão detalhados na seção seguinte.

Este trabalho tem como objetivo fazer uma análise detalhada do número de partículas necessário para obter boas estimativas dos parâmetros de diferentes modelos de espaço de estados utilizando o algoritmo de Nemeth et al. (2016). Tal análise será feita em modelos tanto lineares e gaussianos quanto em não lineares e/ou não gaussianos, como os casos autorregressivo, volatilidade estocástica e Poisson. Também propomos uma nova versão em que o filtro utilizado é o filtro *bootstrap* e fizemos a mesma análise, ou estudo comparativo, do número de partículas ideal. É esperado, e pode ser verificado numericamente na seção resultados, que o número de partículas ideal para obter boas estimativas nos diferentes modelos seja um número não muito alto de partículas e que possamos notar que o uso de um número alto de partículas apenas demanda mais tempo de execução, não trazendo grande melhoria nas estimativas quando comparado com o uso de um número baixo ou médio de partículas. Os resultados desse estudo comparativo são apresentados na terceira seção.

2 Métodos

2.1 Modelos de Espaço de Estados

Um modelo de espaço de estados é basicamente composto por dois processos, $\{X_t\}$ e $\{Y_t\}$, em que o primeiro é latente, ou seja, não observado diretamente. E o segundo processo é observado. É a partir do processo observável que podemos fazer inferência sobre o processo latente. Formalmente, temos

Definição 1. Sejam $\{X_t, 1 \leq t \leq T\}$ um processo de Markov latente que assume valores em $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ e um segundo processo observável $\{Y_t, 1 \leq t \leq T\}$ com valores em $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$. O processo $\{X_t\}$ é definido por uma densidade inicial dada por (note que: o último termo é apenas uma questão de notação para ser utilizado nas próximas equações)

$$p(x_1|\theta) = \mu_\theta(x_1) = f_\theta(x_1|x_0)$$

e densidade de transição

$$p(x_t|x_{1:t-1}, \theta) = p(x_t|x_{t-1}, \theta) = f_\theta(x_t|x_{t-1}) \quad (1)$$

em que θ é o vetor de parâmetros e $x_{1:t-1}$ denota toda a sequência até o tempo $t-1$, isto é, $(x_1, x_2, \dots, x_{t-2}, x_{t-1})$. Além disso $\{X_t\}$ é não observado diretamente mas apenas parcialmente, através de $\{Y_t\}$.

As observações $\{Y_t\}$ são condicionalmente independentes dado $\{X_t\}$ e têm densidade de probabilidade dada por

$$p(y_t|y_{1:t-1}, x_{1:t}, \theta) = p(y_t|x_t, \theta) = g_\theta(y_t|x_t) \quad (2)$$

Para fazer inferência sobre o processo latente $\{X_t\}$, precisamos da distribuição condicional dos estados latentes dadas as observações, $y_{1:T}, T \geq 1$. Considerando que o vetor de parâmetros θ é conhecido, então essa é dada por

$$p(x_{1:T}|y_{1:T}, \theta) \propto p(x_{1:T}, y_{1:T}, \theta)$$

Enquanto que a distribuição conjunta pode ser calculada por

$$\begin{aligned} p(x_{1:T}, y_{1:T}, \theta) &= p(y_{1:T}|x_{1:T}, \theta)p(x_{1:T}|\theta) \\ &= \prod_{t=1}^T p(y_t|x_t, \theta) \prod_{t=1}^T p(x_t|x_{t-1}, \theta) \\ &= \prod_{t=1}^T g_\theta(y_t|x_t) \prod_{t=1}^T f_\theta(x_t|x_{t-1}) \end{aligned} \quad (3)$$

Mesmo com a distribuição conjunta apresentada acima, em modelos de espaço de estados não gaussianos e/ou não lineares não existe forma fechada para a verossimilhança das observações, não sendo possível estimar os estados latentes de forma analítica. Uma solução para o problema é utilizar o algoritmo de Monte Carlo sequencial. Neste trabalho, utilizaremos um caso particular denominado filtro de partículas auxiliar, descrito na próxima subseção, para aproximar tais densidades e assim poderemos fazer inferência sobre os estados latentes.

2.2 Filtro de Partículas Auxiliar

Um dos objetivos dos modelos de espaço de estados é fazer inferência sobre os estados não observáveis, X_t , condicionado às observações, Y_t , isto é, um problema de filtragem. Nos problemas em que o modelo de espaço de estados é não gaussiano

e/ou não linear, não há forma fechada para o cálculo da densidade e, portanto, não é possível observar os estados latentes, X_t .

Para a solução do problema da filtragem, podem ser utilizados filtros de partículas, que pertencem à classe de algoritmos de Monte Carlo sequencial. Neste trabalho será utilizado um filtro de partículas auxiliar. É importante salientar que esse filtro foi escolhido por apresentar uma aproximação que se adapta melhor às observações, principalmente na presença de *outliers* na série observada (Pitt and Shephard (1999)).

Para executar esse filtro, temos que aproximar a densidade condicional $p(x_t|y_{1:t}, \theta)$, considerando θ fixo, ou seja, que os parâmetros são conhecidos e onde $y_{1:t}$ é a sequência das observações. Aproximação é baseada em um conjunto de N partículas, isto é, cada densidade condicional será aproximada por um conjunto de N amostras aleatórias ponderadas e temos

$$\hat{p}(dx_t|y_{1:t}, \theta) = \sum_{i=1}^N w_t^{(i)} \delta_{X_t^{(i)}}(dx_t) \quad (4)$$

onde $w_t^{(i)}$ são os pesos, que devem ser não negativos para todo i e sua soma igual a 1, e o termo $\delta_{X_t^{(i)}}(dx_t)$ é a função delta de Dirac com massa no ponto $X_t^{(i)}$.

Com esta aproximação, conseguimos gerar um conjunto de partículas $\{X_t^{(i)}\}_{i=1}^N$ e seus respectivos pesos $\{w_t^{(i)}\}_{i=1}^N$ que aproximam a função de densidade $p(x_t|y_{1:t}, \theta)$, e possui melhores resultados quanto maior for N (Crisan and Doucet (2002)).

Procedendo recursivamente e usando (1) e (2), podemos escrever a seguinte equação de filtragem

$$\begin{aligned} p(x_t|y_{1:t}, \theta) &= p(x_t|y_t, y_{1:t-1}, \theta) \\ &= \frac{p(x_t, y_t|y_{1:t-1}, \theta)}{p(y_t|y_{1:t-1}, \theta)} \\ &\propto p(y_t|x_t, y_{1:t-1}, \theta)p(x_t|y_{1:t-1}, \theta) \\ &\propto p(y_t|x_t, \theta)p(x_t|y_{1:t-1}, \theta) \\ &\propto g_\theta(y_t|x_t)p(x_t|y_{1:t-1}, \theta) \\ &\propto g_\theta(y_t|x_t) \int p(x_t, x_{t-1}|y_{1:t-1}, \theta)dx_{t-1} \\ &\propto g_\theta(y_t|x_t) \int p(x_t|x_{t-1}, y_{1:t-1}, \theta)p(x_{t-1}|y_{1:t-1}, \theta)dx_{t-1} \\ &\propto g_\theta(y_t|x_t) \int f_\theta(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1}, \theta)dx_{t-1} \end{aligned} \quad (5)$$

A proporcionalidade pode ser explicada pelo fato da densidade $p(y_t|y_{1:t-1}, \theta)$ do denominador não depender de x_t e tampouco de θ . Considerando que podemos fazer isso para cada tempo, podemos assumir que para um tempo $t - 1$ temos o conjunto de partículas $\{X_{t-1}^{(i)}\}_{i=1}^N$ e seus respectivos pesos $\{w_{t-1}^{(i)}\}_{i=1}^N$, que aproximam a densidade $p(x_{t-1}|y_{1:t-1}, \theta)$. Portanto, usando (4) temos a aproximação de Monte Carlo para (5), dada por

$$\begin{aligned}
p(x_t|y_{1:t}, \theta) &\propto g_\theta(y_t|x_t) \int f_\theta(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1}, \theta)dx_{t-1} \\
&\approx g_\theta(y_t|x_t) \int f_\theta(x_t|x_{t-1}) \sum_{i=1}^N w_{t-1}^{(i)} \delta_{X_{t-1}^{(i)}}(x_{t-1})dx_{t-1} \\
&\approx g_\theta(y_t|x_t) \sum_{i=1}^N w_{t-1}^{(i)} \int f_\theta(x_t|x_{t-1})\delta_{X_{t-1}^{(i)}}(x_{t-1})dx_{t-1} \\
&\approx cg_\theta(y_t|x_t) \sum_{i=1}^N w_{t-1}^{(i)} f_\theta(x_t|x_{t-1}^{(i)})
\end{aligned} \tag{6}$$

onde c é uma constante normalizadora.

Agora, podemos calcular a densidade dos estados latentes condicionado às observações usando a aproximação acima através de um filtro de partículas auxiliar proposto por Pitt and Shephard (1999). A ideia é aproximar $cw_{t-1}^{(i)}g_\theta(y_t|x_t)f_\theta(x_t|x_{t-1}^{(i)})$ por $\xi_t^{(i)}q(x_t|x_{t-1}^{(i)}, y_t, \theta)$, onde $\{\xi_t^{(i)}\}_{i=1}^N$ é um conjunto de probabilidades e $q(x_t|x_{t-1}^{(i)}, y_t, \theta)$ é uma densidade de proposta. Nesse sentido, uma partícula no tempo t é calculada com o auxílio do algoritmo a partir de uma partícula do tempo $t - 1$, $x_{t-1}^{(i)}$, escolhida com probabilidade $\xi_t^{(i)}$, depois propagamos essa partícula no tempo t gerando x_t de $q(x_t|x_{t-1}^{(i)}, y_t, \theta)$ com novo peso igual a $cw_{t-1}^{(i)}g_\theta(y_t|x_t)f_\theta(x_t|x_{t-1}^{(i)})/\xi_t^{(i)}q(x_t|x_{t-1}^{(i)}, y_t, \theta)$ como detalhado no Algoritmo 1.

Algoritmo 1: FILTRO DE PARTÍCULAS AUXILIAR

Entrada: observações $(y_{1:T})$, conjunto inicial de partículas, vetor de parâmetros

Saída: Aproximação para os estados latentes

1 **início**

2 **para** cada observação faça

3 Amostre os índices k_i com probabilidades $\xi_t^{(i)}$

4 Gere o conjunto de partículas para o tempo t a partir de $q(x_t|x_{t-1}^{(k_i)}, y_t, \theta)$

5 Atualize os pesos $w_t^{(i)}$

6 **fim**

7 **fim**

8 **retorna** Dados filtrados com a aproximação para os estados latentes

2.3 Estimação de Parâmetros

Como já mencionado, o objetivo deste trabalho é estimar parâmetros dos modelos de espaço de estados utilizando o algoritmo de filtro de partículas auxiliar, descrito na seção anterior. Para estimar o vetor de parâmetros, θ , devemos resolver

$$\hat{\theta} = \arg \max_{\theta \in \Theta} \log p(y_{1:T}|\theta) = \arg \max_{\theta \in \Theta} \sum_{t=1}^T \log p(y_t|y_{1:t-1}, \theta)$$

onde

$$p(y_t|y_{1:t-1}, \theta) = \int \left(g_\theta(y_t|x_t) \int f_\theta(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1}, \theta)dx_{t-1} \right) dx_t$$

Contudo, para a maior parte dos casos, não há forma fechada para a log-verossimilhança, por isso a aproximação via Monte Carlo sequencial, por exemplo, torna-se necessária. Uma maneira de solucionar esse problema é fixando o valor do parâmetro e então calcular a log-verossimilhança para saber qual θ é o máximo da função, mas isso é simples apenas em casos discretos e de dimensão baixa. Nos problemas em que o espaço paramétrico é não enumerável teríamos que fazer um grid dos valores de θ e calcular a log-verossimilhança para cada valor do grid, mas isso se torna computacionalmente ineficiente com o aumento da dimensão de θ .

Uma solução eficiente comparada com as anteriores é usar o método do gradiente (Nemeth et al. (2016)), no qual o parâmetro estimado é atualizado a partir da direção do vetor gradiente, sendo necessário apenas um θ inicial. Recursivamente, podemos escrever:

$$\theta_k = \theta_{k-1} + \gamma_k \nabla \log p(y_{1:T}|\theta)|_{\theta=\theta_{k-1}} \quad (7)$$

em que γ_k é um sequência de passos que satisfaz $\sum_k \gamma_k = \infty$ e $\sum_k \gamma_k^2 < \infty$. Essas condições são necessárias para a convergência de θ para um valor $\hat{\theta}$ em que $\nabla \log p(y_{1:T}|\hat{\theta}) = 0$, de tal forma que a estimativa de θ é consistente. Um valor muito utilizado para γ_k é $k^{-\alpha}$, com $\alpha \in (0.5; 1)$ (Nemeth et al. (2016)).

A partir disso, nosso foco para aproximar a log-verossimilhança passa a ser agora em encontrar uma aproximação do seu gradiente, isto é, o vetor score. Já que em modelos de espaço de estados não gaussianos e/ou não lineares não é possível calcular o gradiente analiticamente.

A partir da identidade de Fisher (Cappé et al. (2005)), podemos escrever o gradiente como

$$\nabla \log p(y_{1:T}|\theta) = \int \nabla \log p(x_{1:T}, y_{1:T}|\theta)p(x_{1:T}|y_{1:T}, \theta)dx_{1:T} \quad (8)$$

por meio do filtro de partículas auxiliar, descrito na seção anterior, obtemos uma aproximação para $p(x_{1:T}|y_{1:T}, \theta)$. Assumindo que as densidades condicionais de x_t e y_t respectivamente dadas por (1) e (2) são duas vezes contínuas e diferenciáveis, então a partir da densidade conjunta dada por (3) podemos escrever

$$\nabla \log p(x_{1:T}, y_{1:T}|\theta) = \sum_{t=1}^T [\nabla \log g_\theta(y_t|x_t) + \nabla \log f_\theta(x_t|x_{t-1})] \quad (9)$$

assim, temos uma maneira de calcular o gradiente acima que em conjunto com a aproximação para os estados latentes, obtida via filtro de partículas, serão utilizadas para obtermos o vetor score.

A partir do filtro auxiliar podemos gerar $x_t^{(i)}$ e assim todo o caminho da partícula, $x_{1:t}^{(i)}$ e seja $\alpha_t^{(i)} := \nabla \log p(x_{1:t}^{(i)}, y_{1:t}|\theta)$ o valor do gradiente dado por (9) para a partícula i no tempo t , que depende de todo o caminho da partícula. Considerando que no tempo t

temos as partículas $\{X_t^{(i)}\}_{i=1}^N$ e seus respectivos pesos $\{w_t^{(i)}\}_{i=1}^N$ e índices $\{k_i\}_{i=1}^N$, a estimativa para $\alpha_t^{(i)}$ pode ser atualizada recursivamente e (9) pode ser escrita da seguinte maneira:

$$\alpha_t^{(i)} = \alpha_{t-1}^{(k_i)} + \nabla \log g_\theta(y_t | x_t^{(i)}) + \nabla \log f_\theta(x_t^{(i)} | x_{t-1}^{(k_i)}) \quad (10)$$

a partir de (8), temos a aproximação do vetor score no tempo t

$$\nabla \log p(y_{1:t} | \theta) = S_t \approx \sum_{i=1}^N w_t^{(i)} \alpha_t^{(i)} \quad (11)$$

Da forma como estamos calculando, cada $\alpha_t^{(i)}$ depende de todo o caminho da partícula, ainda que não é necessário armazenar todo o caminho do processo latente. No entanto, isso pode gerar degeneração das partículas ao longo do tempo. A degeneração das partículas é o fenômeno no qual ocorre um desequilíbrio nos pesos de importância de cada estado a medida que as iterações são executadas, isso faz com que o número de partículas com peso insignificante seja cada vez maior e assim aumentando a variabilidade da estimativa. E além disso, a forma como estamos calculando também pode fazer com que a variância da estimativa do vetor score cresce pelo menos quadraticamente com o tempo (Poyiadjis et al. (2011)). Uma solução para esse problema é utilizar o método de densidade de kernel proposto por Liu and West (2001).

A proposta aplicada aos $\alpha_t^{(i)}$'s visa o encolhimento dos $\alpha_t^{(i)}$'s para a sua média e adicionamos um ruído para evitar os casos de degeneração. Por isso $\alpha_{t-1}^{(k_i)}$ passa a ser calculada como um núcleo gaussiano em que k_i é amostrada a partir de uma distribuição discreta com probabilidades $\xi_t^{(i)}$, com média e variância para $\alpha_{t-1}^{(k_i)}$ dadas por

$$S_{t-1} = \sum_{i=1}^N w_{t-1}^{(i)} \alpha_{t-1}^{(i)} \quad e \quad \Sigma_{t-1}^\alpha = \sum_{i=1}^N w_{t-1}^{(i)} (\alpha_{t-1}^{(i)} - S_{t-1})^T (\alpha_{t-1}^{(i)} - S_{t-1})$$

Seja $0 < \lambda < 1$ o parâmetro de encolhimento, constante fixa. Escolhendo uma largura de densidade $h > 0$, podemos escrever o termo $\alpha_{t-1}^{(k_i)}$ na equação (10) como

$$\lambda \alpha_{t-1}^{(k_i)} + (1 - \lambda) S_{t-1} + \epsilon_t^{(i)} \quad (12)$$

em que $\epsilon_t^{(i)}$ é uma realização de uma normal $N(0, h^2 \Sigma_{t-1}^\alpha)$. Essa aproximação via densidade de kernel preserva a média e a variância dos $\alpha_t^{(i)}$'s, se forem escolhidos λ e h tais que $\lambda^2 + h^2 = 1$ (Liu and West (2001)). No entanto o uso da adição de um ruído resulta em uma atualização linear gaussiana dos $\alpha_t^{(i)}$'s, o que possibilita o uso da ideia de Rao-Blackwell (Doucet et al. (2000)) para reduzir a variância da estimação do vetor score. Usando essa técnica podemos atualizar os $\alpha_t^{(i)}$'s de forma sequencial com uma distribuição apropriada e sem adicionar ruído.

Considerando que, para $t \geq 2$, no tempo $t - 1$ cada $\alpha_{t-1}^{(j)}$ é representada por uma distribuição Normal de forma

$$\alpha_{t-1}^{(j)} \sim N(m_{t-1}^{(j)}, h^2 V_{t-1})$$

Portanto, de (10) e (12), temos para o tempo t que

$$\alpha_t^{(i)} \sim N(m_t^{(i)}, h^2 V_t) \quad (13)$$

em que

$$m_t^{(i)} = \lambda m_{t-1}^{(k_i)} + (1 - \lambda) S_{t-1} + \nabla \log g_\theta(y_t | x_t^{(i)}) + \nabla \log f_\theta(x_t^{(i)} | x_{t-1}^{(k_i)})$$

e

$$V_t = V_{t-1} + \Sigma_{t-1}^\alpha = V_{t-1} + \sum_{i=1}^N w_{t-1}^{(i)} (m_{t-1}^{(i)} - S_{t-1})^T (m_{t-1}^{(i)} - S_{t-1})$$

Portanto, a estimação do vetor score em cada iteração é a média ponderada dos $\alpha_t^{(i)}$'s no instante t , dada por

$$S_t = \sum_{i=1}^N w_t^{(i)} m_t^{(i)} \quad (14)$$

Utilizando a proposta de Nemeth et al. (2016) temos o Algoritmo 2 para o cálculo do vetor score que usa o Algoritmo 1 como auxiliar. No Algoritmo 2, a variância da estimação do vetor score passa a crescer linearmente com o tempo e tem tempo computacional da ordem de $O(N)$, em que N é o número de partículas. Além disso, utilizando esse algoritmo podemos voltar à proposta de Poyiadjis et al. (2011) apenas colocando $\lambda = 1$.

Algoritmo 2: VETOR SCORE

Entrada: observações $(y_{1:T})$, conjunto inicial de partículas, vetor de parâmetros, valor lambda (λ)

Saída: Aproximação para o vetor score, $S_{1:T}$

1 **início**

2 **para** cada observação faça

3 Aplicar o Algoritmo 1 para obter os estados latentes $\{x_{1:T}^{(i)}\}_{i=1}^N$

4 Calcular a média $(m_t^{(i)})$ para os α_t 's usando

5 $m_t^{(i)} = \lambda m_{t-1}^{(k_i)} + (1 - \lambda) S_{t-1} + \nabla \log g_\theta(y_t | x_t^{(i)}) + \nabla \log f_\theta(x_t^{(i)} | x_{t-1}^{(k_i)})$

6 Calcular o vetor score como $S_t = \sum_{i=1}^N w_t^{(i)} m_t^{(i)}$

7 **fim**

8 **fim**

9 **retorna** Aproximação para o vetor score, $S_{1:T}$

Se além de calcular o vetor score, conseguimos calcular a matriz de informação observada, então a convergência da estimação dos parâmetros no método do gradiente será melhor. Para esse cálculo, podemos escrever como o método de Newton-Raphson e a equação (7) passa a ser escrita com o termo de segunda ordem, isto é,

$$\theta_k = \theta_{k-1} - \gamma_k (\nabla^2 \log p(y_{1:T} | \theta))^{-1} \nabla \log p(y_{1:T} | \theta) |_{\theta=\theta_{k-1}} \quad (15)$$

No entanto, a inversa da matriz de informação observada pode ser uma fonte de instabilidade numérica, uma vez que nem sempre é possível calculá-la. Procedendo de forma análoga ao caso do vetor score, temos uma identidade para a matriz de informação observada dada por Louis (1982)

$$-\nabla^2 \log p(y_{1:T}|\theta) = \nabla \log p(y_{1:T}|\theta) \nabla \log p(y_{1:T}|\theta)^T - \frac{\nabla^2 p(y_{1:T}|\theta)}{p(y_{1:T}|\theta)} \quad (16)$$

onde

$$\begin{aligned} \frac{\nabla^2 p(y_{1:T}|\theta)}{p(y_{1:T}|\theta)} &= \int \nabla \log p(x_{1:T}, y_{1:T}|\theta) \nabla \log p(x_{1:T}, y_{1:T}|\theta)^T p(x_{1:T}|y_{1:T}, \theta) dx_{1:T} \\ &+ \int \nabla^2 \log p(x_{1:T}, y_{1:T}|\theta) p(x_{1:T}, y_{1:T}|\theta) dx_{1:T} \end{aligned}$$

Assim como em (8) também é possível calcular (16) a partir do filtro de partículas auxiliar. Derivando (9), podemos escrever

$$\nabla^2 \log p(x_{1:T}, y_{1:T}|\theta) = \sum_{t=1}^T \nabla^2 \log g_\theta(y_t|x_t) + \nabla^2 \log f_\theta(x_t|x_{t-1}) \quad (17)$$

Portanto, a aproximação por partículas para a matriz de informação observada I_t no tempo t é dada por

$$I_t \approx S_t S_t^T - \sum_{i=1}^N w_t^{(i)} \left\{ \alpha_t^{(i)} \left(\alpha_t^{(i)} \right)^T + \beta_t^{(i)} \right\} \quad (18)$$

onde β_t é definido de forma similiar a α_t como $\beta_t^{(i)} := \nabla^2 \log p(x_{1:t}, y_{1:t}|\theta)$. De forma análoga ao que foi feito anteriormente para α_t , podemos utilizar o método de densidade de kernel e aplicar a ideia de Rao-Blackwell para β_t , assim obtendo a estimação da matriz de informação observada em cada tempo t como

$$I_t = S_t S_t^T - \sum_{i=1}^N w_t^{(i)} \left\{ m_t^{(i)} \left(m_t^{(i)} \right)^T + h^2 V_t + n_t^{(i)} \right\} \quad (19)$$

onde $m_t^{(i)}$ é a média da distribuição normal de $\alpha_t^{(i)}$ como mostrado anteriormente e de forma análoga $n_t^{(i)}$ é a média da distribuição normal de $\beta_t^{(i)}$ dada por

$$n_t^{(i)} = \lambda n_{t-1}^{(k_i)} + (1 - \lambda) B_{t-1} + \nabla^2 \log g_\theta(y_t|x_t^{(i)}) + \nabla^2 \log f_\theta(x_t^{(i)}|x_{t-1}^{(k_i)})$$

com $B_t = \sum_{i=1}^N w_t^{(i)} n_t^{(i)}$. Veja que o termo $h^2 V_t$ na equação (19) é importante para corrigir o decréscimo da variabilidade devido ao encolhimento de α_t para S_t em cada iteração.

Finalmente, temos a versão do Algoritmo 2 com o uso da segunda ordem, ou seja, com o uso da matriz de informação observada apresentada no Algoritmo 3.

Também propomos a utilização de uma versão análoga ao Algoritmo 2 e 3, mas com o uso do filtro *bootstrap* para obter os estados latentes, ao invés do filtro de partículas auxiliar do Algoritmo 1.

Algoritmo 3: MATRIZ DE INFORMAÇÃO OBSERVADA

Entrada: observações $(y_{1:N})$, conjunto inicial de partículas, vetor de parâmetros, valor lambda (λ)

Saída: Aproximação para a matriz de informação observada

1 **início**

2 **para** cada observação faça

3 Aplicar o Algoritmo 1 para obter os estados latentes

4 Calcular a média $(m_t^{(i)})$ e $(n_t^{(i)})$ para os α_t 's e β_t 's, respectivamente, usando:

5 $m_t^{(i)} = \lambda m_{t-1}^{(k_i)} + (1 - \lambda) S_{t-1} + \nabla \log g_\theta(y_t | x_t^{(i)}) + \nabla \log f_\theta(x_t^{(i)} | x_{t-1}^{(k_i)})$

6 $n_t^{(i)} = \lambda n_{t-1}^{(k_i)} + (1 - \lambda) B_{t-1} + \nabla^2 \log g_\theta(y_t | x_t^{(i)}) + \nabla^2 \log f_\theta(x_t^{(i)} | x_{t-1}^{(k_i)})$

7 Calcular o vetor score como $S_t = \sum_{i=1}^N w_t^{(i)} m_t^{(i)}$

8 Calcular a matriz de informação observada como:

9 $I_t = S_t S_t^T - \sum_{i=1}^N w_t^{(i)} \left\{ m_t^{(i)} \left(m_t^{(i)} \right)^T + n_t^{(i)} \right\} - h^2 V_t$

10 **fim**

11 **fim**

12 **retorna** Aproximação para a matriz de informação observada

3 Resultados

Nesta seção, apresentaremos os resultados obtidos com intuito de avaliar o método para estimação dos parâmetros de modelos de espaço de estados apresentado anteriormente. Utilizamos a versão com a matriz de informação observada, isto é, o Algoritmo 3 e a equação para estimação dos parâmetros com o termo de segunda ordem dada por (15). O valor do parâmetro de encolhimento, λ , descrito em (12) e usado no Algoritmo 3 foi escolhido como $\lambda = 0.95$. Além disso, o tamanho do passo γ_k das equações (7) e (15) foi utilizado tal como sugerido com $\alpha = 2/3$, isto é, $\gamma_k = k^{-2/3}$. Consideramos que as estimativas dos parâmetros correspondem às estimativas encontradas quando o algoritmo cumprir o critério de convergência estabelecido por

$$\frac{\|\theta_k - \theta_{k-1}\|}{\|\theta_k\|} \leq 10^{-4} \quad (20)$$

onde $\|\cdot\|$ representa a norma do vetor de parâmetros. Esse critério de convergência foi estabelecido para que não seja necessário preestabelecer um número alto de iterações que iriam ser executadas sem grande ganho nas estimativas, isto é, apenas na quinta casa decimal, aumentando o tempo de execução do algoritmo.

Além da avaliação do método, estamos interessados em obter um número de partículas ideal para estimar os parâmetros de cada modelo. Nesse sentido, foi desenhado um estudo de simulação com $T = 1000$ observações de cada modelo e variações do número de partículas em 10 cenários (25; 100; 250; 500; 750; 1000; 1250; 1500; 1750; 2000) com 100 replicações cada. Os cenários foram escolhidos com um intervalo de 250 partículas entre cada a partir de 250 até 2000, além de dois cenários com número baixo de partículas, um com 25 e outro com 100.

Foram escolhidos três diferentes modelos com o intuito de abordar as diferentes

variações de um modelo de espaço de estados entre linearidade e gaussianidade. O primeiro compreende o modelo autorregressivo que é linear e gaussiano, nos serve como ilustração, já que é possível obter resultados analíticos (Durbin and Koopman (2012)). Depois, um modelo um pouco mais complexo onde a linearidade é perdida, o modelo de volatilidade estocástica, muito utilizado na área de economia. Por fim, o modelo Poisson que é não gaussiano e também não linear, além de apresentar observações discretas. Também fizemos o estudo de simulação para os mesmos três modelos utilizando o filtro *bootstrap* com o intuito de comparar com a versão que utiliza o filtro auxiliar apresentada anteriormente. A análise dos resultados das simulações de cada modelo é detalhada abaixo; todos os algoritmos foram implementados na linguagem Ox, que é uma linguagem de programação com biblioteca de funções estatísticas e matemáticas proposta por Jurgen A Doornik, sendo uma alternativa mais rápida com relação ao R, porém não tão rápida quanto à linguagem C. Todas as simulações foram executadas e os resultados aqui apresentados produzidos em máquinas com processador Intel Core i5 - 3330 CPU 3.00 GHz e memória total de 3.92 *gigabytes* com sistema operacional Linux-Kubuntu.

3.1 Modelo Autorregressivo - AR

O modelo autorregressivo de primeira ordem, AR(1), observado com ruído gaussiano é definido como

$$\begin{aligned} X_t &= \phi X_{t-1} + U_t \\ Y_t &= X_t + V_t \end{aligned} \quad (21)$$

para $t = 1, 2, 3, \dots, T$; $U_t \sim \mathcal{N}(0, \sigma^2)$ e $V_t \sim \mathcal{N}(0, \tau^2)$, tal que $U_t \perp V_t \perp X_1$, onde $X_1 \sim \mathcal{N}(0, \sigma^2/(1 - \phi^2))$.

Dessa forma, as equações para os estados latentes e observados, respectivamente, são dadas por

$$\begin{aligned} X_t | X_{t-1} = x_{t-1} &\sim \mathcal{N}(\phi x_{t-1}, \sigma^2) \\ Y_t | X_t = x_t &\sim \mathcal{N}(x_t, \tau^2) \end{aligned} \quad (22)$$

Devido ao fato do modelo autorregressivo ser linear e gaussiano, é possível obter uma densidade de proposta ótima no sentido definido em Álvares da Silva (2016) $q(x_t | x_{t-1}^{(i)}, y_t, \theta)$, isto é,

$$q(x_t | x_{t-1}^{(i)}, y_t, \theta) \propto g_\theta(y_t | x_t) f_\theta(x_t | x_{t-1}^{(i)})$$

tal que como definido em (1) e (2) e, utilizando (22),

$$\begin{aligned} g_\theta &\sim \mathcal{N}(x_t, \tau^2) \\ f_\theta &\sim \mathcal{N}(\phi x_{t-1}, \sigma^2) \end{aligned}$$

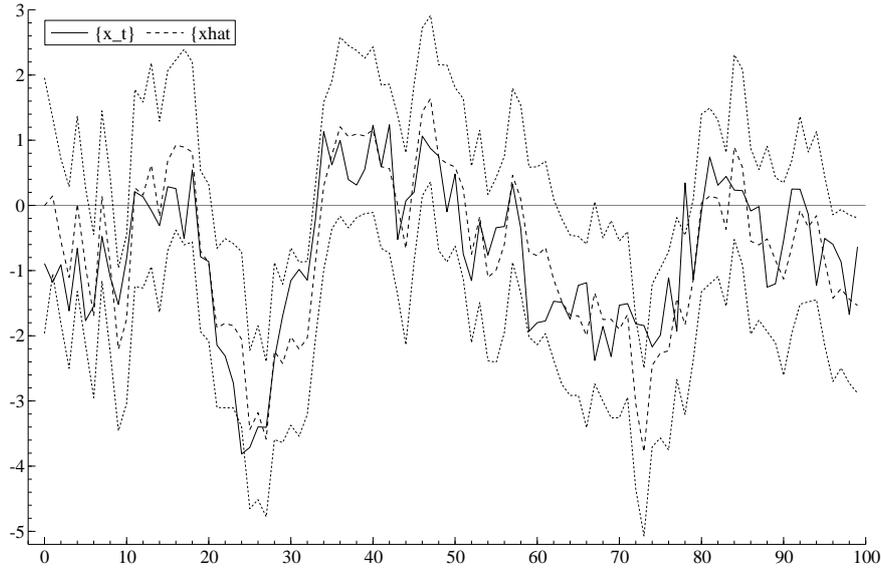


Figura 1: Estados latentes via Filtro de Partículas Auxiliar - Modelo Autorregressivo. Parâmetros reais: $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. Linha mais escura: observações. Linhas mais claras: estados latentes estimados e intervalo de confiança (95%)

Portanto,

$$\begin{aligned}
 q(x_t | x_{t-1}^{(i)}, y_t, \theta) &\propto \exp \left\{ -\frac{(y_t - x_t)^2}{2\tau^2} - \frac{(x_t - \phi x_{t-1}^{(i)})^2}{2\sigma^2} \right\} \\
 &\propto \exp \left\{ \frac{-y_t^2 + 2y_t x_t - x_t^2}{2\tau^2} + \frac{-x_t^2 + 2\phi x_t x_{t-1}^{(i)} - \phi^2 x_{t-1}^{(i)2}}{2\sigma^2} \right\} \\
 &\propto \exp \left\{ -\frac{\tau^2 + \sigma^2}{2\tau^2 \sigma^2} \left(x_t^2 - 2x_t \frac{\tau^2 \phi x_{t-1}^{(i)} + \sigma^2 y_t}{\tau^2 + \sigma^2} \right) \right\}
 \end{aligned}$$

que é o núcleo de uma distribuição normal, então

$$q_\theta \sim \mathcal{N} \left(\frac{\tau^2 \phi x_{t-1}^{(i)} + \sigma^2 y_t}{\tau^2 + \sigma^2}, \frac{\tau^2 \sigma^2}{\tau^2 + \sigma^2} \right)$$

A título de ilustração do filtro de partículas auxiliar, implementamos o Algoritmo 1 para estimar os estados latentes; o resultado pode ser observado na Figura 1

A amostra, apresentada na Figura 1, tem $T = 100$ observações. A estimativa do filtro de partículas auxiliar representada pela linha mais clara segue o comportamento dos estados latentes simulados, representados pela linha mais escura, e exceto em raros casos (menos de 5%) está fora do intervalo de confiança representado pelas linhas pontilhadas externas. Isso mostra que um filtro de partículas auxiliar estima bem os estados latentes deste modelo autorregressivo.

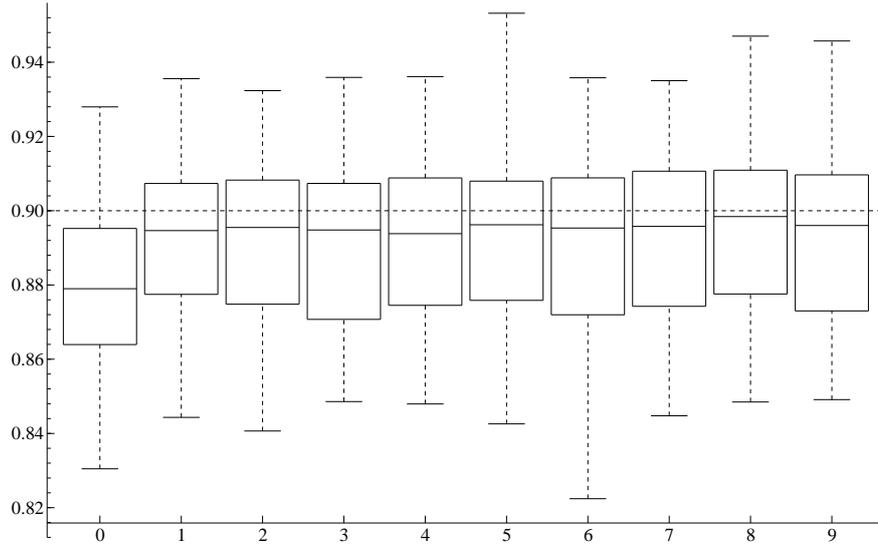


Figura 2: Estimativas para ϕ - Modelo AR. Filtro auxiliar. Parâmetros reais: $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. Linha pontilhada: valor real do parâmetro ϕ .

Uma vez que temos os estados latentes estimados pelo filtro auxiliar através do Algoritmo 1, podemos aplicar o Algoritmo 3 para obter as estimativas do vetor score e da matriz de informação observada.

Com o intuito de obter um número ideal de partículas para estimar os parâmetros de um modelo autorregressivo foi desenhado o estudo de simulação descrito anteriormente com o modelo autorregressivo definido acima com $T = 1000$ observações e parâmetros dados por $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. As estimativas obtidas via filtro auxiliar estão apresentadas na Figura 2 para o parâmetro ϕ , na Figura 3 para σ e na Figura 4 para τ .

Em cada um dos box-plots das Figuras 2, 3 e 4, a linha pontilhada representa o valor real do parâmetro simulado. Então, podemos observar que com exceção do primeiro cenário referente a 25 partículas, que é um caso apenas para ilustração com um número muito pequeno de partículas, em todos os cenários obtiveram boas estimativas para os três parâmetros, sem apresentar uma grande diferença entre os cenários. Com isso, podemos dizer que o segundo cenário (box-plot 1 nas Figuras 2, 3 e 4) que é referente a 100 partículas já é suficiente para uma boa estimativa dos parâmetros de um modelo autorregressivo. Além disso, quando levamos em consideração o tempo médio gasto em cada cenário, na Figura 5, o cenário com apenas 100 partículas converge em menos de um minuto (linha pontilhada representa tempo médio igual a 60 segundos). Portanto, para o modelo autorregressivo, com 100 partículas já se obtém bons resultados e de forma rápida, não sendo necessário por exemplo o uso de 2000 partículas, que apenas aumentariam o tempo, não trazendo grande melhoria nas estimativas dos parâmetros.

Como comparação, obtemos, a partir do mesmo estudo de simulação, as estimativas via filtro *bootstrap* para os parâmetros ϕ , σ e τ apresentadas nas Figuras 6, 7 e 8 respectivamente. Podemos observar que o uso do filtro *bootstrap* melhorou as estimativas de

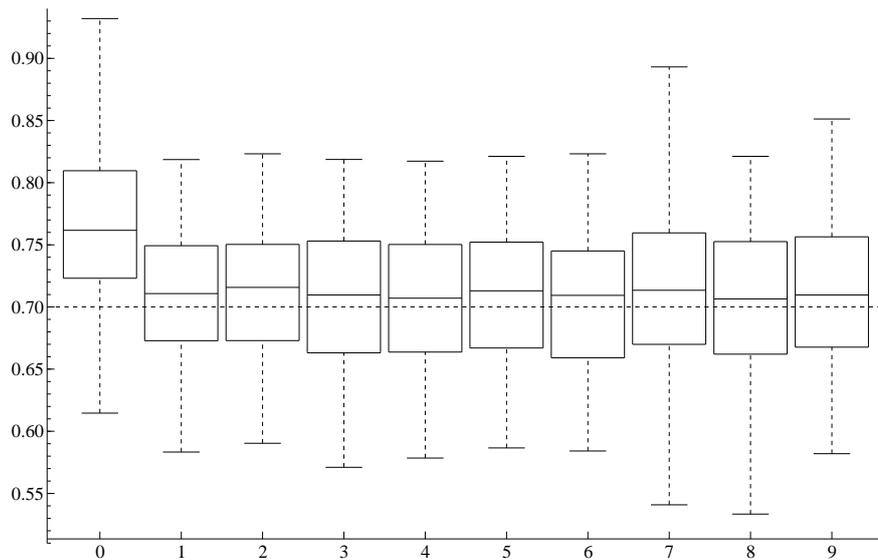


Figura 3: Estimativas para σ - Modelo AR. Filtro auxiliar. Parâmetros reais: $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. Linha pontilhada: valor real do parâmetro σ .

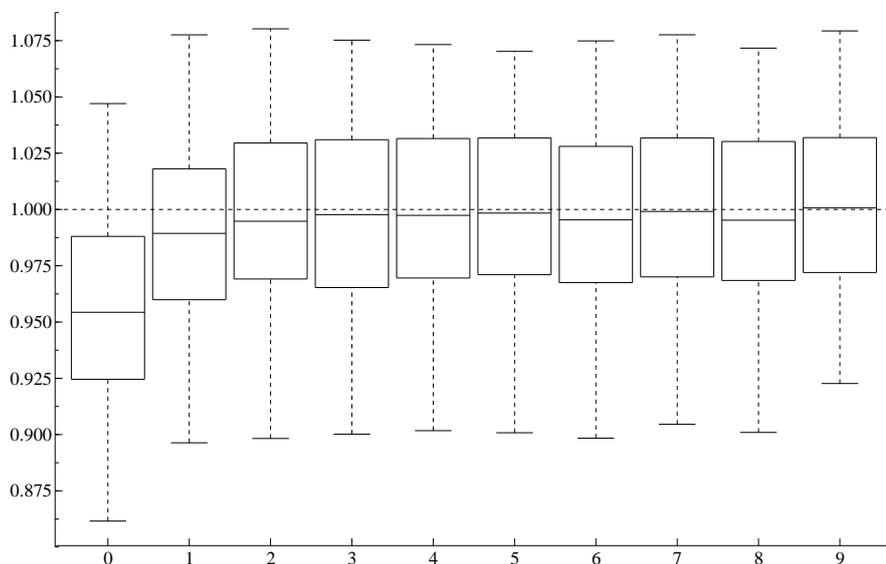


Figura 4: Estimativas para τ - Modelo AR. Filtro auxiliar. Parâmetros reais: $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. Linha pontilhada: valor real do parâmetro τ .

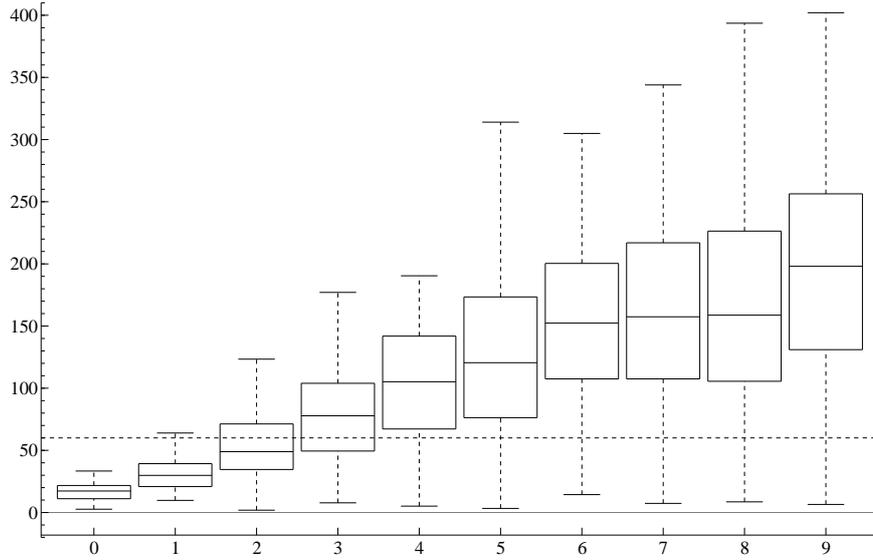


Figura 5: Tempo - Modelo AR. Filtro auxiliar. Parâmetros reais: $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. Linha pontilhada: 60 segundos.

todos os cenários. Por exemplo, para os parâmetros ϕ e σ que antes apresentavam uma subestimação e superestimação, respectivamente, na versão do filtro auxiliar. Agora apresentam estimativas quase exatas em alguns cenários, como, por exemplo, nos cenários 2 e 3. Quanto ao número ideal de partículas, o cenário 4 (box-plot 3 nas Figuras 6, 7 e 8) referente a 500 partículas apresenta estimativas quase exatas, mas o cenário 3 (box-plot 2 nas Figuras 6, 7 e 8) também apresenta boas estimativas para todos os parâmetros, sendo, portanto 250 partículas, uma boa quantidade de partículas para o filtro *bootstrap*. Em comparação ao filtro auxiliar que foi indicado 100 partículas como ideal, o filtro *bootstrap* apresenta estimativas razoáveis nesse cenário, mas com importante ganho quanto às estimativas nos cenários com 250 e 500 partículas e ainda com tempo médio computacional inferior a um minuto quando utilizadas 250 partículas como pode ser observado no box-plot 2 da Figura 9. Além das figuras com os box-plots, os resultados também podem ser analisados a partir da tabela do erro quadrático médio apresentada no apêndice.

3.2 Volatilidade Estocástica - SV

O modelo de volatilidade estocástica é definido como

$$\begin{aligned} X_t &= \phi X_{t-1} + U_t \\ Y_t &= \beta \exp\left(\frac{X_t}{2}\right) V_t \end{aligned} \tag{23}$$

onde $t = 1, 2, 3, \dots, n$; $U_t \sim \mathcal{N}(0, \sigma^2)$ e $V_t \sim \mathcal{N}(0, 1)$, tal que $U_t \perp V_t \perp X_1$, onde $X_1 \sim \mathcal{N}(0, \sigma^2/(1 - \phi^2))$, e ϕ e β são parâmetros do modelo.

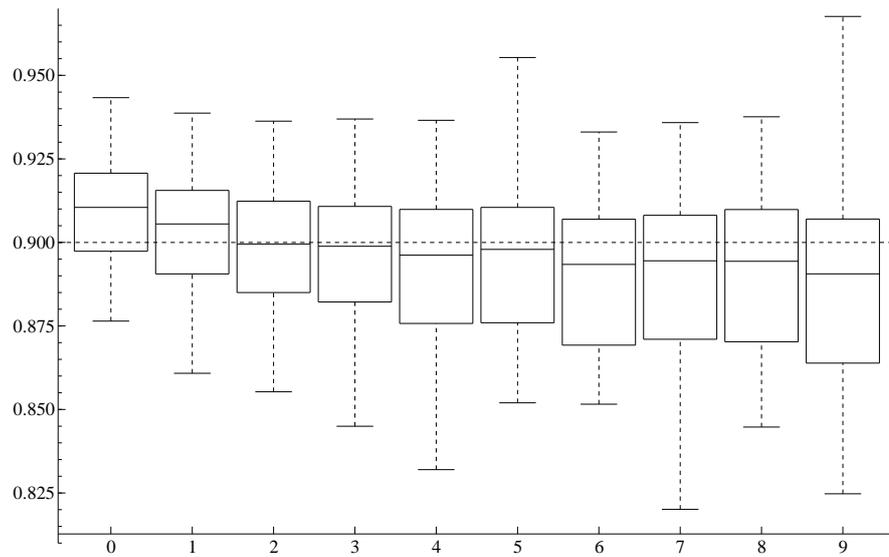


Figura 6: Estimativas para ϕ - Modelo AR. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. Linha pontilhada: valor real do parâmetro ϕ .

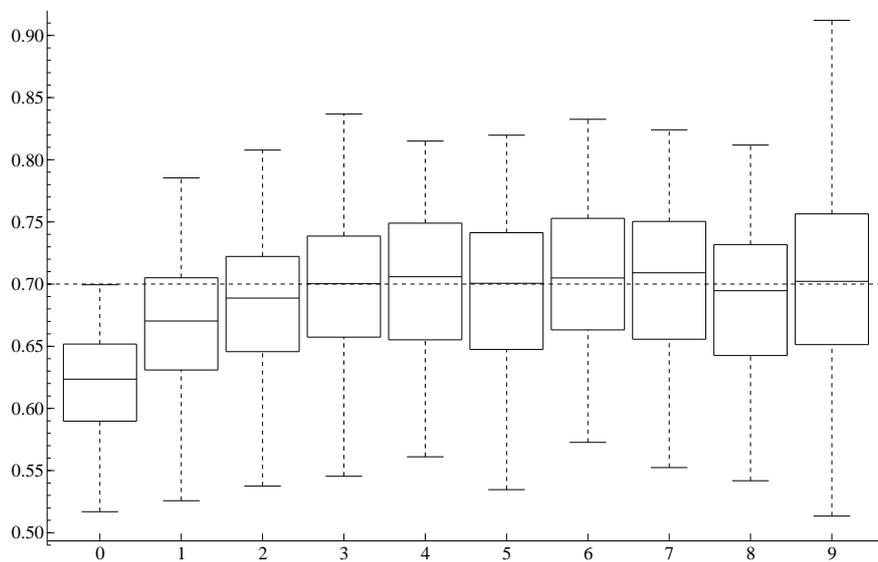


Figura 7: Estimativas para σ - Modelo AR. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. Linha pontilhada: valor real do parâmetro σ .

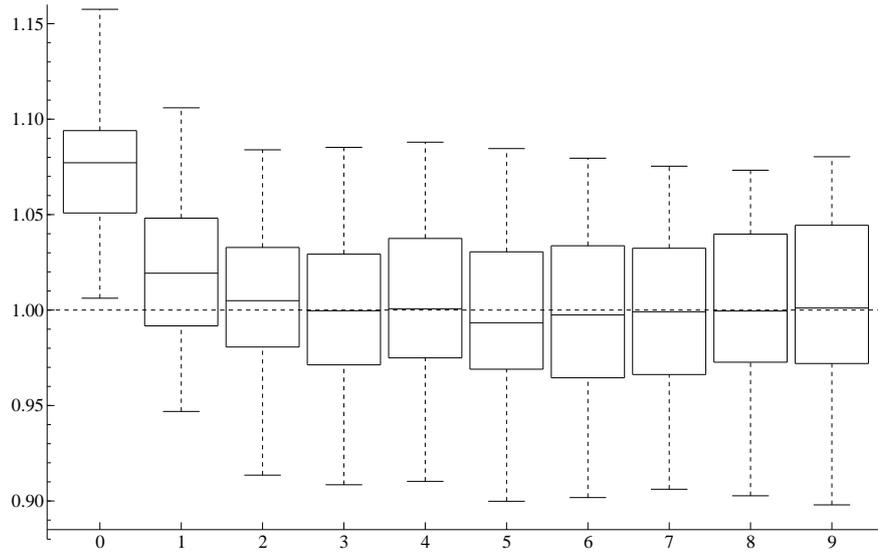


Figura 8: Estimativas para τ - Modelo AR. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. Linha pontilhada: valor real do parâmetro τ .

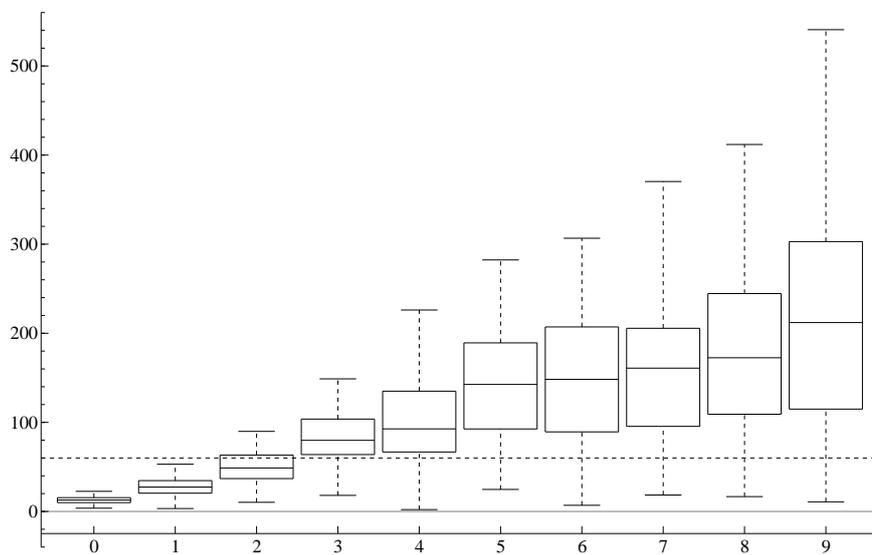


Figura 9: Tempo - Modelo AR. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.9$, $\sigma = 0.7$ e $\tau = 1$. Linha pontilhada: 60 segundos.

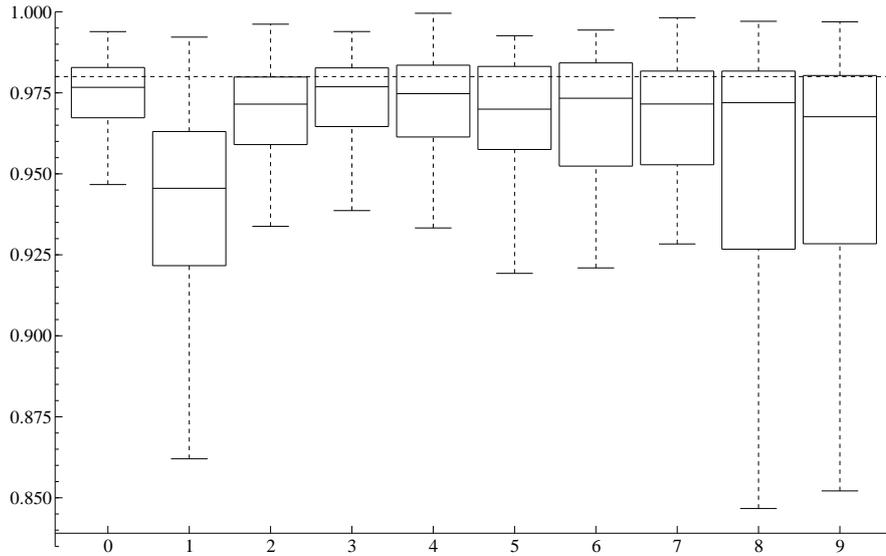


Figura 10: Estimativas para ϕ - Modelo SV. Filtro auxiliar. Parâmetros reais: $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$. Linha pontilhada: valor real do parâmetro ϕ .

O modelo de volatilidade estocástica nos permite observar como se comporta o método quando o modelo ainda é gaussiano, mas não linear. Neste trabalho utilizamos os seguintes parâmetros: $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$.

Como o modelo SV não é linear, não é possível obter uma densidade proposta ótima de forma analítica como foi feito no modelo autorregressivo. Sendo assim, para a implementação do método utilizamos uma densidade de proposta "cega", isto é, que não depende de y_t , $q(x_t|x_{t-1}, y_t, \theta) \sim \mathcal{N}(\phi x_{t-1}^{(i)}, \sigma^2)$.

Uma vez que temos a distribuição proposta, podemos estimar os estados latentes pelo filtro auxiliar e aplicar o Algoritmo 3 para obter as estimativas do vetor score e da matriz de informação observada.

Da mesma forma que no modelo autorregressivo, estamos interessados em obter um número ideal de partículas para estimar os parâmetros de um modelo de volatilidade estocástica. A partir do desenho do estudo de simulação citado anteriormente e com o modelo descrito acima com $T = 1000$ observações e parâmetros dados por $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$, para cada parâmetro ϕ, σ, β temos as seguintes estimativas nas Figuras 10, 11 e 12 respectivamente.

Nas Figuras 10, 11 e 12, a linha pontilhada representa o valor real do parâmetro simulado. Podemos observar que para o caso deste modelo de volatilidade estocástica um número baixo de partícula como 25, 100 ou 250 (box-plots 0, 1 e 2 nas Figuras 10, 11 e 12) não apresenta boas estimativas, como por exemplo quando se utiliza 100 partículas para os parâmetros ϕ e σ , enquanto que o parâmetro β não apresenta estimativas ruins assim como nos demais cenários. No entanto, a partir do quarto cenário (box-plot 3 nas Figuras 10, 11 e 12) referente a 500 partículas, todos os parâmetros tem boas estimativas, onde podemos observar uma subestimação para o ϕ e uma superestimação para o β que é comum para todos os cenários destes parâmetros, enquanto que o parâmetro σ

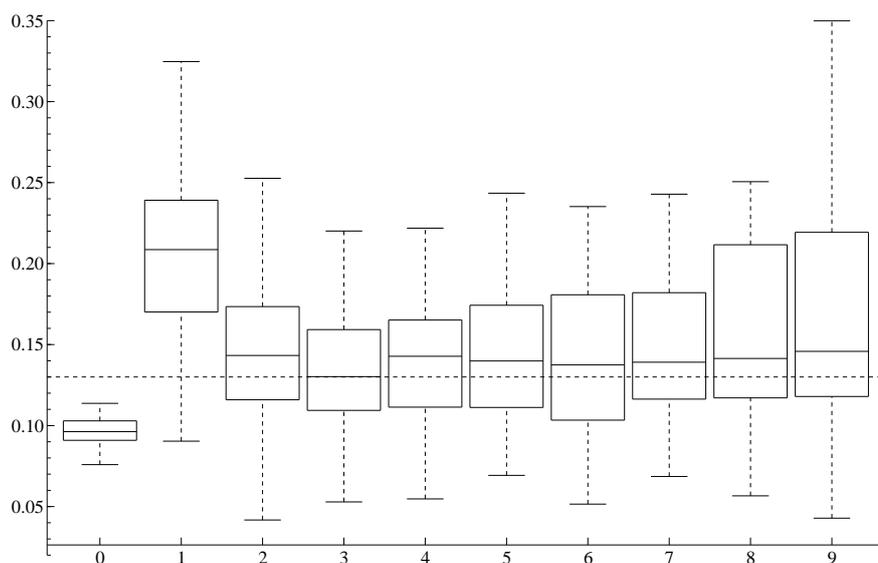


Figura 11: Estimativas para σ - Modelo SV. Filtro auxiliar. Parâmetros reais: $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$. Linha pontilhada: valor real do parâmetro σ .

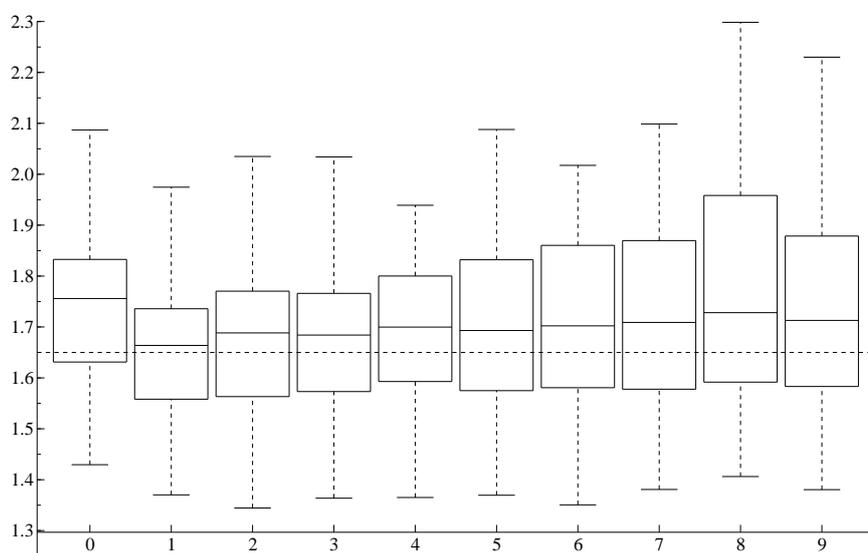


Figura 12: Estimativas para β - Modelo SV. Filtro auxiliar. Parâmetros reais: $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$. Linha pontilhada: valor real do parâmetro β .

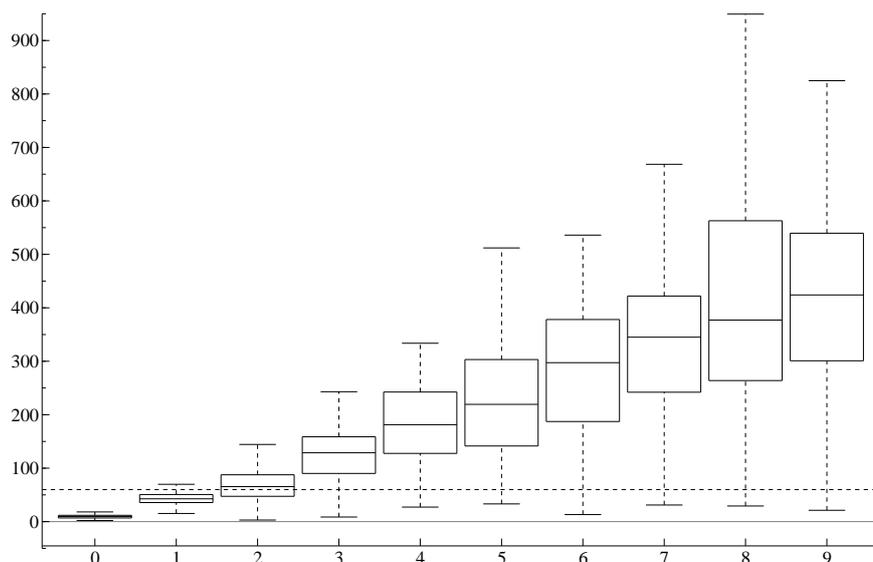


Figura 13: Tempo - Modelo SV. Filtro auxiliar. Parâmetros reais: $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$. Linha pontilhada: 60 segundos.

também apresenta superestimação, no entanto no cenário quatro, a estimativa se difere muito pouco do valor simulado. Portanto, podemos indicar que 500 é o número ideal de partículas para este modelo de volatilidade estocástica, dado que apresenta bons resultados não diferindo muito dos demais cenários com maior número de partículas. E como pode ser observado no box-plot 3 da Figura 13, tem o menor tempo médio gasto para convergência quando comparado com os cenários que apresentaram boas estimativas, aproximadamente dois minutos em média. Mais uma vez, podemos comprovar que nem sempre é necessário um número alto de partículas, que um número baixo de partículas já é suficiente para estimativas tão boas quanto um número alto e obtidas em menos tempo.

A partir do mesmo estudo de simulação, obtemos as estimativas com o uso do filtro *bootstrap* para os parâmetros ϕ , σ e β apresentadas nas Figuras 14, 15 e 16 respectivamente. Podemos observar que o uso do filtro *bootstrap* obteve estimativas tão boas quanto com a utilização do filtro auxiliar em todos os cenários. Podemos destacar que com o uso do filtro *bootstrap* houve uma pequena melhora nas estimativas em todos os cenários com relação ao parâmetro ϕ . E quanto aos parâmetros σ e β também houve uma pequena melhora nas estimativas em alguns cenários, mas mesmo os cenários que não apresentaram melhores estimativas, tiveram resultados semelhantes quando comparado com a versão com o filtro auxiliar. Quanto ao número de partículas ideal que no caso do filtro auxiliar foi indicado como o quarto cenário referente à 500 partículas. Com o uso do filtro *bootstrap*, esse número se manteve como ideal, apesar de que o uso de 250 partículas (terceiro cenário) também gera boas estimativas com a utilização do filtro *bootstrap* e como pode ser observado no box-plot 2 da Figura 17 tem tempo computacional inferior a um minuto na maioria das replicações. Além dos box-plots, os resultados também podem ser analisados a partir da tabela do erro

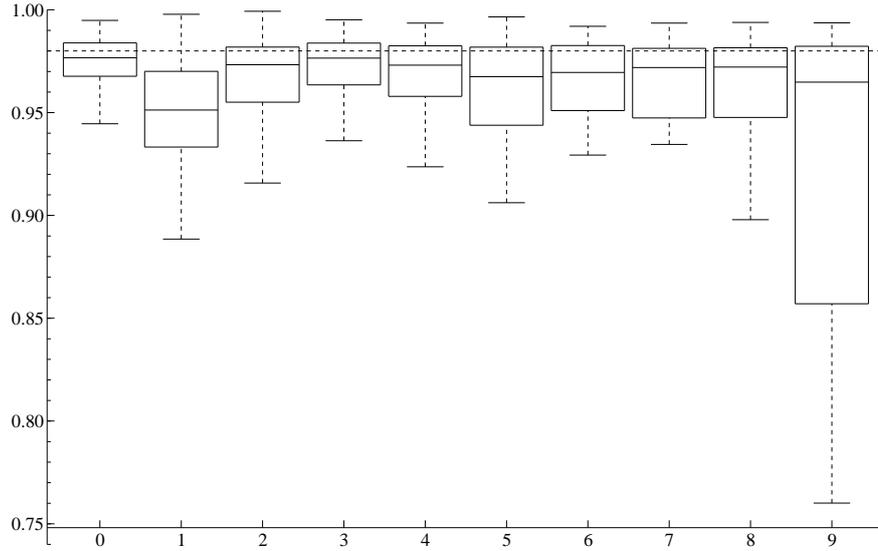


Figura 14: Estimativas para ϕ - Modelo SV. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$. Linha pontilhada: valor real do parâmetro ϕ .

quadrático médio apresentada no apêndice.

3.3 Modelo Poisson

O modelo Poisson é definido como

$$\begin{aligned} X_t &= \phi X_{t-1} + U_t \\ Y_t &= V_t \end{aligned} \quad (24)$$

onde $t = 1, 2, 3, \dots, n$; $U_t \sim \mathcal{N}(0, \sigma^2)$ e $V_t \sim \text{Poisson}(\beta \cdot \exp(X_t))$, tal que $U_t \perp V_t \perp X_1$, onde $X_1 \sim \mathcal{N}(0, \sigma^2/(1 - \phi^2))$.

O modelo Poisson nos permite observar como se comporta o método para modelos discretos, além disso os estados observados não são mais gaussianos, uma vez que agora são de uma distribuição Poisson. Neste trabalho utilizamos os seguintes parâmetros: $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$.

Como o modelo Poisson é não gaussiano, não é possível obter uma densidade proposta ótima de forma analítica como foi feito no modelo autorregressivo. Sendo assim, para a implementação do método utilizamos uma densidade de proposta "cega" assim como no caso anterior do modelo de volatilidade estocástica com $q(x_t | x_{t-1}^{(i)}, y_t, \theta) \sim \mathcal{N}(\phi x_{t-1}^{(i)}, \sigma^2)$.

Com isso, obtemos os estados latentes estimados pelo filtro auxiliar e podemos aplicar o Algoritmo 3 para obter as estimativas do vetor score e da matriz de informação observada.

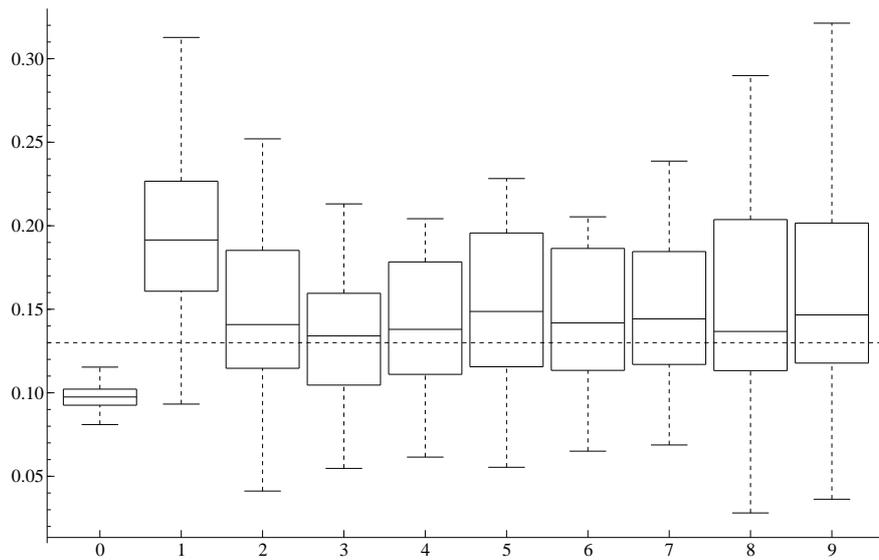


Figura 15: Estimativas para σ - Modelo SV. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$. Linha pontilhada: valor real do parâmetro σ .

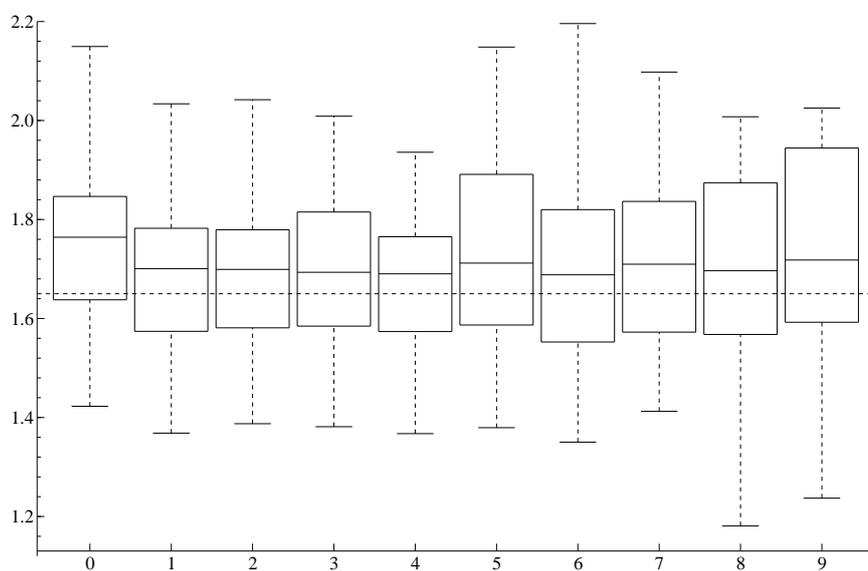


Figura 16: Estimativas para β - Modelo SV. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$. Linha pontilhada: valor real do parâmetro β .

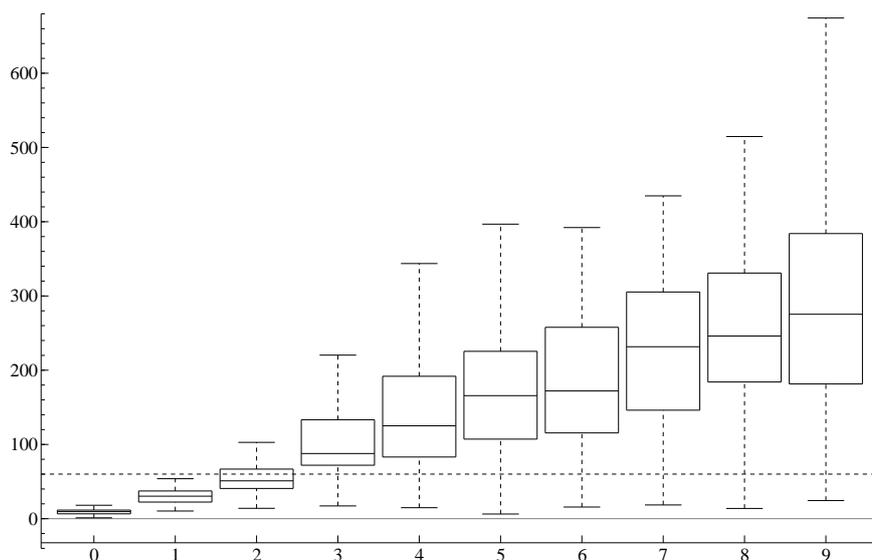


Figura 17: Tempo - Modelo SV. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.98$, $\sigma = 0.13$ e $\beta = 1.65$. Linha pontilhada: 60 segundos.

Como nos modelos anteriores também estamos interessados em obter um número ideal de partículas para estimar os parâmetros de um modelo Poisson. Portanto, foi feito o estudo de simulação descrito anteriormente também para o modelo Poisson descrito acima com $T = 1000$ observações e parâmetros dados por $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$, para cada parâmetro ϕ, σ, β temos as seguintes estimativas nas Figuras 18, 19 e 20 respectivamente.

Assim como nos demais box-plots, nas Figuras 18, 19 e 20, a linha pontilhada representa o valor real do parâmetro simulado. Podemos observar que o modelo Poisson tem um comportamento semelhante ao modelo de volatilidade estocástica quanto à variação do número de partículas. Em cenários com um número baixo de partícula como 25 ou 100 (box-plots 0 e 1 nas Figuras 18, 19 e 20) não apresenta boas estimativas, como por exemplo na estimativa do parâmetro ϕ com 100 partículas e para o parâmetro σ e β no primeiro cenário. No entanto, a partir do terceiro cenário (box-plot 2 nas Figuras 18, 19 e 20) referente a 250 partículas, todos os parâmetros tem boas estimativas, e assim como no modelo de volatilidade estocástica com comportamento semelhante, podemos observar uma subestimação para o ϕ comum para todos os cenários. Enquanto que os parâmetros σ e β apresentam estimativas quase exatas em grande parte dos cenários simulados. Portanto, tendo em consideração também a análise do tempo médio gasto para convergência como pode ser observado no box-plot 2 da Figura 21 podemos indicar que 250 é o número ideal de partículas para este modelo Poisson, dado que apresenta bons resultados não diferindo muito dos demais cenários com maior número de partículas e com o menor tempo médio gasto para convergência quando comparado com os cenários que apresentaram boas estimativas.

As estimativas obtidas via filtro *bootstrap* para os parâmetros ϕ, σ e β , a partir do mesmo estudo de simulação feito anteriormente, são apresentadas nas Figuras 22, 23 e

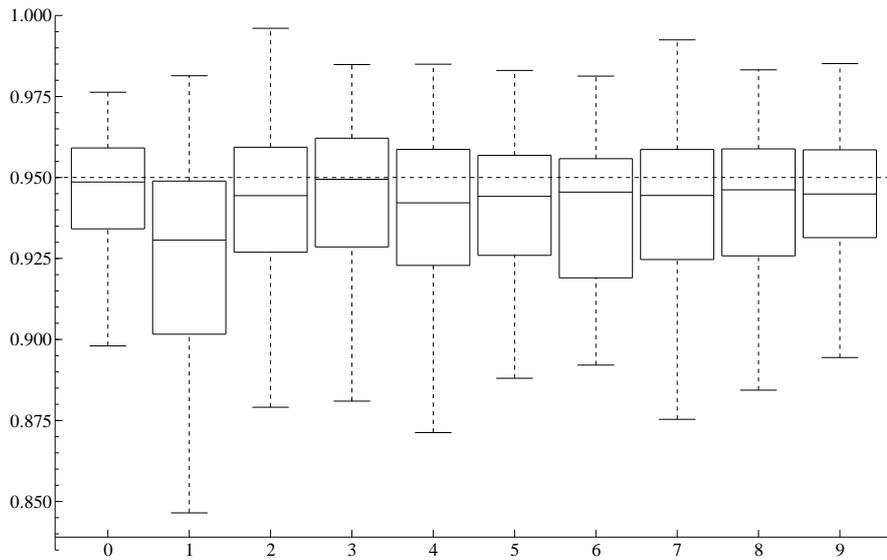


Figura 18: Estimativas para ϕ - Modelo Poisson. Filtro auxiliar. Parâmetros reais: $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$. Linha pontilhada: valor real do parâmetro ϕ .

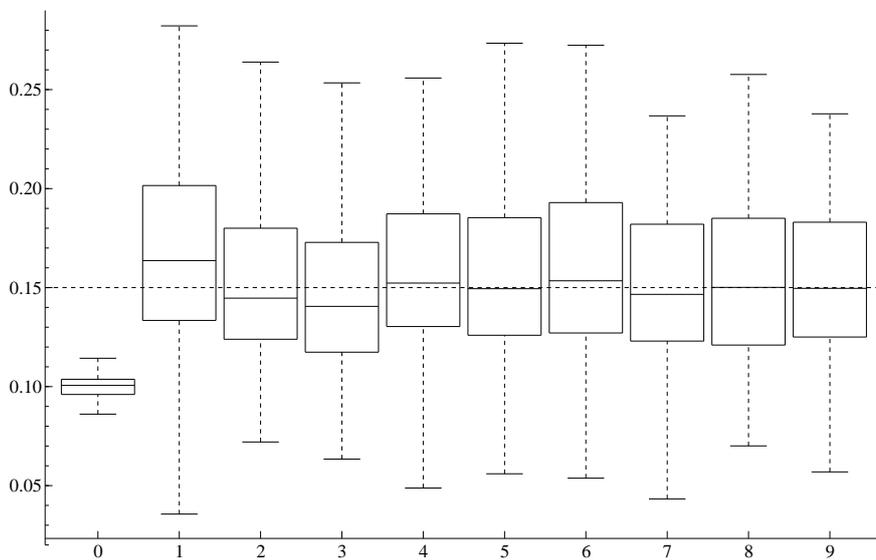


Figura 19: Estimativas para σ - Modelo Poisson. Filtro auxiliar. Parâmetros reais: $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$. Linha pontilhada: valor real do parâmetro σ .

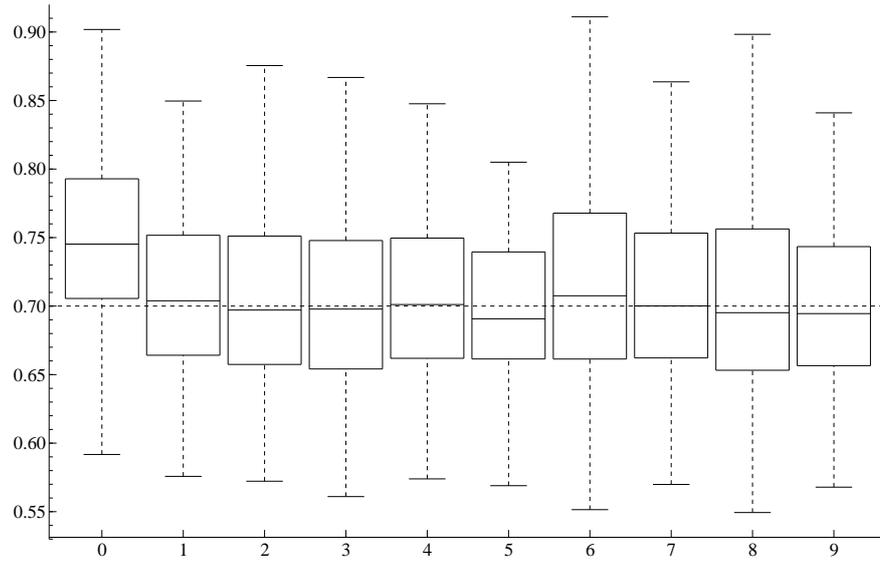


Figura 20: Estimativas para β - Modelo Poisson. Filtro auxiliar. Parâmetros reais: $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$. Linha pontilhada: valor real do parâmetro β .

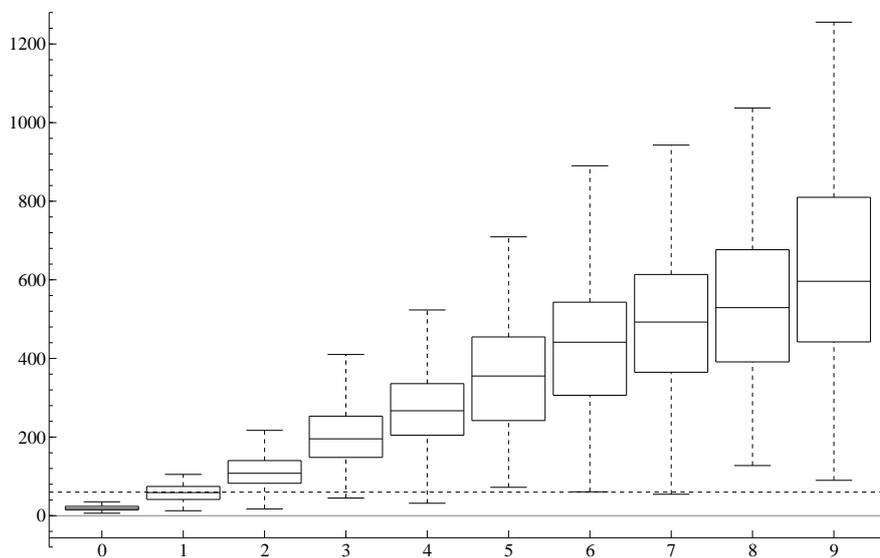


Figura 21: Tempo - Modelo Poisson. Filtro auxiliar. Parâmetros reais: $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$. Linha pontilhada: 60 segundos.

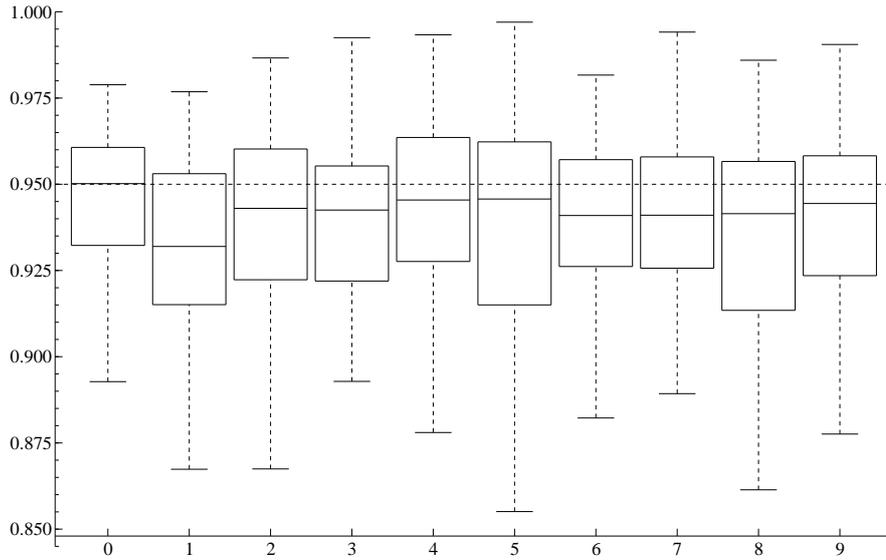


Figura 22: Estimativas para ϕ - Modelo Poisson. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$. Linha pontilhada: valor real do parâmetro ϕ .

24, respectivamente. No modelo Poisson podemos observar que o uso do filtro *bootstrap* proporcionou uma significativa melhora das estimativas em muitos cenários quando comparado à versão com o filtro auxiliar. Destacamos a melhora nas estimativas dos parâmetros σ e β que com o uso do filtro *bootstrap* passaram a ser quase exatas na maioria dos cenários. Com relação ao número ideal de partículas, o cenário 3 referente a 250 partículas continua sendo o ideal, mas, agora, com uma melhora nas estimativas, principalmente dos parâmetros σ e β que apresentam resultados quase exatos. Como pode ser observado na Figura 25, o tempo médio computacional no cenário de 250 partículas (box-plot 2 na Figura 25) foi reduzido com relação ao caso anterior, mas ainda assim acima de um minuto. Os resultados também podem ser analisados a partir da tabela do erro quadrático médio apresentada no apêndice.

4 Conclusão

Este trabalho apresentou um estudo comparativo sobre um método para estimar os parâmetros de modelos de espaço de estados. Esse método foi proposto por Nemeth et al. (2016) e apresenta um método Monte Carlo sequencial, mais especificamente o filtro de partículas auxiliar, para estimar o vetor score e a matriz de informação observada, e, então, o método do gradiente é utilizado para obter as estimativas dos parâmetros de um modelo de espaço de estados. Esse algoritmo utiliza métodos de densidade de kernel e Rao-Blackwell para melhorar as estimativas com relação aos algoritmos propostos anteriormente por Poyiadjis et al. (2011), que apresentava a variância da estimativa crescendo pelo menos quadraticamente com o tempo e tempo computacional linear no número de partículas ou variância linear no tempo, mas custo

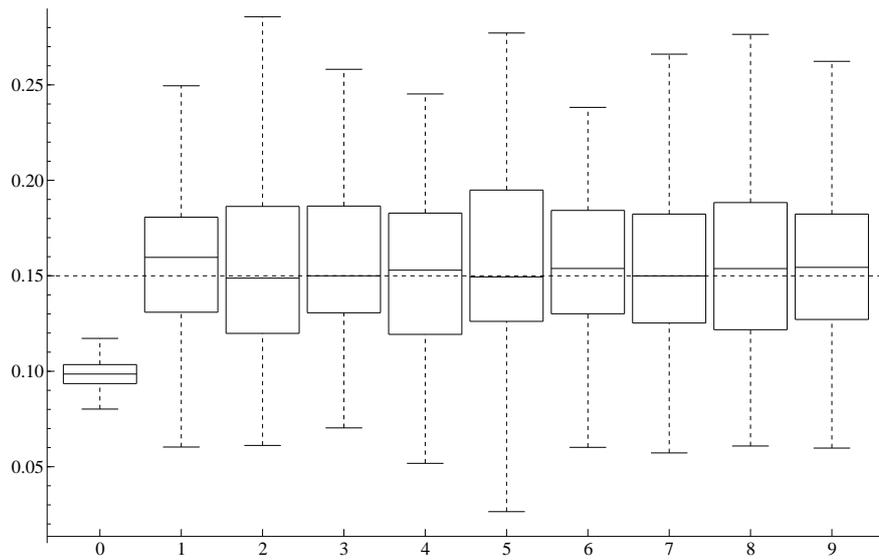


Figura 23: Estimativas para σ - Modelo Poisson. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$. Linha pontilhada: valor real do parâmetro σ .

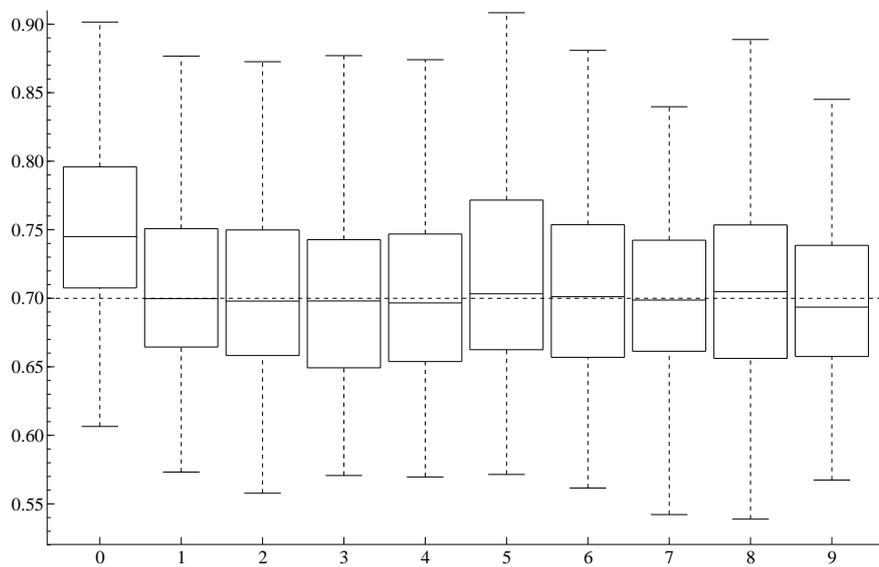


Figura 24: Estimativas para β - Modelo Poisson. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$. Linha pontilhada: valor real do parâmetro β .

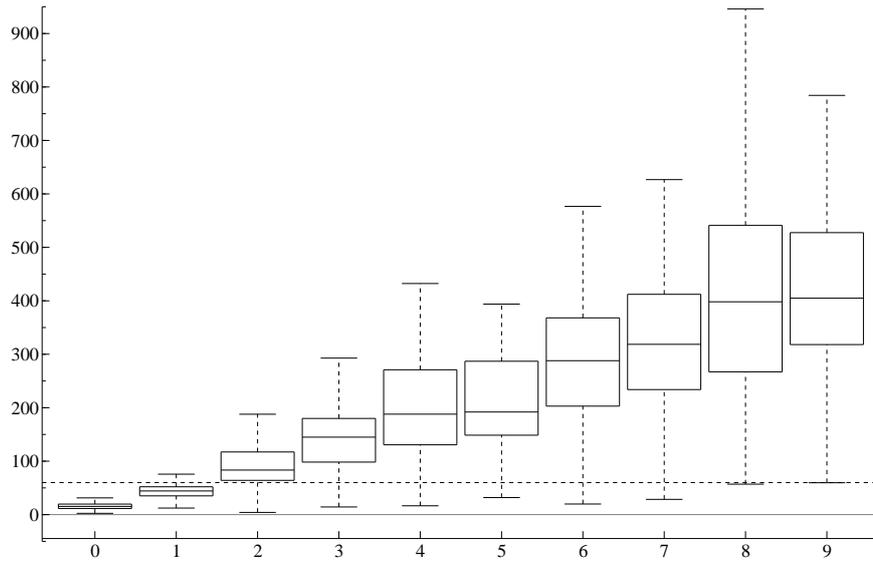


Figura 25: Tempo - Modelo Poisson. Filtro *bootstrap*. Parâmetros reais: $\phi = 0.95$, $\sigma = 0.15$ e $\beta = 0.70$. Linha pontilhada: 60 segundos.

computacional quadrático. Enquanto que a proposta de Nemeth et al. (2016) apresenta variância da estimativa linear com o tempo, e tem o tempo computacional também linear no número de partículas. Também propomos uma versão com o uso do filtro *bootstrap* ao invés do filtro de partículas auxiliar, a fim de comparar com o modelo proposto anteriormente e sendo o filtro *bootstrap* mais simples que o auxiliar.

O método apresentou boas estimativas dos parâmetros de diferentes modelos de espaço de estados, como autoregressivo, volatilidade estocástica e Poisson. Além disso, foi feita uma análise mais detalhada quanto ao número de partículas necessário a se utilizar para tal estimativa dependendo de cada modelo (autorregressivo, volatilidade estocástica, Poisson). Essa análise foi feita preestabelecendo um critério de convergência para até a quarta casa decimal. Conforme apresentado na seção anterior, podemos perceber que nem sempre é necessário um número grande de partículas. Um número pequeno de partículas, como, por exemplo, 100 partículas no modelo autorregressivo, já pode ser considerado um número suficiente para obter estimativas tão boas quanto com um número maior de partículas e consumindo consideravelmente menos tempo de execução. Na comparação do método com a versão com a utilização do filtro *bootstrap*, este último apresentou melhorias nas estimativas da maioria dos cenários para os modelos analisados na seção anterior e demandando menos tempo de execução quando comparado com a versão do filtro auxiliar. Outras análises mais detalhadas ou melhorias ainda podem ser feitas sobre o método, como, por exemplo, uma análise detalhada sobre o tamanho do passo γ_k nas equações (7) e (15) e até mesmo a troca da equação de atualização das estimativas dos parâmetros.

Apêndice

O Algoritmo 3 utiliza as derivadas de primeira e segunda ordem das log-equações de cada modelo. Tais derivadas são apresentadas, a seguir, para cada um dos modelos analisados. Primeiro, o modelo autorregressivo

$$\begin{aligned}\nabla \log f_{\theta}(x_t^{(i)} | x_{t-1}^{(k_i)}) &= \left(\frac{\partial X_t | X_{t-1}}{\partial \phi}, \frac{\partial X_t | X_{t-1}}{\partial \sigma}, \frac{\partial X_t | X_{t-1}}{\partial \tau} \right) = \left(\frac{\partial X_t | X_{t-1}}{\partial \phi}, \frac{\partial X_t | X_{t-1}}{\partial \sigma}, 0 \right) \\ \nabla \log g_{\theta}(y_t | x_t^{(i)}) &= \left(\frac{\partial Y_t | X_t}{\partial \phi}, \frac{\partial Y_t | X_t}{\partial \sigma}, \frac{\partial Y_t | X_t}{\partial \tau} \right) = \left(0, 0, \frac{\partial Y_t | X_t}{\partial \tau} \right)\end{aligned}$$

onde

$$\begin{aligned}\frac{\partial X_t | X_{t-1}}{\partial \phi} &= \frac{x_{t-1}^{(k_i)}}{\sigma^2} (x_t^{(i)} - \phi x_{t-1}^{(k_i)}) \\ \frac{\partial X_t | X_{t-1}}{\partial \sigma} &= -\frac{1}{\sigma} + \frac{(x_t^{(i)} - \phi x_{t-1}^{(k_i)})^2}{\sigma^3} \\ \frac{\partial Y_t | X_t}{\partial \tau} &= -\frac{1}{\tau} + \frac{(y_t - x_t^{(i)})^2}{\tau^3}\end{aligned}$$

e para as derivadas de segunda ordem

$$\begin{aligned}\frac{\partial^2 X_t | X_{t-1}}{(\partial \phi)^2} &= -\frac{(x_{t-1}^{(k_i)})^2}{\sigma^2} \\ \frac{\partial^2 X_t | X_{t-1}}{\partial \phi \partial \sigma} &= \frac{\partial^2 X_t | X_{t-1}}{\partial \sigma \partial \phi} = -\frac{2x_{t-1}^{(k_i)}}{\sigma^3} (x_t^{(i)} - \phi x_{t-1}^{(k_i)}) \\ \frac{\partial^2 X_t | X_{t-1}}{(\partial \sigma)^2} &= \frac{1}{\sigma^2} - \frac{3(x_t^{(i)} - \phi x_{t-1}^{(k_i)})^2}{\sigma^4} \\ \frac{\partial^2 Y_t | X_t}{(\partial \tau)^2} &= \frac{1}{\tau^2} - \frac{3(y_t - x_t^{(i)})^2}{\tau^4}\end{aligned}$$

Agora, para o modelo de volatilidade estocástica, as derivadas de primeira ordem são

$$\begin{aligned}\nabla \log f_{\theta}(x_t^{(i)} | x_{t-1}^{(k_i)}) &= \left(\frac{\partial X_t | X_{t-1}}{\partial \phi}, \frac{\partial X_t | X_{t-1}}{\partial \sigma}, \frac{\partial X_t | X_{t-1}}{\partial \beta} \right) = \left(\frac{\partial X_t | X_{t-1}}{\partial \phi}, \frac{\partial X_t | X_{t-1}}{\partial \sigma}, 0 \right) \\ \nabla \log g_{\theta}(y_t | x_t^{(i)}) &= \left(\frac{\partial Y_t | X_t}{\partial \phi}, \frac{\partial Y_t | X_t}{\partial \sigma}, \frac{\partial Y_t | X_t}{\partial \beta} \right) = \left(0, 0, \frac{\partial Y_t | X_t}{\partial \beta} \right)\end{aligned}$$

onde

$$\begin{aligned}\frac{\partial X_t|X_{t-1}}{\partial \phi} &= \frac{x_{t-1}^{(k_i)}}{\sigma^2} (x_t^{(i)} - \phi x_{t-1}^{(k_i)}) \\ \frac{\partial X_t|X_{t-1}}{\partial \sigma} &= -\frac{1}{\sigma} + \frac{(x_t^{(i)} - \phi x_{t-1}^{(k_i)})^2}{\sigma^3} \\ \frac{\partial Y_t|X_t}{\partial \beta} &= -\frac{1}{\beta} + \frac{y_t^2 \exp(-x_t^{(i)})}{\beta^3}\end{aligned}$$

e para as derivadas de segunda ordem

$$\begin{aligned}\frac{\partial^2 X_t|X_{t-1}}{(\partial \phi)^2} &= -\frac{(x_{t-1}^{(k_i)})^2}{\sigma^2} \\ \frac{\partial^2 X_t|X_{t-1}}{\partial \phi \partial \sigma} &= \frac{\partial^2 X_t|X_{t-1}}{\partial \sigma \partial \phi} = -\frac{2x_{t-1}^{(k_i)}}{\sigma^3} (x_t^{(i)} - \phi x_{t-1}^{(k_i)}) \\ \frac{\partial^2 X_t|X_{t-1}}{(\partial \sigma)^2} &= \frac{1}{\sigma^2} - \frac{3(x_t^{(i)} - \phi x_{t-1}^{(k_i)})^2}{\sigma^4} \\ \frac{\partial^2 Y_t|X_t}{(\partial \beta)^2} &= \frac{1}{\beta^2} - \frac{3y_t^2 \exp(-x_t^{(i)})}{\beta^4}\end{aligned}$$

Por fim, para o modelo Poisson, as derivadas de primeira ordem são

$$\begin{aligned}\nabla \log f_{\theta}(x_t^{(i)}|x_{t-1}^{(k_i)}) &= \left(\frac{\partial X_t|X_{t-1}}{\partial \phi}, \frac{\partial X_t|X_{t-1}}{\partial \sigma}, \frac{\partial X_t|X_{t-1}}{\partial \beta} \right) = \left(\frac{\partial X_t|X_{t-1}}{\partial \phi}, \frac{\partial X_t|X_{t-1}}{\partial \sigma}, 0 \right) \\ \nabla \log g_{\theta}(y_t|x_t^{(i)}) &= \left(\frac{\partial Y_t|X_t}{\partial \phi}, \frac{\partial Y_t|X_t}{\partial \sigma}, \frac{\partial Y_t|X_t}{\partial \beta} \right) = \left(0, 0, \frac{\partial Y_t|X_t}{\partial \beta} \right)\end{aligned}$$

onde

$$\begin{aligned}\frac{\partial X_t|X_{t-1}}{\partial \phi} &= \frac{x_{t-1}^{(k_i)}}{\sigma^2} (x_t^{(i)} - \phi x_{t-1}^{(k_i)}) \\ \frac{\partial X_t|X_{t-1}}{\partial \sigma} &= -\frac{1}{\sigma} + \frac{(x_t^{(i)} - \phi x_{t-1}^{(k_i)})^2}{\sigma^3} \\ \frac{\partial Y_t|X_t}{\partial \beta} &= -\exp(x_t^{(i)}) + \frac{y_t}{\beta}\end{aligned}$$

e para as derivadas de segunda ordem

$$\begin{aligned}\frac{\partial^2 X_t | X_{t-1}}{(\partial \phi)^2} &= -\frac{(x_{t-1}^{(k_i)})^2}{\sigma^2} \\ \frac{\partial^2 X_t | X_{t-1}}{\partial \phi \partial \sigma} &= \frac{\partial^2 X_t | X_{t-1}}{\partial \sigma \partial \phi} = -\frac{2x_{t-1}^{(k_i)}}{\sigma^3} (x_t^{(i)} - \phi x_{t-1}^{(k_i)}) \\ \frac{\partial^2 X_t | X_{t-1}}{(\partial \sigma)^2} &= \frac{1}{\sigma^2} - \frac{3(x_t^{(i)} - \phi x_{t-1}^{(k_i)})^2}{\sigma^4} \\ \frac{\partial^2 Y_t | X_t}{(\partial \beta)^2} &= -\frac{y_t}{\beta^2}\end{aligned}$$

A análise dos resultados, além dos diversos box-plots apresentados nas figuras anteriores, também pode ser feita a partir do erro quadrático médio (EQM). O EQM mede a média dos erros quadrados, isto é, a diferença do estimador e do parâmetro que estamos estimando e é calculado por

$$EQM = \frac{1}{m} \sum_{i=1}^m (\hat{\theta}_i - \theta_i)^2$$

onde m é o número de replicações, aqui $m = 100$, e θ é o vetor de parâmetros enquanto que $\hat{\theta}$ é o vetor das estimativas para θ . Os resultados apresentados nas próximas tabelas são as raízes quadradas dos EQM's de cada parâmetro ($\phi, \sigma, \tau, \beta$) para cada um dos cenários simulados tanto para a versão com filtro auxiliar quanto para a versão com filtro *bootstrap*.

Tabela 1: EQM - Modelo AR. Filtro auxiliar

Partículas	25	100	250	500	750	1000	1250	1500	1750	2000
ϕ	0.032	0.047	0.130	0.152	0.131	0.165	0.079	0.111	0.085	0.138
σ	0.085	0.159	1.914	1.247	0.123	1.144	0.149	1.848	1.133	5.099
τ	0.060	0.189	14.36	13.95	73.79	20.10	13.29	97.11	21.74	56.56

Tabela 2: EQM - Modelo SV. Filtro auxiliar

Partículas	25	100	250	500	750	1000	1250	1500	1750	2000
ϕ	0.018	0.054	0.034	0.023	0.102	0.141	0.219	0.114	0.248	0.230
σ	0.034	0.096	0.180	1.162	3.164	3.453	16.69	21.03	12.65	10.06
β	0.199	0.168	2.592	44.83	174.8	139.1	216.4	82.05	229.5	172.4

Tabela 3: EQM - Modelo Poisson. Filtro auxiliar

Partículas	25	100	250	500	750	1000	1250	1500	1750	2000
ϕ	0.030	0.065	0.044	0.056	0.085	0.120	0.141	0.128	0.136	0.157
σ	0.050	0.051	2.168	0.855	9.475	1.858	5.119	3.813	6.530	6.900
β	0.089	0.068	2.525	0.331	19.68	6.178	43.22	1587.6	2.143	48.96

Tabela 4: EQM - Modelo AR. Filtro *bootstrap*

Partículas	25	100	250	500	750	1000	1250	1500	1750	2000
ϕ	0.018	0.019	0.087	0.085	0.132	0.101	0.159	0.157	0.141	0.120
σ	0.087	0.062	0.278	0.137	0.386	0.451	1.645	0.207	0.729	0.237
τ	0.082	0.042	0.686	3.949	50.21	10.37	64.94	303.98	36.96	31.34

Tabela 5: EQM - Modelo SV. Filtro *bootstrap*

Partículas	25	100	250	500	750	1000	1250	1500	1750	2000
ϕ	0.026	0.046	0.069	0.115	0.205	0.265	0.148	0.259	0.249	0.284
σ	0.035	1.032	1.833	5.387	10.05	52.74	50.31	64.49	0.072	0.102
β	0.204	153550	68.1	22.28	502.9	339.8	92.87	365.3	398.9	1.584

Tabela 6: EQM - Modelo Poisson. Filtro *bootstrap*

Partículas	25	100	250	500	750	1000	1250	1500	1750	2000
ϕ	0.033	0.036	0.094	0.145	0.125	0.126	0.111	0.142	0.123	0.099
σ	0.053	0.043	1.020	3.375	7.007	6.995	19.16	6.317	6.071	12.09
β	0.094	0.071	109.5	110.3	12.12	28.71	13.47	58.92	4.401	2.684

Referências Bibliográficas

- Cappé, O., Moulines, E., and Ryden, T. (2005). *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer New York.
- Crisan, D. and Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746.
- Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208.
- Durbin, J. and Koopman, S. J. (2012). *Time Series Analysis by State Space Methods: Second Edition (Oxford Statistical Science Series)*. Oxford University Press.
- Gordon, N., Salmond, D., and Smith, A. F. M. (1993). Novel approach to non-linear/non-gaussian bayesian state estimation. *Proceedings of the IEEE F*, 140(2):107–113.
- Liu, J. and West, M. (2001). *Combined parameter and state estimation in simulation based filtering* em Doucet, A., de Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.
- Louis, T. (1982). Finding the observed information matrix when using the em algorithm. *Journal of the Royal Statistical Society. Series B*, 44(2):226–233.
- Nemeth, C., Fearnhead, P., and Mihaylova, L. (2016). Particle approximations of the score and observed information matrix for parameter estimation in state–space models with linear computational cost. *Journal of Computational and Graphical Statistics*, 25(4):1138–1157.
- Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.
- Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80.
- Álvares da Silva, F. C. (2016). *Métodos de Monte Carlo Sequenciais Bayesianos : Aspectos Computacionais, Inferenciais e Aplicações (PhD thesis)*. Universidade Federal de Minas Gerais.