Gauss Doğrusal Durum-Uzay Modellerinde İleri Doğru Düzleştirme ve Anında Beklenti-En Büyütme Algoritması
Forward Smoothing and Online Expectation-Maximisation in Gaussian Linear State-Space Models

Sinan Yıldırım[1], *A. Taylan Cemgil*[2]

1. Statistical Laboratory, Cambridge Üniversitesi, sy276@cam.ac.uk
2. Bilgisayar Mühendisliği Bölümü, Boğaziçi Üniversitesi, taylan.cemgil@boun.edu.tr

ÖZETÇE

Bu calışmada[1]*, Gauss doğrusal durum-uzay (GDDU) modellerinde ileri doğru düzleştirme özyinelemesini inceledik. Bu özyinelemenin stokastik bir yaklaşıklamasından faydalanarak GDDU modelleri için var olan beklenti-enbüyütme (EM) algoritmasının anında sürümünü geliştirdik. Anında EM algoritmasının başarımını geleneksel EM ile karşılaştırarak algoritmanın uzun veri dizileri için kullanımındaki yararını gösterdik.*

ABSTRACT

In this work, we studied forward-only smoothing recursion in Gaussian linear state-space (GLSS) models. We exploited a stochastic approximation of this recursion to develop an online version of the expectation-maximisation (EM) algorithm for GLSS models. We compared the performance of online EM with the conventional EM and demonstrated the advantages of its use in case of long data sequences.

1. Giriş

Gauss doğrusal durum-uzay (GDDU) modelleri, genel durum-uzay modellerinin kullanımı çok geniş olan bir türüdür. Bu yüzden, bu modellerde parametre kestirimi önemli bir problemdir. Parametre kestirimi için kullanılan en yaygın yöntem bir en büyük olabilirlik (EBO) kestirimi yöntemi olan *beklenti-en büyütme* (EM) algoritmasıdır [1]. Bu algoritma, belli bir takım yeterli istatistiklerin bir sonsal dağılım üzerinden kestirilmesini gerektirir. Bu kestirimler için bilinen en yaygın yol veriyi bir ileri bir de geri yönde işleyen *ileri doğru süzgeçleme-geriye doğru düzleştirme* (İDS-GDD) yöntemidir [2]. Bu calışmada incelenecek olan bir diğer yöntem ise veriyi yalnızca ileri yönde işleyen *ileri-doğru düzleştirme özyinelemesi* (İDDÖ) yöntemidir. Bu yöntem için gerekli olan özyineleme kuralları GDDU modelleri için ilk defa [3]'de türetilmiştir. Bu çalışmada ise, aynı özyineleme farklı bir şekilde türetilecektir.

İleri doğru düzleştirmenin bir diğer önemli yararı ise EM algoritmasının anında sürümüne olanak vermesidir. İDDÖ'nün staokastik bir yaklaşıklamasına dayanan *anında EM algoritması* sonlu durum-uzay modellerinde [4]'te gösterilmiştir. [5]'te ise bu stokastik yaklaşıklamanın genel durum-uzay modellerine uygulanabileceği gösterilmiştir. Burada gösterdiğimiz anında EM algoritması, [5]'in GDDU modellerine özel uygulaması olup, bu modellerin yaygınlığı düşünüldüğünde bizce önem taşımaktadır.

Makalenin geri kalanında; önce GDDU modeli tanıtılacak, daha sonra İDDÖ anlatılacak ve bu özyinelemenin EM ve anında EM algoritmaları için nasıl kullanılabileceği gösterilecektir. İki algoritmanın başarımları yapay veri üzerinde karşılaştırılacaktır.

2. Gauss doğrusal durum-uzay modelleri

Gauss doğrusal durum-uzay (GDDU) modeli, vektör değerli rastgele süreçler olan $\{X_k \in \mathbb{R}^n\}_{k\geq 0}$ ve $\{Y_k \in \mathbb{R}^r\}_{k\geq 0}$'den oluşur. Bir Markov süreci olan $\{X_k\}_{k\geq 0}$ *gizli süreç*, $\{X_k\}_{k\geq 0}$'ya koşullu bağımsız olan $\{Y_k \in \mathbb{R}^r\}_{k\geq 0}$ ise *gözlemlenen sürec* olarak adlandırılır. X_0 için başlangıç dağılımı $\pi_\theta(x)$, $\{X_k\}_{k\geq 1}$ için Markov geçiş dağılımları $f_{k+1,\theta}(x'|x)$ ve Y_k'larin X_k'lara koşullu dağılımları $g_{k,\theta}(y|x)$ Gauss dağılımlarıdır. Yani, $k \geq 0$ için

$$\begin{aligned} \pi_\theta(x) &= \mathcal{N}(x;\mu_0,U_0),\ f_{k+1,\theta}(x'|x) = \mathcal{N}(x';F_k x, U_k) \\ g_{k,\theta}(y|x) &= \mathcal{N}(y;G_k x, V_k) \end{aligned}$$

Burada, $\{F_k\}_{k\geq 0}$ ve $\{G_k\}_{k\geq 0}$ sırasıyla $n \times n$ ve $r \times n$ matris dizileri olup; $\{U_k\}_{k\geq 0}$ ve $\{V_k\}_{k\geq 0}$ ise sırasıyla $n \times n$ yarı-kesin artı ve $r \times r$ kesin-artı matris dizileridir. Bu matrisler modelin parametreleri olup θ ile gösterilmiştir.

2.1. Yeterli istatistikler ve EM algoritması

GDDU modellerinde çoğu zaman şu yeterli istatistiklerin kestirimleri aranır: (Fonksiyonların Y_k'lara olan bağımlılığını yazımımızda ihmal edecek olursak)

$$\begin{aligned} S_k^0(x_{1:k}) &= \sum_{i=1}^{k} x_i x_i^T,\ S_k^1(x_{0:k}) = \sum_{i=0}^{k} x_i x_i^T, \\ S_k^2(x_{0:k-1}) &= \sum_{i=1}^{k} x_{i-1} x_{i-1}^T,\ S_k^3(x_{0:k}) = \sum_{i=1}^{k} x_{i-1} x_i^T, \\ S_k^4(x_{0:k}) &= \sum_{i=0}^{k} x_i y_i^T,\ S_k^5 = \sum_{i=0}^{k} y_i y_i^T \end{aligned} \quad (1)$$

[1]ATC, bu çalışmada TUBITAK 110E292 Bayesci Tensör ayrıştırma (BAYTEN) projesi kapsamında desteklenmektedir.

Genel olarak, herhangi bir $S_k(\cdot)$ için gösterecek olursak, aranan kestirim

$$\widehat{S}_k(\theta) = \mathbb{E}_\theta \left[S_k(X_{0:k})|Y_{0:k}\right] \tag{2}$$

$S_k(\cdot)$'nin $X_{0:k}$'nin sonsal dağılımı üzerinden elde edilen beklenen değeridir; bu nedenle de düzleştirilmiş toplanır fonksiyonel olarak adlandırılır.

Bu yeterli istatistiklerin hesaplanması gereken en tanıdık durumlardan biri EM algoritması ile EBO parametre kestirmidir: Diyelim ki model matrisleri zamanda değişmemektedir. Bu durumda model parametremiz $\theta = (F, U, G, V)$'dir. $Y_{1:K} = y_{1:K}$ verildiğinde θ için EM algoritması aşağıdaki gibidir:

Algoritma 1. ***GDDU modeli icin EM algoritması***

Başlangıç: $\theta_0 = \left(F^{(0)}, G^{(0)}, U^{(0)}, V^{(0)}\right)$.

$j = 0, 1, \ldots$'inci döngülerde:

- ***B adımı:*** *$m = 0, 1, 2, 3, 4, 5$ için $\widehat{S}^m_K(\theta_j)$'yi hesapla.*
- ***E adımı:*** *$\theta_{j+1} = \left(F^{(j+1)}, G^{(j+1)}, U^{(j+1)}, V^{(j+1)}\right)$'i güncelle:*

$$\theta_{j+1} = \Lambda\left(\left\{\widehat{S}^m_K(\theta_j)\right\}_{m=0,1,2,3,4,5}\right) \tag{3}$$

En büyütme adımındaki $\Lambda(\cdot)$ belli kurallar kümesidir ve bu kurallar [2] ve [3] gibi bir çok çalışmada verilmiştir. EM algoritmasındaki beklenti adımı için en bilinen yöntem olan İDS-GDD yönteminde, verilen $y_{1:K}$'nin bir ileriye, bir de geriye doğru işlenmesi gerekmektedir, bu yüzden de $y_{1:K}$'nin saklanması şarttır. Bir sonraki kısımda, $y_{1:K}$'yı sıralı olarak yalnızca ileri doğru işleyen, bu yüzden de bir döngü içinde k anında $y_{0:k-1}$'in saklanmasına gerek duymayan İDDO yöntemi anlatılacaktır.

3. Yalnızca ileri doğru düzleştirme

Genel bir durum-uzay modelleri şu özelliğe sahiptir: X_k ve $Y_{0:k-1}$ verildiğinde, $X_{0:k-1}$ $\{Y_i\}_{i\geq k}$ ve$\{X_i\}_{i>k}$ dan bağımsızdır. Bu durumda, sonsal dağılım $p_\theta(x_{0:k}|y_{0:k})$'i

$$p_\theta(x_{0:k}|y_{0:k}) = p_\theta(x_k|y_{0:k})p_\theta(x_{0:k-1}|y_{0:k-1}, x_k) \tag{4}$$

şeklinde ayırabiliriz. Aynı özellikten, $p_\theta(x_{0:k-1}|y_{0:k-1}, x_k)$'i

$$p_\theta(x_{0:k-1}|y_{0:k-1}, x_k) = \prod_{i=1}^{k} p_\theta(x_{i-1}|y_{0:i-1}, x_i) \tag{5}$$

şeklinde çarpanlarına ayırabiliriz. Varsayalım ki yeterli istatistikler aşağıdaki gibi toplanır biçimde yazılabilir:

$$S_k(x_{0:k}) = s_0(x_0) + \sum_{i=1}^{k} s_i(x_{i-1}, x_i) \tag{6}$$

Şimdi anahtar T_k fonksiyonunu tanımlayabiliriz: $T_0(x_0, \theta) = s_0(x_0)$ ve $k \geq 1$ için

$$T_k(x_k, \theta) := \int S_k(x_{0:k})p_\theta(x_{0:k-1}|y_{0:k-1}, x_k)dx_{0:k-1}$$

$T_k(x_k, \theta)$ ile $\widehat{S}_k(\theta)$ arasındaki ilişki (4)'den:

$$\widehat{S}_k(\theta) = \int T_k(x_k, \theta)p_\theta(x_k|y_{0:k})dx_k. \tag{7}$$

Son olarak, (5)'i kullanarak $\{T_k\}_{k\geq 1}$'yi hesaplamak için aşağıdaki özyinelemeyi elde ederiz:

$$\begin{aligned} T_k(x_k, \theta) = \int &\left[T_{k-1}(x_{k-1}, \theta) + s_k(x_{k-1}, x_k)\right] \\ &\times p_\theta(x_{k-1}|y_{0:k-1}, x_k)dx_{k-1}. \end{aligned} \tag{8}$$

Bu özyineleme düzleştirilmiş toplanır fonksiyonları elde etmek için kullanıldığından ileri doğru düzleştirme özyinelemesi (İDDÖ) olarak da çağrılır.

Gösterilebilir ki burada anlatılan İDDÖ yöntemi İDS-GDD yöntemine denktir, (ayrıntılar için bknz. [2], Bölüm 4). İDDÖ yöntemine denk olan bir başka özyineleyici yaklaşım ise [6] ve [7] tarafından $\{T_k(x_k, \theta)\}_{k\geq 0}$ yerine $\{T_k(x_k, \theta)p_\theta(x_k|y_{0:k})\}_{k\geq 0}$ için geliştirilmiş ve [3]'ta GDDU modellerine uygulanmıştır.

3.1. İDDÖ'nün GDDU modeline uygulanması ve EM

GDDU modellerinde $p_\theta(x_k|y_{0:k})$ özyinelemeli olarak $p_\theta(x_{k-1}|y_{0:k-1})$'den hesaplanabilir ve her bir $p_\theta(x_k|y_{0:k})$ ortalaması $\mu_{k|k}$ ve ortak değişinti matrisi $\Sigma_{k|k}$ olan Gauss dağılımdır (bknz. [2], Bölüm 5). Bunun sonucunda, İDDÖ için gereken $p_\theta(x_k|y_{0:k}, x_{k+1})$ da Gauss'tur: $I_{n\times n}$ $n \times n$ birim matris olmak üzre, $D_k = \Sigma_{k|k}F_k^T(F_k\Sigma_{k|k}F_k^T + U_k)^{-1}$ ve $d_k = (I_{n\times n} - D_kF_k)\mu_{k|k}$ yazarsak,

$$p_\theta(x_k|y_{0:k}, x_{k+1}) = \mathcal{N}(x_k; \mu_{k|k+1}, \Sigma_{k|k+1})$$
$$\mu_{k|k+1} = D_kx_{k+1} + d_k,\ \Sigma_{k|k+1} = \Sigma_{k|k} - D_kF_k\Sigma_{k|k}$$

Birazdan da göreceğimiz gibi, $p_\theta(x_k|y_{0:k}, x_{k+1})$'lerin ve $p_\theta(x_k|y_{0:k})$'lerin Gauss olması İDDÖ'yü analitik olarak türetebilmemizi sağlayacaktır.

Açıkça görülmektedir ki, GDDU modeli için (1)'de tanımlanan yeterli istatistikler (6)'daki gibi yazılabilmektedir:

$$\begin{aligned} s^1_k(x_k) &= x_kx_k^T,\ s^2_k(x_{k-1}) = x_{k-1}x_{k-1}^T, \\ s^3_k(x_{k-1}, x_k) &= x_{k-1}x_k^T,\ s^4_k(x_k) = x_ky_k^T. \end{aligned}$$

(Yukarıda, $s^0_k(x_k)$ $k = 0$ dışında $s^1_k(x_k)$ ile aynı olduğu için, $\widehat{S}^5_k$ de S^5_k'in kendisi olduğu için bunlara karşılık gelen $s_k(\cdot)$'lar ayrıca yazılmamıştır. Bundan sonra da $S^0_k(x_{1:k})$ ve S^5_k'in kestirimi için düzleştirme işlemleri ayrıca incelenmeyecektir.) Dolayısıyla, bu yeterli istatistiklere karşılık gelen düzleştirilmiş toplanır fonksiyoneller de İDDÖ ile hesaplanabilir. Bu hesaplamanın analitik olarak devam ettirilebilir olması için özyineleme işlemi $s^1_k(x_k)$, $s^2_k(x_{k-1})$ ve $s^3_k(x_{k-1}, x_k)$'nın her bir elemanı için ayrı ayrı, $s^4_k(x_k)$'nın da her bir sütunu için ayrı ayrı yapılmalıdır. Bu durumda, $i, j = 1, \ldots, n$ ve $l = 1, \ldots, r$ için

$$\begin{aligned} s^1_{k,ij}(x_k) &= x_{k,i}x_{k,j},\ s^2_{k,ij}(x_{k-1}) = x_{k-1,i}x_{k-1,j} \\ s^3_{k,ij}(x_k, x_{k-1}) &= x_{k-1,i}x_{k,j},\ s^4_{k,l}(x_k) = x_ky_{k,l} \end{aligned}$$

şeklinde yazarsak, her bir eleman (ve sütun) için özyineleme ayrı ayrı olacaktır. Bu da $m = 1, 2, 3$ için $T^m_{k,ij}(x_k, \theta)$ ve

$T^4_{k,l}(x_k,\theta)$'nın hesaplanması demektir. Örneğin, $T^1_{k,ij}(x_k,\theta)$ için,

$$T^1_{k,ij}(x_k,\theta) = \int \left(T^1_{k-1,ij}(x_{k-1},\theta) + x_{k,i}x_{k,j}\right) \times p_\theta(x_{k-1}|y_{0:k-1},x_k)dx_{k-1}$$

Gösterebiliriz ki, $i,j=1,\ldots,n$ ve $l=1,\ldots,r$ için,

$$\begin{aligned} T^m_{k,ij}(x_k) &= x_k^T A^m_{k,ij}x_k + b^{mT}_{k,ij}x_k + c^m_{k,ij},\ m=1,2,3 \\ T^4_{k,l}(x_k) &= A^4_{k,l}x_k + b^4_{k,l} \end{aligned}$$

ve özyineleme A, b ve c katsayıları üzerinden aşağıdaki gibidir:

$$\begin{aligned}
A^1_{0,ij} &= e_i e_j^T,\ A^2_{0,ij} = A^3_{0,ij} = A^4_{0,l} = 0_{n\times n}; \\
b^1_{0,ij} &= b^2_{0,ij} = b^3_{0,ij} = b^4_{0,l} = 0_{n\times 1}; \\
c^1_{0,ij} &= c^2_{0,ij} = c^3_{0,ij} = 0 \\
A^1_{k+1,ij} &= D_k^T A^1_{k,ij} D_k + e_i e_j^T \\
b^1_{k+1,ij} &= D_k^T b^1_{k,ij} + D_k^T\left(A^1_{k,ij} + A^{1T}_{k,ij}\right)d_k \\
c^1_{k+1,ij} &= c^1_{k,ij} + \mathbf{Tr}\left(A^1_{k,ij}\Sigma_{k|k+1}\right) + b^{1T}_{k,ij}d_k + d_k^T A^1_{k,ij}d_k \\
A^2_{k+1,ij} &= D_k^T\left(A^2_{k,ij} + e_i e_j^T\right)D_k \\
b^2_{k+1,ij} &= D_k^T b^2_{k,ij} + D_k\left(A^2_{k,ij} + A^{2T}_{k,ij} + e_i e_j^T + e_j e_i^T\right)d_k \\
c^2_{k+1,ij} &= c^2_{k,ij} + \mathbf{Tr}\left(\left(A^2_{k,ij} + e_i e_j^T\right)\Sigma_{k|k+1}\right) + b^{2T}_{k,ij}d_k \\
&\quad + d_k^T\left(A^2_{k,ij} + e_i e_j^T\right)d_k \\
A^3_{k+1,ij} &= D_k^T A^3_{k,ij}D_k + e_j e_i^T D_k \\
b^3_{k+1,ij} &= D_k^T b^3_{k,ij} + D_k^T\left(A^3_{k,ij} + A^{3T}_{k,ij}\right)d_k + e_j d_k^T e_i \\
c^3_{k+1,ij} &= c^3_{k,ij} + \mathbf{Tr}\left(A^3_{k,ij}\Sigma_{k|k+1}\right) + b^{3T}_{k,ij}d_k + d_k^T A^3_{k,ij}d_k \\
A^4_{k+1,l} &= A^4_{k,l}D_k + y_{k+1,l}I_{n\times n} \\
b^4_{k+1,l} &= b^4_{k,l} + A^4_{k,l}d_k
\end{aligned} \tag{9}$$

Yukarıda e_i uygun büyüklükteki birim matrisin i'inci sütunudur. O halde, düzleştirilmiş toplanır fonksiyonelleri aşağıdaki gibi hesaplayabiliriz: $m = 0,1,2,3,5$; $i,j=1,\ldots,n$ ve $l=1,\ldots,r$ için,

$$\begin{aligned}
\widehat{S}^m_{k,ij}(\theta) &= \left[\widehat{S}^m_k(\theta)\right]_{ij} = \int T^m_{k,ij}(x_k,\theta)p_\theta(x_k|y_{0:k})dx_k \\
&= \mathbf{Tr}\left(A^m_{k,ij}\left(\Sigma_{k|k} + \mu_{k|k}\mu^T_{k|k}\right)\right) + b^{mT}_{k,ij}\mu_{k|k} + c^m_{k,ij} \\
\widehat{S}^4_{k,l}(\theta) &= \left[\widehat{S}^4_k(\theta)\right]_{1:n,l} = \int T^4_{k,l}(x_k,\theta)p_\theta(x_k|y_{0:k})dx_k \\
&= A^4_{k,l}\mu_{k|k} + b^4_{k,l}
\end{aligned}$$

İDDÖ'nün zaman başi hesaplama yükü matrislerin her elemanı için $n\times n$ matrislerin tersinin alınması işlemini gerektirdiğinden $\mathcal{O}(n^5)$'ken, İDS-GDD için ise $\mathcal{O}(n^3)$'tür. Ancak, İDDÖ'de matrislerin elemanları için yapılan özyinelemeler birbirinden bağımsız olarak gerçekleştirilebildiğinden, yöntem paralel hesaplamalara uygundur. (Başka bir deyişle n^2 tane $\mathcal{O}(n^3)$'lük hesaplama paralel yapılabilir.)

3.2. Anında EM algoritması

Aninda EM algoritmasindaki fikir yeni bir gözlem geldiğinde parametre kestirimi güncellenmesidir. Önerilen anında kestirim yöntemi, İDDÖ'nün stokastik bir yaklaşıklamasına dayanmaktadır: Varsayalım ki $\{\theta_i\}_{0\le i<k}$ anında EM algoritmasının $y_{0:k-1}$'e dayanarak $k-1$ anına kadar yaptığı kestirimlerin dizisi olsun. k aninda, İDDÖ'ye şu şekilde yakınsanır:

$$T_k(x_k,\theta_{0:k-1}) = \int \Big[(1-\eta_k)T_{k-1}(x_{k-1},\theta_{0:k-2}) + \eta_k s_k(x_{k-1},x_k)\Big] p_{\theta_{0:k-1}}(x_{k-1}|y_{0:k-1},x_k)dx_{k-1} \tag{10}$$

Burada $\{\eta_k\}_{k\ge0}$ $\sum_{k=0}\eta_k=\infty$ ve $\sum_{k=0}\eta_k^2<\infty$ koşullarını sağlayan artı sayılardan oluşan bir dizidir. Örneğin, $0.5<\eta\le 1$ için $\eta_k = k^{-\eta}$ alınabilir.

EM algoritmasının en büyütme adımı her k anında yapılır. (Ancak, uygulamada kestirimin güncellenmesinin sayısal olarak iyi huylu olmasını garantilemek için bu adım belli bir sayıda (k_b) döngü boyunca atlanır.) Bu şekilde, algoritmanın çok uzun bir veri dizisi verildiğinde bu verinin üzerinden sadece bir kez geçerek EBO kestirimine ulaşması beklenir.

Bu yaklaşım için elde edilen özyineleme (9)'deki özyineleme kurallarına benzerdir; sadece eşitliklerde $T_{k-1}(\cdot)$'ye karşılık gelen terimler $1-\eta_k$ ile çarpılırken $s_k(x_{k-1},x_k)$ ile ilgili terimler ise η_k ile çarpılmalıdır. Anında EM algoritması asağıda verilmiştir.

Algoritma 2. ***GDDU modeli için anında EM***

Baslangic: Rastgele bir θ_0 ile basla.

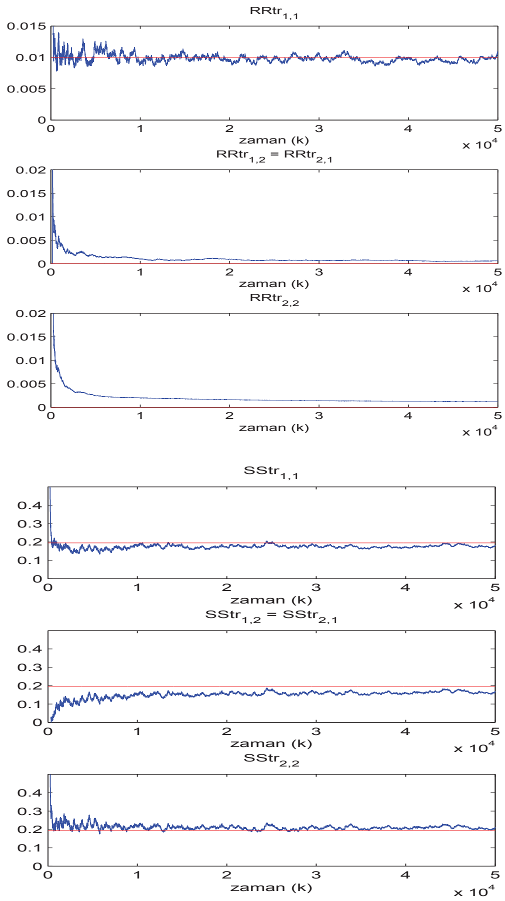
$k=0$ icin

- ***B adımı:***
  - *$i,j=1,\ldots,n$ ve $l=1,\ldots,r$ için, $\left(\left\{A^m_{0,ij},b^m_{0,ij},c^m_{0,ij}\right\}_{m=0,1,2,3}, A^4_{0,l}, b^4_{0,l}\right)$. Ayrıca, $\widehat{S}^5_0 = \eta_0 y_0 y_0^T$*
  - *θ_0'i kullanarak $\mu_{0|0}$, $\Sigma_{0|0}$, D_0, d_0 ve $\Sigma_{0|1}$'i hesapla.*

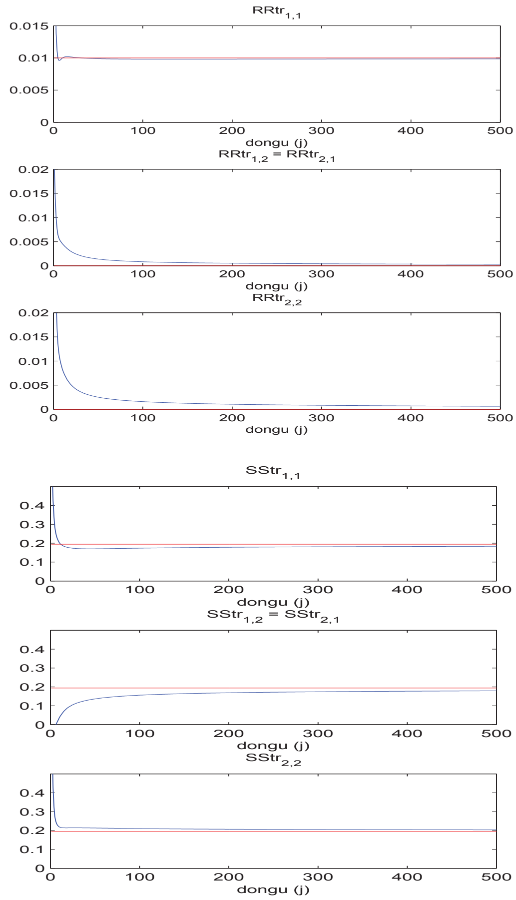
$1\le k$ için

- ***B adımı:***
  - *$\left(\theta_{k-1},\mu_{k-1|k-1},\Sigma_{k-1|k-1}\right)$'i kullanarak $\left(D_{k-1},d_{k-1},\Sigma_{k-1|k}\right)$'i hesapla.*
  - *$i,j=1,\ldots,n$ ve $l=1,\ldots,r$ icin, $(\{A^m_{k,ij},b^m_{k,ij},c^m_{k,ij}\}_{m=0,1,2,3}, A^4_{k,l}, b^4_{k,l})$ katsayılarını bir önceki değerlerini ve $\left(D_{k-1},d_{k-1},\Sigma_{k-1|k},\theta_{k-1}\right)$'i kullanarak güncelle. Ayrıca, $\widehat{S}^5_k = (1-\eta_k)\widehat{S}^5_{k-1} + \eta_k y_k y_k^T$.*
  - *$\left(\mu_{k|k},\Sigma_{k|k}\right)$'yi $\left(\mu_{k-1|k-1},\Sigma_{k-1|k-1}\right)$ ve θ_{k-1} kullanarak hesapla.*
- ***E adımı:*** *Eğer $k<k_b$ ise, $\theta_k=\theta_{k-1}$. Değilse; $i,j=1,\ldots,n$ ve $l=1,\ldots,r$ icin $\left\{\widehat{S}^m_k(\theta_{0:k-1})\right\}_{m=0,1,2,3,4}$'i $\left(\mu_{k|k},\Sigma_{k|k}\right)$ ve $(\{A^m_{k,ij},b^m_{k,ij},c^m_{k,ij}\}_{m=0,1,2,3}, A^4_{k,l}, b^4_{k,l})$ kullanarak hesapla, $\theta_k = \left(F^{(k)},G^{(k)},U^{(k)},V^{(k)}\right)$'yi güncelle:*

$$\theta_k = \Lambda\left(\left\{\widehat{S}^m_k(\theta_{0:k-1})\right\}_{m=0,1,2,3,4,5}\right) \tag{11}$$



Şekil 1: Aninda EM algoritmasi kestirimleri



Şekil 2: EM algoritmasi kestirimleri

4. Deneyler ve Sonuçlar

Bu kısımda, EM ve anında EM algoritmalarının başarımlarını benzetimlenmiş veri kullanarak karşılaştırdık. Durum ve uzay vektörlerinin boyutları $n = 2$ ve $r = 2$ olarak alındı. $K = 50000$ uzunluğunda bir veri dizisi üretildi. Deneylerimizde U ve V bilinmiyor varsayılarak kestirilmeye calışılmıstır. Şekil 1 ve 2, EM ve anında EM algoritmasının sonuclarını göstermektedir. Düz çizgiler gerçek değerlerdir. Görüldüğü gibi, iki algoritma da yaklaşık aynı değerlere yakınsamıştır. Ancak, anında EM algoritması bunu gözlemlenen verinin üzerinden sadece bir kez geçerek yaparken, EM algoritması için 400'den fazla döngü gerekmiştir. Buradan da anında EM'nin 400 kat daha hızlı yakınsadığını söyleyebiliriz.

5. Tartisma

Bu çalışmada görmüs olduk ki, GDDU parametre kestirimi için gereken yeteri istatistikler sadece ileri-doğru bir sekilde kestirilebilir. Bunu mümkün kılan IDDÖ aynı zamanda anında cok uzun veri dizileri icin yararlı olabilecek anında EM algoritmasının yolunu açmaktadır. Ayrıca, $O(n^5)$'lik bu yöntem paralel hesaplamalara uygun olacağından $O(n^3)$'luk bir yöntemmiş gibi gercekleştirilebilir.

6. KAYNAKÇA

[1] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[2] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, Springer, 2005.

[3] R.J. Elliott and V. Krishnamurthy, "New finite-dimensional filters for parameter estimation of discrete-time linear Gaussian models," *Automatic Control, IEEE Transactions on*, vol. 44, no. 5, pp. 938 –951, may. 1999.

[4] G. Mongillo and S. Deneve, "Online learning with hidden Markov models," *Neural Computation*, vol. 20, no. 7, pp. 1706–1716, 2008.

[5] O Cappé, "Online EM algorithm for hidden Markov models," http://arxiv.org/pdf/0908.2359, August 2009.

[6] O. Zeitouni and A. Dembo, "Exact filters for the estimation of the number of transitions of finite-state continuous-time Markov processes," *IEEE Trans. Inform. Theory*, vol. 34, no. 4, pp. 890 – 893, 1988.

[7] R.J. Elliott, "New finite dimensional filters and smoothers for Markov chains observed in Gaussian noise," *IEEE Trans. Signal Process*, vol. 39, pp. 265 –271, may. 1993.