

Bağlam Ağaçları ile Ardışık Doğrusal Olmayan Bağlanım

Sequential Nonlinear Regression via Context Trees

N. Denizcan Vanlı, Süleyman S. Kozat
Elektrik ve Elektronik Mühendisliği Bölümü
Bilkent Üniversitesi
{vanli, kozat}@ee.bilkent.edu.tr

Özetçe —Bu bildiride, ardışık doğrusal olmayan bağlanım problemi incelenmiş ve bağlam ağaçları kullanarak etkili bir öğrenme algoritması sunulmuştur. Bu amaçla, bağlanım alanı parçalara ayrılmış ve oluşan bölgeler bağlam ağacı ile simgelenmiştir. Her bölgede bağımsız bağlanım algoritmaları kullanılarak bağlam ağacı tarafından gösterilebilen tüm doğrusal olmayan modellerin kestirimleri, hesaplama karmaşıklığı bağlam ağacının düğüm sayısı ile doğrusal olan bu algoritma ile uyarlanır olarak birleştirilmiştir. Önerilen algoritmanın performans limitleri, veriler üzerinde istatistiksel varsayımlarda bulunmaksızın incelenmiştir. Ayrıca, teorik sonuçları izah etmek için sayısal bir örnek sunulmuştur.

Anahtar Kelimeler—ardışık, uyarlanır, doğrusal olmayan bağlanım, bağlam ağacı.

Abstract—In this paper, we consider the problem of sequential nonlinear regression and introduce an efficient learning algorithm using context trees. Specifically, the regressor space is partitioned and the resulting regions are represented by a context tree. In each region, we assign an independent regression algorithm and the outputs of the all possible nonlinear models defined on the context tree are adaptively combined with a computational complexity linear in the number of nodes. The upper bounds on the performance of the algorithm are also investigated without making any statistical assumptions on the data. A numerical example is provided to illustrate the theoretical results.

Keywords—sequential, adaptive, nonlinear regression, context tree.

I. GİRİŞ

Doğrusal bağlanım modelleri uzun zamandır sinyal işleme ve otomatik öğrenme literatürlerinin temel temalarından biri olmuştur [1]. Ancak, işleme gücünün artması ile verilerin belirlen özelliklerini öğrenebilmek için doğrusal olmayan bağlanım yöntemleri sıkça uygulanmaya başlanmıştır [2]–[4]. Doğrusal olmayan yöntemler doğrusal yöntemlere göre daha güçlü modelleme performansı gösterse de bu yöntemlerin uygulamaları hesaplama karmaşıklığı ve kararlılık problemleri gibi sebeplerden ötürü güçtür [5]. Bu sorunları en aza indirmek için bağlam ağaçları veya karar ağaçları gibi etkili simgeleme yöntemleri kullanılmaktadır [6].

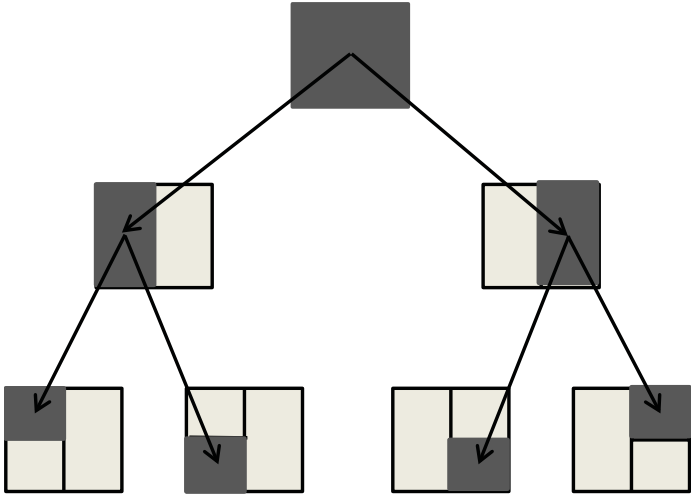
Bir bağlam ağacı, bağlanım alanının iç içe bölümlerini simgeler. Örnek olarak, Şekil 1’deki 2 derinlikli bağlam ağacı,

iki boyutlu bağlanım düzlemini böler. Bağlam ağacının üstündeki her düğüm, bağlanım alanının bir bölgesini simgeler. Bu iç içe bölüm tanımlandıktan sonra, her düğümdeki bağlanım yapıları birbirlerinden bağımsız olarak seçilebilir. Bu yöntem ile doğrusal olmayan bağlanım, doğrusal bağlanım yöntemlerinin dolaysız uzantısı olarak görülebilir. d derinlikli bir bağlam ağacının üzerinde, toplamda yaklaşık olarak 1.5^{2d} farklı doğrusal olmayan model tanımlıdır [7]. Örnek olarak, Şekil 1’deki 2 derinlikli bağlam ağacı, Şekil 2’deki gibi 5 farklı doğrusal olmayan modeli simgeler.

Bağlam ağacı üstünde tanımlı olan çifte üstel farklı doğrusal olmayan modeller içinden herhangi bir tanesini, veri işlemeye başlamadan önce seçip, sadece bu modelin parametrelerini uyarlamak bağlanım algoritmasının performansını önemli ölçüde kısıtlar. Bu sebepten ötürü, bu bildiride, sadece tek bir modele bağlanmak yerine, bu modellerin kestirimleri uyarlanır bir algoritma ile birleştirilmiştir. Bu sayede, yüksek doğrusal olmayan modelleme gücüne ulaşılmıştır.

Literatürdeki benzer yöntemler, maliyet fonksiyonunu küçültecek birleştirme parametrelerini aramak yerine, evrensel birleştirme katsayıları tanımlayarak tüm çifte üstel modeller içinden en iyisinin performansına ulaşmaya çalışmışlardır [2]. Fakat bu yöntemler, veriler üstünde işleme başlamadan önsel bilgiler (üst sınır gibi) gerektirmektedir. Ayrıca, bu yöntemlerin başarılı sonuçlar vermesi algoritma parametrelerinin ince ayarlanmasıyla mümkündür. Öte yandan, bu bildiride önerilen algoritma, evrensel birleştirme katsayıları kullanmadan uyarlanabilir bir yöntem yardımıyla doğrudan bağlanım hatasını küçültme esasına dayanır. Bu sayede, önerilen yöntem, bilinmeyen ve istenen veri üzerinde herhangi bir önsel bilgiye ihtiyaç duymaz ve gerçek hayat problemlerinde rahatlıkla kullanılabilir. Dolayısıyla, bildiride sunulan problem tanımı ve önerilen yöntemler literatürdeki yöntemlere göre oldukça farklılık göstermektedir.

Bu bildiride önerilen algoritma, *i)* en gelişkin teknoloji yöntemlere göre performans artışı sağlamaktadır, *ii)* veriler üzerinde önsel bilgi gerektirmemektedir, *iii)* performans üst sınırları hiçbir istatistiksel varsayım gerekmeksizin sağlanmaktadır, ve *iv)* hesaplama karmaşıklığı sadece bağlam ağacının düğüm sayısı ile doğrusaldır. Önerilen algoritma bu performans, tüm çifte üstel sayıdaki modellerin kestirimlerinin en iyi doğrusal katışımına yakınsayarak ulaşmaktadır. Bu sayede, evrensel birleştirme parametreleri kullanmak yerine, önerilen algoritma doğrudan bağlanım hatasını en küçültmeye çalış-



Şekil 1: 2 boyutlu bağlanım alanının 2 derinlikli bağlam ağacı ile bölünmesi. Her bir düğümde bağlanım alanı birer defa sıradüzensel olarak parçalanmıştır.

maktadır. Ayrıca, önerilen algoritma, literatürdeki benzer ağaç kullanan algoritmalarla birleştirilebilir ve farklı yöntemleri kapsayacak şekilde genişletilebilir.

II. PROBLEM TANIMI

Bu bildiride, ardışık doğrusal olmayan bağlanım problemi incelenmektedir. İstenen sinyal $\{x[t]\}_{t \geq 1}$, $x[t] \in \mathbb{R}$ ile gösterilirken, bağlanım vektörleri $\{\mathbf{y}[t]\}_{t \geq 1}$, $\mathbf{y}[t] \in \mathbb{R}^m$ ile gösterilmektedir. Bu çerçevede t anındaki istenen sinyal aşağıdaki şekilde kestirilmektedir:

$$\hat{x}[t] = f_t(\mathbf{y}[t]).$$

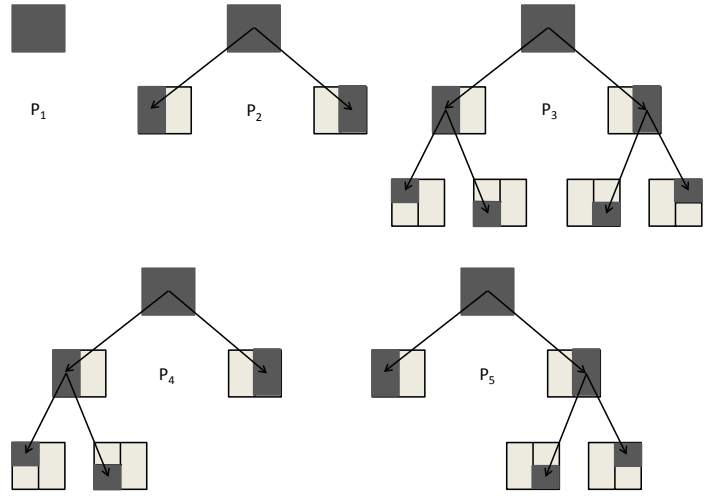
Burada, $f_t(\cdot)$ bir uyarlanabilir doğrusal olmayan bağlanım fonksiyonudur. Her t anında, bağlanım hatası ise

$$e[t] = x[t] - \hat{x}[t]$$

şeklinde hesaplanmaktadır.

Uyarlanabilir doğrusal bağlanım fonksiyonu çeşitli şekillerde seçilebilirken, bu bildiride, parçalı doğrusal fonksiyonlar kullanılmıştır. Bu amaçla, bağlanım alanı veri işleme başlamadan önce, hiperdüzlemler kullanılarak parçalanmıştır. Bu parçalama işlemi sonucu oluşan her bölgeye bir doğrusal bağlayıcı atanmıştır. Bu seçimler bildirinin anlaşılabilirliğini artırmak için tercih edilmiştir. Önerilen yöntem, hiperdüzlemler yerine farklı ayıraç fonksiyonları ve/veya her bölgedeki doğrusal bağlayıcılar yerine literatürdeki doğrusal olmayan bağlanım yöntemleri kullanılarak da genelleştirilebilir.

Probleme açıklık kazandırmak için şu örnek düşünülebilir. Şekil 1'deki 2 derinlikli bağlam ağacı üç farklı ayıraç fonksiyonu kullanılarak oluşturulmuştur. İkili bağlam ağacı kullanımından dolayı bu ayıraç fonksiyonları bağlanım alanı üzerinde 4 farklı bölge oluşturmaktadır. Bağlam ağacı üzerindeki bölüm işlemi sıradüzensel şekilde tanımlıdır. Dolayısıyla, t anındaki bağlanım vektörü $\mathbf{y}[t]$ önce kök düğümdeki ayıraç fonksiyonu tarafından işlenmektedir, ve daha sonra kök düğümün alt düğümlerinden biri tarafından işlenir. Bu



Şekil 2: Şekil 1'deki 2 derinlikli bağlam ağacı tarafından bölünen bağlanım alanı üzerinde tanımlanan tüm çiftte üstel sayıdaki modeller.

çerçevede, ayıraç fonksiyonları kesin kararlar vermektedir. Eğer bağlanım vektörü $\mathbf{y}[t]$, $\mathcal{R}$ ile gösterilen bölgeye düşerse ayıraç fonksiyonunun değeri 1'dir, aksi takdirde bu değer 0 olarak bulunur.

Bağlam ağacı üzerinde tanımlı olan herhangi bir doğrusal olmayan model tüm bağlanım alanını kapsayacak şekilde parçalara ayrılmıştır. d derinlikli bağlam ağacı tarafından gösterilen tüm farklı doğrusal olmayan modellerin sayısını $f(d) \approx 1.5^{2^d}$ ile gösterelim. Bu durumda, d derinlikli bağlam ağacı üstünde tanımlı olan herhangi bir $i = 1, \dots, f(d)$ modeli için bu modelin yaprak düğümleri ile gösterilen bölgelerin toplamı bağlanım alanına eşittir:

$$\bigcup_{k=1}^{l(i)} \mathcal{R}_{i,k} = \mathbb{R}^m.$$

Burada, $l(i)$, i modelindeki yaprak düğüm sayısını ve $\mathcal{R}_{i,k}$ ise i modelinin k bölgesini göstermektedir.

Şekil 2'deki üçüncü doğrusal olmayan modeli, P_3 , ele alırsak, bu model 4 farklı bölgenin (4 yaprak düğüme karşılık gelen bölgelerin) birleşimi ile oluşturulmuştur. Bu model üzerindeki her bölge için birer doğrusal bağlanım algoritması tanımlanmıştır. Başka bir deyişle, tüm $k = 1, \dots, l(3)$ için her $\mathcal{R}_{i,k}$ bölgesinde

$$\hat{x}_{i,k}[t] = \mathbf{w}_{i,k}^T[t] \mathbf{y}[t]$$

kestirimi yapılmaktadır. Burada, $\mathbf{w}_{i,k}[t] \in \mathbb{R}^m$, i modelinin k bölgesine atanmış olan t anındaki doğrusal bağlanım vektörüdür. Yukarıda tanımlanan sıradüzensel yöntemle göre, P_3 modelinin kestirimi

$$\hat{x}_3[t] = \hat{x}_{3,1}[t]I_{3,1}[t] + \hat{x}_{3,2}[t]I_{3,2}[t] + \hat{x}_{3,3}[t]I_{3,3}[t] + \hat{x}_{3,4}[t]I_{3,4}[t]$$

şeklinde hesaplanır. Burada, $I_{i,k}[t]$, P_3 modelindeki k düğümü için t anındaki ayıraç (gösterge) fonksiyonunun değeridir. Bu fonksiyonun değeri eğer t anındaki bağlanım vektörü k düğümü ile gösterilen bölgeye düşmüşse 1, aksi takdirde 0 olarak gösterilmektedir. Başka bir deyişle,

$$I_{i,k}[t] = \begin{cases} 1 & , \text{ if } \mathbf{y}[t] \in \mathcal{R}_{i,k} \\ 0 & , \text{ if } \mathbf{y}[t] \notin \mathcal{R}_{i,k} \end{cases}. \quad (1)$$

Dolayısıyla, her doğrusal olmayan model için o modelin t anındaki kestirimi bağlanım vektörünün düştüğü yaprak düğümün kestirimine eşittir.

Yukarıda belirtilen yöntem ile tüm çifte üstel sayıdaki modellerin kestirimleri $\hat{x}_1[t], \dots, \hat{x}_{f(d)}[t]$ hesaplanabilir. Bu kestirimleri doğrusal olarak birleştirerek, katışım katsayıları eniyileştirilebilir. Bu doğrultuda, önerilen algoritmanın kestirimi

$$\begin{aligned}\hat{x}[t] &\triangleq \sum_{i=1}^{f(d)} \pi_i[t] \hat{x}_i[t] \\ &= \boldsymbol{\pi}[t]^T \hat{\mathbf{x}}[t]\end{aligned}\quad (2)$$

şeklinde hesaplanmaktadır. Burada, $\hat{\mathbf{x}}_t \triangleq [\hat{x}_1[t], \dots, \hat{x}_{f(d)}[t]]^T$ olarak tanımlanmıştır.

III. BAĞLAM AĞAÇLARI İLE BİRLEŞTİRİM

Bu bölümde, (2)'de tanımlanan ve tüm çifte üstel sayıdaki ardışık bağlanım fonksiyonlarını uyarlanır olarak birleştiren algoritma sunulmaktadır.

Teorem 1: $\{x[t]\}_{t \geq 1}$ ve $\{y[t]\}_{t \geq 1}$ genişgüzel gerçel değerli seriler olsun. *Algorithm 1*'de verilen $\hat{x}[t]$ bağlanım algoritması, genişgüzel uzunluktaki $n \geq 1$ herhangi serilere uygulandığında aşağıdaki performans üst sınırına ulaşır:

$$\sum_{t=1}^n (x[t] - \hat{x}[t])^2 - \min_{\boldsymbol{\pi} \in \mathbb{R}^{f(d)}} \sum_{t=1}^n (x[t] - \boldsymbol{\pi}^T \hat{\mathbf{x}}[t])^2 \leq O(\ln(n)).$$

Bu teorem, önerilen algoritmanın performansının, $f(d)$ sayıda farklı modelin en iyi birleştirmesinin performansına ulaştığını göstermektedir. Ayrıca, önerilen algoritma $O(1.5^{2^d})$ farklı modeli birleştirirken, algoritmanın hesaplama karmaşıklığı yalnızca $O(d2^d)$ 'dir. Daha önce de belirtildiği gibi bu performansa ulaşmak için önerilen algoritmanın veriler üzerinde herhangi bir istatistiksel varsayıma veya önsel bilgiye ihtiyacı yoktur.

Algorithm 1 Bağlam Ağaçları ile Uyarlanır Bağlanım

```

1: for  $t = 1$  to  $n$  do
2:    $\mathcal{U}[t] \leftarrow c = 1, \dots, g(d) : y[t] \in \mathcal{R}_c$ 
3:    $\hat{x}[t] \leftarrow 0$ 
4:   for all  $c_1 \in \mathcal{U}[t]$  do
5:      $\hat{x}_{c_1}[t] \leftarrow w_{c_1}[t]^T y[t]$ 
6:     for all  $c_2 \in \mathcal{D}_d$  do
7:        $\hat{x}[t] \leftarrow \hat{x}[t] + w_{c_1, c_2} \lambda_{c_2}[t] \hat{x}_{c_1}[t]$ 
8:     end for
9:   end for
10:   $e[t] \leftarrow x[t] - \hat{x}[t]$ 
11:  for all  $c \in \mathcal{U}[t]$  do
12:     $w_c[t+1] \leftarrow w_c[t] + \mu[t] e[t] y[t]$ 
13:     $\lambda_c[t+1] \leftarrow \lambda_c[t] + \mu[t] e[t] \hat{x}_c[t]$ 
14:  end for
15: end for
```

A. Teorem 1'in İspatı ve Algoritmanın Kurgulanması

Algoritmanın kurgusunda olasılıksal bayır inişi (SGD) algoritması kullanıldığı için, algoritmanın performans üst sınırı [8]'den bulunabilir. Algoritma kurgusu ana hatlarıyla aşağıda sunulmaktadır.

Katışım parametreleri olasılıksal bayır inişi algoritması ile güncellendiği için herhangi bir $i = 1, \dots, f(d)$ modelinin katışım parametresini, aynı modelin düğümlerinin birleşim parametrelerinin toplamı cinsinden ifade edebiliriz:

$$\pi_i[t] = \sum_{k=1}^{l(i)} \pi_{i,k}[t].$$

Dolayısıyla, $\mathcal{D}_d$, d derinlikli bir bağlam ağaç üzerindeki tüm düğümleri simgeliyorken, herhangi bir $c \in \mathcal{D}_d$ düğümünün t anındaki birleşim parametresini ve kestirimini sırasıyla $\lambda_c[t]$ ve $\hat{x}_c[t]$ olarak tanımlarsak, $\lambda_c[t]$ üzerinde aşağıdaki olasılıksal bayır inişi güncellemesine ulaşırız:

$$\lambda_c[t+1] \triangleq \lambda_c[t] + \mu[t] e[t] \hat{x}_c[t].$$

Burada, gösterim yükünden kurtulmak için yukarıdaki tanımlamalar yapılmıştır. Herhangi bir i modeline ait herhangi bir k düğümü için $\pi_{i,k}[t] = \lambda_k[t]$ ve $\hat{x}_{i,k}[t] = \hat{x}_k[t]$ eşitlikleri sağlanmaktadır.

Benzer şekilde, tüm modellerin kestirimlerini de bağlam ağaç üzerinde tanımlı olan düğümlerin kestirimleri cinsinden yazalım. (1)'deki tanıma göre, her doğrusal olmayan model için o modelin t anındaki kestirimi bağlanım vektörünün düştüğü yaprak düğümün kestirimine eşittir. Başka bir deyişle, d derinlikli bağlam ağaç üzerinde tanımlı olan i modeli için, t anında bağlanım vektörünün düştüğü bölgeyi gösteren bir fonksiyon tanımlayalım: $r_i[t] = k$, eğer $y[t] \in \mathcal{R}_{i,k}$ ise. Bu gösterime göre her i modelinin t anındaki kestirimi

$$\hat{x}_i[t] = \hat{x}_{i, r_i[t]}[t]$$

olarak bulunur.

Bu durumda, önerilen algoritmanın kestirimi

$$\hat{x}[t] = \sum_{i=1}^{f(d)} \left(\sum_{k=1}^{l(i)} \pi_{i,k}[t] \right) \hat{x}_{i, r_i[t]}[t] \quad (3)$$

olarak bulunur. (3)'teki eşitlikte şu gözlemler yapılabilir: Herhangi bir t anında $r_i[t]$ fonksiyonu sadece $y[t]$ 'yi içeren yaprak düğüm ve bu düğümün üst düğümlerini gösterebilir. Dolayısıyla, tüm bu düğümlerin kümesini $\mathcal{U}[t]$ ile gösterirsek, $|\mathcal{U}[t]| = d + 1$ olarak bulunur. Diğer bir gözlem ise şu şekildedir: (3)'teki eşitlik bağlam ağaç üzerinde tanımlı olan tüm modeller üzerinden toplanarak bulunmuştur. Ancak bu hesaplama karmaşıklığını oldukça artırmaktadır. Herhangi bir $r_i[t]$ düğümü için $\hat{x}_{i, r_i[t]}[t]$ kestiriminin katışım katsayısında yine herhangi bir $c \in \mathcal{D}_d$ için $\lambda_c[t]$ katsayısı c ve $r_i[t]$ 'nin aynı anda yaprak düğüm olduğu model sayısı ile çarpılmaktadır.

Herhangi bir $c \in \mathcal{D}_d$ düğümü için bu düğümün derinliğini $\delta(c)$ ile gösterelim. Bu durumda, c düğümünün d derinlikli bir bağlam ağaçta yaprak düğüm olduğu model sayısı

$$h_d(c) \triangleq \prod_{j=1}^{\delta(c)} f(d-j)$$

ile bulunur. $h_d(c)$ kullanılarak, herhangi iki c_1, c_2 düğümü için iki düğümün de yaprak düğüm olarak bulunduğu model sayısı [8]'de detaylıca gösterildiği gibi hesaplanabilir. Bu değeri $\omega(c_1, c_2)$ ile simgelersek, (3)'teki kestirim

$$\hat{x}[t] = \sum_{c_1 \in \mathcal{U}[t]} \left\{ \sum_{c_2 \in \mathcal{D}_d} \omega(c_1, c_2) \pi_{c_2}[t] \right\} \hat{\chi}_{c_1}[t] \quad (4)$$

olarak bulunur. $\omega(c_1, c_2)$ 'in değeri zaman ile değişmediğinden ötürü, bu değerler veri işleme başlamadan önce hesaplanabilir, dolayısıyla hesaplama karmaşıklığını artırmaz. Bu sebeple, (4)'teki işlemin hesaplama karmaşıklığı $O(d2^d)$ 'dir. Böylece algoritmanın kurgusu tamamlanır. $\square$

IV. SAYISAL ÖRNEKLER

Bu bölümde, önerilen algoritmanın performansı en gelişkin teknoloji algoritmaların performansı ile karşılaştırılmıştır. Bu amaçla, aşağıdaki parçalı doğrusal model düşünülmüştür:

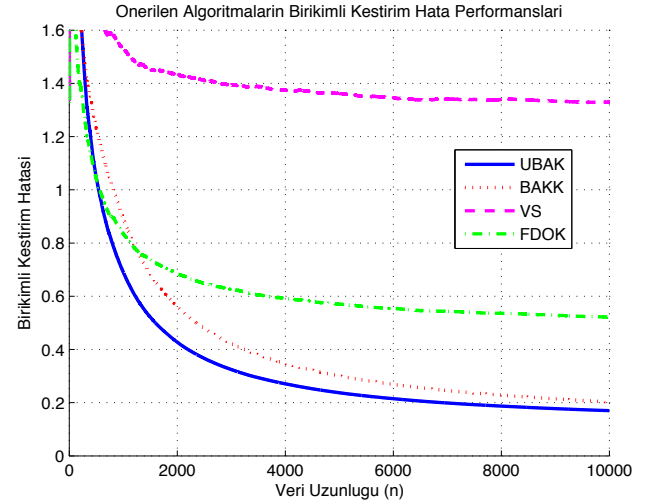
$$x_t = \begin{cases} y_1[t] + y_2[t] + \psi[t], & y_1[t] \geq 0 \text{ ve } y_2[t] \geq 0 \\ -y_1[t] - y_2[t] + \psi[t], & y_1[t] \geq 0 \text{ ve } y_2[t] < 0 \\ -y_1[t] - y_2[t] + \psi[t], & y_1[t] < 0 \text{ ve } y_2[t] \geq 0 \\ y_1[t] + y_2[t] + \psi[t], & y_1[t] < 0 \text{ ve } y_2[t] < 0 \end{cases} \quad (5)$$

Burada, $\psi[t]$ sıfır ortalamalı ve 0.1 değişintili beyaz Gaussian sürecinden alınmış örnek değerlerdir, $y_1[t]$ ve $y_2[t]$ sıfır ortalama vektörlü ve I_2 değişinti matrisli birleşik Gaussian sürecinden alınmış örnek değerlerdir. t anındaki istenen veri $x[t]$ ile, bağlanım vektörü ise $y[t] = [y_1[t], y_2[t]]^T$ ile gösterilmiştir.

Şekil 3'te önerilen algoritmanın performansı en gelişkin teknoloji algoritmaların performansı ile karşılaştırılabilir. Şekilde "UBAK" bu bildiride önerilen ve 2 derinliği olan uyarlanırlı bağlanım ağaç kestirim algoritmasını, "BAKK" [2]'de önerilen ve 2 derinliği olan bağlanım ağaç katışım kestirim algoritmasını, "VS" ikinci dereceden Volterra süzgecini [5], ve son olarak "FDOK" ise [3]'te önerilen üçüncü dereceden Fourier doğrusal olmayan kestirim algoritmasını göstermektedir.

Bu örnekte, öğrenme hızları UBKK, BAKK ve FDOK algoritmaları için 0.005, VF algoritması için ise 0.05 olarak seçilmiştir. Bu değerler, algoritmaların benzer yaklaşım performansı göstermesi için seçilmiştir.

Şekil 3'te görüldüğü üzere, önerilen algoritma literatürdeki en gelişkin teknoloji algoritmalara göre gözlenebilir bir performans artışı sağlamaktadır. Bu performans artışı literatürdeki evrensel katışım parametreleri kullanımı yerine en iyi katışım parametrelerinin uyarlanırlı olarak güncellenmesiyle ulaşılmıştır. Ek olarak, önerilen algoritma veriler üzerinde herhangi bir önsel bilgi gerektirmemektedir. Bu sayede, gerçek hayat problemlerine uygulanabilirliği oldukça yüksektir. Ayrıca, önerilen algoritma doğrusal olmayan bağlanıma, bağlanım alanını parçalayarak ulaşmaktadır. Dolayısıyla, algoritmanın hesaplama karmaşıklığı bağlanım vektörünün boyutuyla doğrusal olarak artmaktadır. Bu sayede, önerilen algoritma büyük veri problemleri gibi yüksek boyutlu işlemlerde kullanılabilir.



Şekil 3: (5)'teki parçalı doğrusal model için önerilen algoritmaların birikimli bağlanım hata performansları. Şekildeki değerler 10 bağımsız denemenin ortalaması alınarak bulunmuştur.

V. SONUÇLAR

Bu bildiride ardışık doğrusal olmayan bağlanım problemi incelenmiş ve uyarlanırlı parçalı doğrusal süzgeçleme tekniği kullanılarak bağlanım hatası performansı eniyileştirilmiştir. Parçalı doğrusal fonksiyonların kullanımı bağlanım alanının hiperdüzlemler ile parçalanması ile sağlanmıştır. Düşük hesaplama karmaşıklığına sahip bir algoritmaya ulaşmak için bağlanım alanının parçalanması sonucu oluşan bölgeler bağlanım ağaçları ile simgelenmiştir. Önerilen algoritma ile bağlanım ağacı üzerinde tanımlı olan çifte üstel sayıdaki farklı doğrusal olmayan modellerin kestirimlerinin katışımı için gereken hesaplama karmaşıklığının bağlanım ağacındaki düğüm sayısı ile doğrusal olarak arttığı gösterilmiştir. Ayrıca, önerilen algoritmanın performansının en gelişmiş teknoloji algoritmalarından daha iyi olabileceği gözlemlenmiştir.

KAYNAKÇA

- [1] A. C. Singer, S. S. Kozat, and M. Feder, "Universal linear least squares prediction: upper and lower bounds," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2354–2362, 2002.
- [2] S. S. Kozat, A. C. Singer, and G. C. Zeitler, "Universal piecewise linear prediction via context trees," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3730–3745, 2007.
- [3] A. Carini and G. L. Sicuranza, "Fourier nonlinear filters," *Signal Processing*, vol. 94, no. 0, pp. 183 – 194, 2014.
- [4] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Nonlinear spline adaptive filtering," *Signal Processing*, vol. 93, no. 4, pp. 772 – 783, 2013.
- [5] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. NJ: John Wiley & Sons, 1980.
- [6] J. Gama, "Functional trees," *Machine Learning*, vol. 55, no. 3, pp. 219–250, 2004.
- [7] A. V. Aho and N. J. A. Sloane, "Some doubly exponential sequences," *Fibonacci Quarterly*, vol. 11, pp. 429–437, 1970.
- [8] N. D. Vanli and S. S. Kozat, "A comprehensive approach to universal nonlinear regression based on trees," *submitted to IEEE Transactions on Signal Processing*, 2013.