İkinci Dereceden Metotlar Kullanarak Doğrusal Olmayan Bağlanım Nonlinear Regression Using Second Order Methods

Burak C. Civek¹, İbrahim Delibalta², Süleyman S. Kozat¹

¹Elektrik ve Elektronik Mühendisliği Bölümü, Bilkent Üniversitesi, Ankara, Türkiye

{civek,kozat}@ee.bilkent.edu.tr

²Türk Telekom Labs, İstanbul, Türkiye

{ibrahim.delibalta}@turktelekom.com.tr

Özetçe —Bu bildiride doğrusal olmayan çevrimiçi bağlanım problemine yönelik oldukça etkili bir algoritma geliştirilmiştir. Gözlemlenen veri anlık olarak işlenip tekrar kullanılmamaktadır, dolayısıyla depolama ihtiyacı da ortadan kaldırılmıştır. Doğrusal olmayan modelleme için bağlanım uzayının parçalandığı ve oluşan her bir bölgeye doğrusal bir modelin atandığı parçalı doğrusal modeller kullanılmıştır. Literatürde ilk defa, bağlanım uzayının parçalanması ve oluşan her bölgeye atanan doğrusal modeller ikinci derece metotlar, Newton-Raphson Metotları, ile uyarlanır olarak tasarlanmıştır. Bu sayede sıkça karşılaşılan aşırı/yetersiz uyma sorunlarına da bir çözüm geliştirilmiştir. Literatürde bulunan diğer algoritmalar ile karşılaştırıldığında önemli derecede performans artışı sağlanmıştır.

Anahtar Kelimeler—*doğrusal olmayan bağlanım, Newton, parçalı doğrusal model.*

Abstract—We present a highly efficient algorithm for the online nonlinear regression problem. We process only the currently available data and do not reuse it, hence, there is no need for storage. For the nonlinear regression, we use piecewise linear modeling, where the regression space is partitioned into several regions and a linear model is fit to each region. As the first time in the literature, we use second order methods, e.g. Newton-Raphson Methods, and adaptively train both the region boundaries and the corresponding linear models. Therefore, we overcome the well known overfitting and underfitting problems. The proposed algorithm provides a substantial improvement in the performance compared to the state of the art.

Keywords—*nonlinear regression, Newton, piecewise linear model.*

I. GİRİŞ

Doğrusal olmayan bağlanım problemi, otomatik öğrenme ve sinyal işleme literatüründe en önemli çalışma alanlarından birisidir ve sinyal modelleme, finansal market, yönelim analizi, tavsiye gibi birçok farklı gerçek hayat uygulamasının temelini oluşturur [1], [2]. Fakat geleneksel bağlanım teknikleri, büyük veri içeren gerçek hayat uygulamalarında, elde edilen verinin konvansiyonel sinyal işleme metotları ile etkili bir şekilde işlenememesi, depolama konusundaki sorunlar ve geleneksel metotların veri istatistiğine oldukça bağlı olması dolayısıyla, yeterli performansı gösterememektedir [3]. Bu bağlamda, belirtilen bu sorunların üstesinden gelmek amacıyla, elde edilen

verinin depolama ihtiyacı duyulmadan, anlık olarak işlendiği bir çevrimiçi bağlanım algoritması sunuyoruz. Sunmuş olduğumuz algoritmamız çevrimiçi ortamlarda oldukça değişken olan veri istatistiğini kolayca takip edebilmesi sayesinde, varyasyonlar ve belirsizlikler konusunda üstün bir performans sergilemektedir.

Bu bildiride, doğrusal olmayan bağlanım için karar ağaç yapısını temel alan hiyerarşik parçalı doğrusal modeller kullanılmaktadır. Bu durumda, bağlanım vektörleri uzayı uyarlamalı olarak parçalara ayrılmakta ve performansı arttırmak için sürekli eniyilemesi gerçekleştirilmektedir. Parçalı doğrusal modeller, doğrusal olmayan modellerin kullanılmasından kaynaklanan aşırı eğitim sorunlarını azaltmak için sinyal işleme literatüründe büyük oranda kullanılmaktadır. Bununla birlikte, parçalı doğrusal modellerin başarısı büyük ölçüde veriyi modelleyen uzayın doğru bir şekilde parçalanmasına bağlı olduğu için, bu modeller de gerçek hayat uygulamalarında beklenen performansı gösterememektedir. Çevrimiçi ortamlarda veri sıralı olarak elde edildiğinden, en iyi bölüntüleme başlangıçta yapılamamaktadır. Aynı şekilde, gerçek hayat uygulamalarında veri istatistiği ve dağılımı oldukça değişkenlik göstermekte ve en iyi bölüntüleme sürekli olarak değişmektedir. Bu nedenle, literatürde ilk defa, bağlanım uzayının parçalı doğrusal bölüntülenmesi ve oluşturulan her bölgedeki doğrusal modeller yüksek derecede etkili ikinci dereceden metotlar, Newton-Raphson metotları [4], kullanılarak öğrenilmektedir. Dolayısıyla, sık karşılaşılan aşırı/yetersiz uyma sorunlarından kaçınılmış olup, bölge sınırlarının ve karşılık gelen doğrusal modellerin ikinci dereceden metotlar ile eğitilmesi sayesinde, en yeni gelişmiş algoritmalarla kıyasla azımsanmayacak ölçüde daha etkili performans elde edilmiştir.

II. PROBLEM TANIMI

Bu bildiride, bütün vektörler sütun vektörleridir ve küçük koyu harflerle gösterilmektedir. Matrisler ise büyük koyu harflerle gösterilmiştir. Vektör devriği, $\mathbf{x}$ vektörü için, $\mathbf{x}^T$ ile gösterilmektedir. $\mathbf{I}_m$ ise $m \times m$ boyutlu birim matrisi temsil etmektedir.

Bildiri çerçevesinde çevrimiçi ortamda çalışılmaktadır, başka bir ifadeyle, $t \geq 1$ için kestirimini yaptığımız veri, $y_t \in \mathbb{R}$, ve öznitelik vektörü, $\mathbf{x}_t \in \mathbb{R}^m$, sıralı olarak gözlemlenmektedir. Amacımız y_t değerinin, aşağıda belirtildiği

üzere kestirimini yapmaktır:

$$\hat{y}_t = f(\mathbf{x}_t)$$

Burada, $\hat{y}_t$ ile t anındaki kestirim ve $f(\cdot)$ ile de geçmiş zaman gözlemlerinin bir fonksiyonu belirtilmiştir. Bu çalışmada, birçok doğrusal olmayan modeli yaklaşıklama yeteneği nedeniyle, parçalı doğrusal modeller kullanılmaktadır. Parçalı doğrusal modelleri oluşturmak için, öznelik vektörlerinin uzayı, $\bigcup_{k=1}^K S_k^m = \mathbb{R}^m$ ve $S_i^m \cap S_j^m = \emptyset$, $i \neq j$ koşullarını sağlamak şartıyla, örneğin K , farklı bölgeye ayrılır. Burada, S_k^m her bir bölgeyi simgelemektedir. Elde edilen her bölge için, $\hat{y}_{t,i} = \mathbf{w}_{t,i}^T \mathbf{x}_t + c_{t,i}$, şeklinde belirtilen doğrusal bir model tanımlanmaktadır. Burada, $\mathbf{w}_{t,i}$, $\hat{y}_{t,i}$ ve $c_{t,i}$ sırasıyla i ile gösterilen bölgedeki bağlanım vektörünü, kestirimi ve göreceli konumu temsil etmektedir. Bu model $\mathbf{x}_t = [\mathbf{x}_t^T : 1]^T$ ve $\mathbf{w}_{t,i} = [\mathbf{w}_{t,i}^T : c_{t,i}]^T$ değişimleri yapıldığında daha sade olarak, $\hat{y}_{t,i} = \mathbf{w}_{t,i}^T \mathbf{x}_t$ şeklinde ifade edilebilmektedir.

Parçalı doğrusal modellerin kullanımında iki farklı temel sorun bulunmaktadır: *i)* bölüntülenme sonrasında aşırı/yetersiz uyma sorunlarından kaçınılması [5] ve *ii)* oluşturulan her bir bölge için çevrimiçi olarak en iyi doğrusal modelin belirlenmesi [6]. Bu bağlamda sunduğumuz tamamı ile uyarlanabilir yapıya sahip bağlanım modeli ile her iki problem de çözülmektedir. Sunulan algoritmanın maliyet fonksiyonu olarak anlık karesel hata, $e_t^2 = (y_t - \hat{y}_t)^2$, kullanılmaktadır. Algoritma geliştirilirken ise toplam bağlanım hatasını olabildiğince küçültmek amaçlanmıştır.

III. KARAR AĞAÇ YAPISINI TEMEL ALAN ARDIŞIK PARÇALI DOĞRUSAL KESTİRİCİ

Bu bölümde parçalı doğrusal modeller temel alınarak geliştirilen oldukça etkili çevrimiçi algoritmamız sunulmaktadır. Bağlanım uzayının bölüntülenmesi ağaç yapısı üzerinden sistematik bir şekilde gerçekleştirilmektedir. İki boyutlu bağlanım düzleminin bu metot kullanılarak parçalanması Şekil 1 üzerinde gösterilmiştir. Genel durum için ise, m boyutlu uzayın parçalanması için, $p_{t,k}$ ile temsil edilen, $(m-1)$ boyutlu ayırma fonksiyonları tanımlanmaktadır:

$$p_{t,k} = \frac{1}{1 + e^{-\mathbf{x}_t^T \mathbf{n}_{t,k}}} \quad (1)$$

Burada, $\mathbf{n}_{t,k}$ normal vektörünü, k ise bölge tanımlamasını ifade etmektedir. Göreceli konum $c_{t,k}$, normal vektörünün son elemanı olarak tutulmaktadır. Şekil 1'de belirtilen durum özelinde, gözlemlenen öznelik vektörlerinin, $\mathbf{x}_t = [x_{t,1} : x_{t,2}]^T$, herhangi bir $A > 0$ değeri için $-A \leq x_{t,1}, x_{t,2} \leq A$ koşulunu sağladığını, başka bir ifadeyle $\{\Omega\}$ ile belirtilen sınırlı bir setten geldiğini kabul edelim. Tek bir ayırma fonksiyonu kullandığımızda bütün düzlemi $\{0\}$ ve $\{1\}$ olarak simgelenen ve $\{0\} \cup \{1\} = \{\Omega\}$ ifadesini sağlayan iki farklı bölgeye ayırırız. Daha sonra uygulanan her bir ayırma fonksiyonu yalnızca tek bir bölgeye etki etmektedir. Şekil 1'de gösterilen $p_{t,0}$ ve $p_{t,1}$ sırasıyla $\{0\}$ ve $\{1\}$ numaralı bölgeleri ayırmaktadır. Son durumda $\{00\}$, $\{01\}$, $\{10\}$ ve $\{11\}$ ile simgelenen, birleşimleri başlangıç düzlemini veren, dört farklı ve bağımsız bölge elde edilir. Burada, derinlik adı verilen ve uzayın bölüntülenmesinin ne kadar derin olduğunu belirten bir tanımlayıcı kullanılmaktadır. Örnek olarak, Şekil 1'de gösterilen modelin derinliği 2'dir. Genel durumda, d derinliğe sahip olan bir model, toplamda 2^d farklı bölge

Şekil 1: İki Boyutlu Bağlanım Uzayının 2 Derinlikli Hiyerarşik Model ile Parçalanması

oluşturur. Dolayısıyla, derinlik arttıkça hesaplama karmaşıklığı da $O(2^d)$ ile artar. Geliştirdiğimiz algoritmada, "en iyi" olarak tanımlanan ve en derin seviyede oluşturulan bölgeleri ifade eden modeli kullanmaktayız. Şekil 1 özelinde "en iyi" model, düzlemin $\{00\}$, $\{01\}$, $\{10\}$ ve $\{11\}$ olarak 4 farklı bölgeye ayrıldığı yapıdır.

Şekil 1'de gösterilen model göz önüne alındığında, sırasıyla $\mathbf{n}_{t,\Omega}, \mathbf{n}_{t,0}, \mathbf{n}_{t,1} \in \mathbb{R}^2$ ile tanımlanan, toplamda üç farklı ayırma fonksiyonu $p_{t,\Omega}, p_{t,0}, p_{t,1} \in \mathbb{R}$ bulunmaktadır. $\{00\}$ bölgesine karşılık gelen kestirim $\hat{y}_{t,00} = \mathbf{w}_{t,00}^T \mathbf{x}_t$ şeklinde hesaplanmaktadır. Bütün bölgelere ait kestirimlere sahip olduğumuzda, nihai kestirim

$$\hat{y}_t = p_{t,\Omega} p_{t,0} \hat{y}_{t,00} + p_{t,\Omega} (1 - p_{t,0}) \hat{y}_{t,01} + (1 - p_{t,\Omega}) p_{t,1} \hat{y}_{t,10} + (1 - p_{t,\Omega}) (1 - p_{t,1}) \hat{y}_{t,11} \quad (2)$$

şeklinde verilir. Bu sonuç kolaylıkla derinliğin 2'den fazla olduğu modellere uyarlanabilir.

Her bölgeye ait bağlanım vektörünü ve bölge sınırlarını hatayı en aza indirmek amacıyla ikinci dereceden bir metot olan Çevrimiçi Newton Adımı (ONS) [7] kullanarak uyarlanabilir şekilde güncellemekteyiz. Dolayısıyla, $\{00\}$ bölgesine karşılık gelen bağlanım vektörü şu şekilde güncellenmektedir:

$$\begin{aligned} \mathbf{w}_{t+1,00} &= \mathbf{w}_{t,00} - \frac{1}{\beta} \mathbf{A}_t^{-1} \nabla e_t^2 \\ &= \mathbf{w}_{t,00} + \frac{2}{\beta} e_t p_{t,\Omega} p_{t,0} \mathbf{A}_t^{-1} \mathbf{x}_t \end{aligned} \quad (3)$$

Burada, β adım büyüklüğünü, ∇ ise bu durum özelinde $\mathbf{w}_{t,00}$ 'a göre bayır işlecini ifade etmektedir. $m \times m$ boyutlu $\mathbf{A}_t$ matrisi şu şekilde tanımlanmıştır:

$$\mathbf{A}_t = \sum_{i=1}^t \nabla_i \nabla_i^T + \epsilon \mathbf{I}_m \quad (4)$$

Burada, $\epsilon > 0$, $\mathbf{A}_t$ matrisinin kesin artı, $\mathbf{A}_t > 0$, ve tersinir olmasını sağlamaktadır. Verilen tanımdan anlaşılacağı üzere, $\mathbf{A}_t$ matrisi hata fonksiyonunun Hessian'ı ile ilişkilidir ve ikinci dereceden bilgi içerir [7].

Bölge sınırları da aynı yol izlenerek güncellenir. Aşağıda Sekil 1'de görülen $p_{t,\Omega}$ için güncelleme kuralı verilmiştir:

$$\begin{aligned} \mathbf{n}_{t+1,\Omega} &= \mathbf{n}_{t,\Omega} - \frac{1}{\eta} \mathbf{A}_t^{-1} \nabla e_t^2 \\ &= \mathbf{n}_{t,\Omega} + \frac{2}{\eta} e_t [p_{t,0} \hat{y}_{t,00} + (1 - p_{t,0}) \hat{y}_{t,01} \\ &\quad - p_{t,1} \hat{y}_{t,10} - (1 - p_{t,1}) \hat{y}_{t,11}] \mathbf{A}_t^{-1} \frac{\partial p_{t,\Omega}}{\partial \mathbf{n}_{t,\Omega}} \end{aligned} \quad (5)$$

Burada, η adım büyüklüğünü, ∇ ise yine bu durum özelinde $\mathbf{n}_{t,\Omega}$ 'a göre bayır işlecini ifade etmektedir. $\mathbf{A}_t$ ise yine (4)'te verildiği şekilde tanımlanmıştır. Ayırma fonksiyonu $p_{t,\Omega}$ 'nın $\mathbf{n}_{t,\Omega}$ 'ya göre kısmi türevi:

$$\frac{\partial p_{t,\Omega}}{\partial \mathbf{n}_{t,\Omega}} = \frac{\mathbf{x}_t e^{-\mathbf{x}_t^T \mathbf{n}_{t,\Omega}}}{(1 + e^{-\mathbf{x}_t^T \mathbf{n}_{t,\Omega}})^2}. \quad (6)$$

şeklinde verilir. Her bir ayırma fonksiyonu aynı yol takip edilerek güncellenir. Algoritmamızın nihai kestirimi genelleyici bir formda şu şekilde hesaplanır:

$$\hat{y}_t = \sum_{j=1}^{2^d} \hat{\psi}_{t,R_d(j)} \quad (7)$$

Burada R_d , d uzunluğundaki bütün bölge etiketlerini içeren kümeyi, $R_d(j)$ ise bu kümenin j . elemanını ifade eder. Her bir bölgeye karşılık gelen ağırlıklı kestirimler ise:

$$\hat{\psi}_{t,r} = \hat{y}_{t,r} \prod_{i=1}^d \hat{p}_{t,r_i} \quad (8)$$

şeklinde hesaplanır. Burada r_i , r etiketinin ilk $(i-1)$ karakterini dizgi olarak tutmaktadır. $\hat{p}_{t,r_i}$ ise şu şekilde verilmiştir:

$$\hat{p}_{t,r_i} = \begin{cases} p_{t,r_i} & , r(i) = 0 \\ 1 - p_{t,r_i} & , r(i) = 1 \end{cases}. \quad (9)$$

Burada $r(i)$, r etiketinin i . karakterini belirtmektedir. Verilen formülasyonlar kullanılarak, bağlanım vektörü güncellemesi genel formda şu şekilde hesaplanır:

$$\mathbf{w}_{t+1,r} = \mathbf{w}_{t,r} + \frac{2}{\beta} e_t \mathbf{A}_t^{-1} \mathbf{x}_t \prod_{i=1}^d \hat{p}_{t,r_i} \quad (10)$$

Bölge sınırları ise aynı şekilde:

$$\begin{aligned} \mathbf{n}_{t+1,k} &= \mathbf{n}_{t,k} \\ &+ \frac{2}{\eta} e_t \mathbf{A}_t^{-1} \left[\sum_{j=1}^{2^{d-\ell(k)}} \hat{y}_{t,\hat{r}} (-1)^{\hat{r}(\ell(k)+1)} \prod_{\substack{i=1 \\ \hat{r}_i \neq k}}^d \hat{p}_{t,\hat{r}_i} \right] \frac{\partial p_{t,k}}{\partial \mathbf{n}_{t,k}} \end{aligned} \quad (11)$$

olarak verilir. Burada $\hat{r}$, ayırma fonksiyonu tanımlayıcısı, k , ve $R_{(d-\ell(k))}$ kümesinin j . elemanı art arda eklenerek oluşturulan karakter dizgisidir, $\hat{r} = [k; R_{(d-\ell(k))}(j)]$. İlk defa burada tanımlanan $\ell(k)$ ise, k dizgisinin uzunluğunu belirtmektedir. Bölge sınırlarını lojistik bağlanım fonksiyonu kullanarak ifade ettiğimizden dolayı, $p_{t,k}$ 'nın $\mathbf{n}_{t,k}$ 'ya göre kısmi türevi şu şekilde hesaplanabilir:

$$\frac{\partial p_{t,k}}{\partial \mathbf{n}_{t,k}} = p_{t,k}(1 - p_{t,k}) \mathbf{x}_t. \quad (12)$$

Algoritma 1: Hiyerarşik Uyarların Bölüntüleme

```

1  $\mathbf{A}_0^{-1} = \frac{1}{\epsilon} \mathbf{I}_m$ ;
2 for  $t = 1$  to  $n$  do
3    $\hat{y}_t = 0$ ;
4   for  $j = 1$  to  $2^d$  do
5      $r = R_d(j)$ ;
6      $\hat{y}_{t,r} = \mathbf{w}_{t,r}^T \mathbf{x}_t$ ;
7      $\hat{\psi}_{t,r} = \hat{y}_{t,r}$ ;
8      $\gamma_{t,r} = 1$ ;
9     for  $i = 1$  to  $d$  do
10      if  $r(i) = 0$  then
11         $\hat{p}_{t,r_i} = p_{t,r_i}$ ;
12      else
13         $\hat{p}_{t,r_i} = 1 - p_{t,r_i}$ ;
14       $\hat{\psi}_{t,r} = \hat{\psi}_{t,r} \hat{p}_{t,r_i}$ ;
15       $\gamma_{t,r} = \gamma_{t,r} \hat{p}_{t,r_i}$ ;
16    $\hat{y}_t = \hat{y}_t + \hat{\psi}_{t,r}$ ;
17   for  $i = 1$  to  $2^d - 1$  do
18      $k = P(i)$ ;
19     for  $j = 1$  to  $2^{d-\ell(k)}$  do
20        $r = \text{concat}[k : R_{d-\ell(k)}(j)]$ ;
21        $\alpha_{t,k} = (-1)^{r(\ell(k)+1)} (\hat{\psi}_{t,r} / \hat{p}_{t,k})$ ;
22    $e_t = y_t - \hat{y}_t$ ;
23   for  $j = 1$  to  $2^d$  do
24      $r = R_d(j)$ ;
25      $\nabla_{t,r} = -2e_t \gamma_{t,r} \mathbf{x}_t$ ;
26      $\mathbf{A}_{t,r}^{-1} = \mathbf{A}_{t-1,r}^{-1} - \frac{\mathbf{A}_{t-1,r}^{-1} \nabla_{t,r} \nabla_{t,r}^T \mathbf{A}_{t-1,r}^{-1}}{1 + \nabla_{t,r}^T \mathbf{A}_{t-1,r}^{-1} \nabla_{t,r}}$ ;
27      $\mathbf{w}_{t+1,r} = \mathbf{w}_{t,r} - \frac{1}{\beta} \mathbf{A}_{t,r}^{-1} \nabla_{t,r}$ ;
28   for  $i = 1$  to  $2^d - 1$  do
29      $k = P(i)$ ;
30      $\nabla_{t,k} = -2e_t \alpha_{t,k} p_{t,k} (1 - p_{t,k}) \mathbf{x}_t$ ;
31      $\mathbf{A}_{t,k}^{-1} = \mathbf{A}_{t-1,k}^{-1} - \frac{\mathbf{A}_{t-1,k}^{-1} \nabla_{t,k} \nabla_{t,k}^T \mathbf{A}_{t-1,k}^{-1}}{1 + \nabla_{t,k}^T \mathbf{A}_{t-1,k}^{-1} \nabla_{t,k}}$ ;
32      $\mathbf{n}_{t+1,k} = \mathbf{n}_{t,k} - \frac{1}{\eta} \mathbf{A}_{t,k}^{-1} \nabla_{t,k}$ ;

```

Hesaplama karmaşıklığını düşürmek adına, her adımda $m \times m$ boyutlu $\mathbf{A}_t$ matrisinin tersini hesaplamak yerine, matris evirme önsavı kullanılarak $\mathbf{A}_t^{-1}$ şu şekilde hesaplanır:

$$\mathbf{A}_t^{-1} = \mathbf{A}_{t-1}^{-1} - \frac{\mathbf{A}_{t-1}^{-1} \nabla_t \nabla_t^T \mathbf{A}_{t-1}^{-1}}{1 + \nabla_t^T \mathbf{A}_{t-1}^{-1} \nabla_t} \quad (13)$$

Burada, $\nabla_t \triangleq \nabla e_t^2$ tanımlaması yapılmıştır. Geliştirilen algoritma başlangıç değerleri ve güncelleme kuralları ile birlikte Algoritma 1 üzerinden takip edilebilir.

Bağlanım uzayı, d derinlikli bir model kullanıldığında 2^d bölgeye ayrılır. Dolayısıyla her adımda $O(2^d)$ güncelleme gerektirmektedir. Bağlanım uzayının m boyutlu olduğu kabul edilirse, her güncelleme için matris vektör çarpımından kaynaklanan $O(m^2)$ hesaplama gerekmektedir. Sonuç olarak algo-

ritmimizin hesaplama karmaşıklığı $O(m^2 2^d)$ olarak verilmektedir.

Teorem 1. $\{y_t\}_{t \geq 1}$ ve $\{x_t\}_{t \geq 1}$ rasgele seçilmiş gerçel veri dizilerini temsil etsin. $G, A, \alpha > 0$ olmak üzere $\|\nabla(y_t - \hat{y}_{t,r})^2\| \leq G$, $\|w_{t,r} - w_r\|^2 \leq A^2$ ve $\exp(-\alpha(y_t - \hat{y}_{t,r})^2)$ içbükey ise, kestirim $\hat{y}_t$ için Algoritma 1 takip edildiğinde, verilen logaritmik sınır sağlanmaktadır:

$$\sum_{t=1}^n (y_t - \hat{y}_{t,r})^2 - \min_{w_r \in \mathbb{R}^m} \sum_{t=1}^n (y_t - w_r^T x_t)^2 \leq 5 \left(GA + \frac{1}{\alpha} \right) m \log(n)$$

Teorem 1'de, her bir bölgede yapılan kestirim için n . adımdaki "pişmanlık" değerinin logaritmik bir üst sınırı olduğu gösterilmiştir ve ispatı [7]'de verilen yaklaşım üzerinden yapılır.

IV. SAYISAL SONUÇLAR

Bu bölümde, sunulan algoritmanın performans değerlendirilmesi literatürde halihazırda bulunmakta olan diğer algoritmalar ile karşılaştırmalı olarak gerçekleştirilmiştir. Değerlendirme için aşağıda verilen parçalı doğrusal model kullanılarak üretilen iki boyutlu veri dizisi kullanılmıştır:

$$\hat{y}_t = \begin{cases} w_1^T x_t + v_t, & x_t^T n_0 \geq 0.5 \text{ ve } x_t^T n_1 \geq -0.5 \\ w_2^T x_t + v_t, & x_t^T n_0 \geq 0.5 \text{ ve } x_t^T n_1 < -0.5 \\ w_2^T x_t + v_t, & x_t^T n_0 < 0.5 \text{ ve } x_t^T n_2 \geq -0.5 \\ w_1^T x_t + v_t, & x_t^T n_0 < 0.5 \text{ ve } x_t^T n_2 < -0.5 \end{cases} \quad (14)$$

Burada, bağlanım vektörleri; $w_1 = [1, 1]^T$, $w_2 = [1, -1]^T$, ayırma fonksiyonlarının normal vektörleri ise; $n_0 = [2, -1]^T$, $n_1 = [-1, 1]^T$ ve $n_2 = [2, 1]^T$ olarak tanımlanmıştır. Öznitelik vektörü, $x_t = [x_{t,1}, x_{t,2}]^T$, $[0, 0]^T$ ortalama ve I_2 varyans değerleri ile Gaussian dağılıma sahip iki rastgele değişkenden oluşmuştur. v_t ise sıfır ortalama ve 0.1 varyansa sahip Gaussian dağılımının bir örneklemevidir. Üretilen verinin iki boyutlu düzlem üzerindeki verilen bu dağılımı ile değerlendirmede yer alan algoritmaların başlangıç bölgeleri eşleşmemektedir, dolayısıyla bölge sınırlarının veriye göre nasıl uyarlandığı gözlemlenebilecektir.

Bu bölüm boyunca, "HUB" sunmuş olduğumuz Hiyerarşik Uyarlanırlı Bağlanım algoritmasını, "KUA" Karar Uyarlanırlı Ağaç (DAT) [8] algoritmasını, "BAA" Bağlam Ağaç Ağırlıklandırma (CTW) [6] algoritmasını, "GKB" Gaussian-Kernel Bağlanımını (GKR) [9], "VS" Volterra Süzgecini (VF) [10], "FDOS" ve "ÇYFDOS" ise Fourier ve Çift Yansı Fourier Doğrusal Olmayan Süzgeçlerini (FNF), (EMFNF) [11] temsil etmektedir. Algoritmaların öğrenme oranları "HUB" için 0.04, "KUA" için 0.01, "BAA" ve "FDOS" için 0.005, "GKB" için 0.5, "VS" ve "ÇYFDOS" için ise 0.025 olarak belirlenmiştir.

Şekil 2 üzerinde algoritmaların birikimli düzelenmiş hata oranları gösterilmektedir. Burada görüldüğü üzere, bölge sınırlarını uyarlayabilme kabiliyeti sayesinde, sunmuş olduğumuz "HUB" algoritması, oluşturulan senaryo çerçevesinde rakip algoritmalarla göre üstün bir performans sergilemektedir.

V. SONUÇLAR

Bu bildiride, çevrimiçi öğrenme problemleri için oldukça etkili ve verimli doğrusal olmayan bir bağlanım algoritması sunulmuştur. Bu algoritma ile elde edilen veri anlık

Şekil 2: Bağlanım Uzayı ve Algoritmaların Başlangıç Bölgelerinin Eşleşmemiş Olduğu Durumda Birikimli Hata Oranları

olarak işlenip öğrenme gerçekleştirildikten sonra tekrar kullanılmamaktadır, dolayısıyla depolama ihtiyacı ortadan kaldırılmıştır. Doğrusal olmayan modelleme için bağlanım uzayının bölgelere ayıran ve her bölgeye doğrusal bir model uyarlayan parçalı doğrusal modeller kullanılmıştır. Literatürde ilk defa, hem bölge sınırları hem de bölgelere atanan doğrusal modeller ikinci derece güncelleme metotları, Newton-Raphson metotları ile eğitilmiştir. Geliştirdiğimiz algoritmanın, halihazırda literatürde yer alan algoritmalarla göre üstün bir performans gösterdiği sergilenmiştir.

KAYNAKÇA

- [1] A. C. Singer, G. W. Wornell, and A. V. Oppenheim, "Nonlinear autoregressive modeling and estimation in the presence of noise," *Digital Signal Processing*, vol. 4, no. 4, pp. 207–221, 1994.
- [2] O. J. J. Michel, A. O. Hero, and A.-E. Badel, "Tree-structured nonlinear signal modeling and prediction," *IEEE Transactions on Signal Processing*, vol. 47, no. 11, pp. 3027–3041, 1999.
- [3] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing (NIPS)*, 2007, pp. 1–8.
- [4] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*, ser. Athena scientific series in optimization and neural computation. Belmont (Mass.): Athena Scientific, 1997. [Online]. Available: <http://opac.inria.fr/record=b1094316>
- [5] S. Dasgupta and Y. Freund, "Random projection trees for vector quantization," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3229–3242, 2009.
- [6] 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.
- [7] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, 2007.
- [8] N. Vanli and S. Kozat, "A comprehensive approach to universal piecewise nonlinear regression based on trees," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5471–5486, Oct 2014.
- [9] R. Rosipal and L. J. Trejo, "Kernel partial least squares regression in reproducing kernel hilbert space," *J. Mach. Learn. Res.*, vol. 2, pp. 97–123, Mar. 2002. [Online]. Available: <http://dl.acm.org/citation.cfm?id=944790.944806>
- [10] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. NJ: John Wiley & Sons, 1980.
- [11] A. Carini and G. L. Sicuranza, "Fourier nonlinear filters," *Signal Processing*, vol. 94, no. 0, pp. 183 – 194, 2014.