Hastalık Teşhisi İçin Bağlamsal Üstel Ağırlıklar İle Çevrimiçi Sınıflandırma

Online Classification with Contextual Exponential Weights for Disease Diagnostics

Kubilay Ekşioğlu, Muhammad Anjum Qureshi, Cem Tekin
Elektrik ve Elektronik Mühendisliği Bölümü, Bilkent Üniversitesi, Ankara, Türkiye
{eksioglu, qureshi, cemtekin}@ee.bilkent.edu.tr

Özetçe—Bu bildiride Ağırlıklı Ortalamalı Tahminci (Weighted Average Forecaster) algoritmasının bağlamsal varyantına dayanan yeni bir çevrimiçi sınıflandırma algoritması önerilmiştir. Veri kümesini çevrimiçi olarak bağlamlara göre bölümleyen bu algoritma, uzmanlardan gelen önerilerin ağırlıklarını bağlama göre güncelleyerek, uzman önerilerinin bağlamsal füzyonunu sağlamaktadır. Önerilen algoritma, UCI veri deposunda mevcut bulunan hastalık verileri üzerinde denenmiştir. Elde edilen sonuçlar, tıbbi teşhis alanında, önerilen sistemin performans ve düşük hesaplama maliyeti açısından sağlamlık, etkililik ve çok önlülüğünü kanıtlamaktadır.

Anahtar Kelimeler—Üstel ağırlıklar, bağlamsal haydutlar, çevrimiçi sınıflandırma, karar ağaçları, sınıf dengesiz veri kümeleri

Abstract—In this paper, a novel online scheme for classification, which is based on the contextual-variant of Weighted Average Forecaster Algorithm is proposed. The proposed method adaptively partitions the data space based on contexts, and trade-offs exploration and exploitation when fusing the predictions of the experts. The proposed algorithm is verified on disease data available in UCI Online Machine Learning Repository. These results prove the robustness, effectiveness and versatility in terms of performance and low computational cost of the proposed system in the field of medical diagnostics.

Keywords—Exponential weights, contextual bandits, online classification, decision trees, biased datasets

I. GİRİŞ

Hastalık teşhisinde, negatif (sağlıklı) örnekler veri kümesinin büyük bir kısmını oluşturmaktadır. Sınıflar arası dengesizlik olan veri kümelerinde sınıflandırma yapılırken, bilinen algoritmaları doğrudan kullanmak mümkün değildir. Genellikle bu gibi sınıflandırma problemlerinde çözüm, farklı yanlış sınıflandırma maliyetleri eklemek, lehine dengesizlik olan sınıftan az sayıda örnek almak veya aleyhinde dengesizlik olan sınıftan sentetik yeni örnekler üretmektir [3], [5].

Birden fazla bağımsız uzmanın (sınıflandırıcının) bir araya getirildiği toplu sınıflandırıcıların doğruluk performansı, kendisini oluşturan uzmanların performansından üstündür [4].

Bu çalışma TÜBİTAK tarafından 2232 Burs Programı kapsamında desteklenmektedir (Proje no: 116C043).

978-1-5090-6494-6/17/\$31.00 ©2017

Ağırlık temelli toplu sınıflandırıcılar gerçek zamanlı çalışabilir. Uzmanlardan önerileri alınır, bu öneriler uzmanların ağırlığına bağlı olarak birleştirilir ve sınıfa karar verilir. Sonrasında sistem, uzmanların ağırlıklarını önerilerinin doğru karara uzaklığına bağlı olarak günceller [6], [9].

Bu çalışmada, çok sayıda eksik bilgilendirilmiş bağımsız uzmanın önerilerinin birleştirilmesinden oluşmuş yeni bir çevrimiçi sınıflandırma yöntemi sunulmaktadır. Her bir uzman veri kümesinin küçük ve diğer uzmanlardan farklı bir bölümüyle eğitilir, bu sayede her uzmanın diğerlerinden farklı örnektüleri öğrenmesi sağlanır. Örneklerin oluşturduğu uzay, gelen örneklerin özelliğine bağlı olarak parçalara bölünür. Her parça, içerisinde kendi Ağırlıklı Ortalamalı Tahminci (AOT) parametrelerini bulundurur. Bir örnek geldiğinde önce örneğin hangi parçada olduğu hesaplanır, sonrasında ise bu parçanın parametrelerine bakılarak sınıflandırma gerçekleştirilir. Sınıflandırma sonucuna göre parçanın parametreleri güncellenir. Deneyisel sonuçlar, Bağlamsal Ağırlıklı Ortalamalı Tahminci (BAOT) ismini verdiğimiz bu algoritmanın sınıflar arası dengesizlik olan veri kümelerinde AOT yönteminden ve diğer iyi bilinen sınıflandırma yöntemlerinden daha üstün performans gösterdiğini ortaya koymaktadır.

Bu bildirinin katkıları aşağıda listelenmiştir.

- Önerilen algoritma AOT [2] algoritmasının yeni bir varyantıdır. Algoritmaya bağlamsal çerçeve eklenerek algoritmanın performansı artırılmıştır.
- Önerilen algoritma bölümlemeyi Benzerlik Bilgili Uyarlamalı Bağlamsal Haydutlar [8] için kullanılan yakınlıkla tekniğinin bir çeşitlemesine göre gerçekleştirir. Bu çeşitlemede [8]'dekinden farklı olarak eylem uzayının değil sadece bağlam uzayının uyarlamalı bölümlemesi yapılmaktadır.
- Algoritmanın sınıflandırma performansı hem kendini oluşturan uzman havuzuyla, hem de güncel sınıflandırma yöntemleriyle karşılaştırılmış ve performans üstünlüğü gösterilmiştir.

II. PROBLEM TANIMI

Veriler d boyutlu $\mathcal{X}$ kümesinden gelmektedir. Her boyut bir niteliği temsil etmektedir. Sistemdeki uzman sayısı N ile, sınıf sayısı K ile gösterilir. Sistemdeki i 'nci uzman, verilen bir $\mathbf{x} \in \mathcal{X}$ için sınıflar üzerinde $e_i(\mathbf{x})$ ile gösterilen $K \times 1$ boyutlu bir olasılıksal dağılım vektörü önerir. Tüm uzmanların önerilerinin birleşimi $\mathbf{e}(\mathbf{x}) = [e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_N(\mathbf{x})]^T$ olarak gösterilir. Yine e_i uzmanının $\mathbf{x}(t)$ örneğine verdiği öneri ise $\mathbf{f}_i(t) \triangleq [f_{i,1}(t), \dots, f_{i,K}(t)]^T = e_i(\mathbf{x}(t))$ ile gösterilir. $\mathbf{f}(t) = [\mathbf{f}_1(t), \dots, \mathbf{f}_N(t)]^T$ bütün uzmanların $\mathbf{x}(t)$ 'ye verdikleri önerileri birleştiren $N \times K$ matrisi ifade eder. Bu öneriler algoritma tarafından sınıflandırmada kullanılır. $\mathbb{1}$ indikatör fonksiyonu

$$\mathbb{1}(\alpha) = \begin{cases} 0 & \alpha \text{ yanlış ise} \\ 1 & \alpha \text{ doğru ise} \end{cases}$$

olarak tanımlanmıştır. Algoritmanın performansı pişmanlık kriteri ile ölçülür. Pişmanlık, algoritmanın T anına kadar en iyi uzmanının üstüne yaptığı beklenen yanlış sınıflandırma sayısını ifade eder. Amaç, T anına kadar olan pişmanlığı minimize edebilen bir algoritma geliştirmektir.

III. ÖNERİLEN ÇEVİRİMİÇİ SINIFLANDIRMA YÖNTEMİ

A. Uzmanların Eğitimi

Uzmanların birleşiminden oluşan sistemlerin performansı, sistemi oluşturan uzmanların performansına bağlıdır [6]. Önerilen algoritma bağlamsal bir öğrenme gerçekleştireceği için farklı bağlamlarda uzmanlara farklı ağırlıklar ataması performansı artıracaktır. Bu nedenle, eğitim veri kümesinden rastgele N küme oluşturulmuş, her bir küme ile bir CART Karar Ağacı [1] eğitilmiş ve her uzmanın eğitim veri kümesinden farklı bilgiler edinmesi sağlanmıştır. Kümelerin nasıl oluşturulduğunun detayı Deney Sonuçları bölümünde verilmiştir.

B. Ağırlıklı Ortalamalı Tahminci (AOT)

AOT [2] yöntemi, uzmanlara daha önceki önerilerinin isabetlerine bakarak birer ağırlık atar. Sonrasında ise uzmanların önerilerinin ağırlıklı toplamını hesaplar. i 'nci uzman için ağırlık $w_i(t)$, uzmanların ağırlık vektörü $\mathbf{w}(t) = [w_1(t), \dots, w_N(t)]^T$, t zamanındaki örneğin j sınıfı için birleştirilmiş olasılığı $p_j(t)$ aşağıdaki gibi hesaplanır:

$$p_j(t) = \sum_{i=1}^N w_i(t) f_{i,j}(t). \quad (1)$$

t anındaki birleştirilmiş olasılık vektörü $\mathbf{p}(t) = [p_1(t), p_2(t), \dots, p_K(t)]$ ile gösterilir. Algoritma tarafından tahmin edilen sınıf, birleştirilmiş olasılığı en yüksek olan sınıftır:

$$\hat{y}(t) = \arg \max_j (p_j(t)). \quad (2)$$

Gerçek sınıf $y(t)$ ortaya çıktığında, tüm uzmanlar önerilerinin gerçek sınıftan ne kadar uzak olduğuna bağlı olarak $\ell_i(t)$ değerinde kayba uğrarlar. $K \times 1$ boyutlu $\mathbf{z}(t)$ vektörü $\mathbf{x}(t)$ örneğinin sınıf üyeliğini temsil etmektedir:

$$\mathbf{z}(t) = [\mathbb{1}(y(t) = 1), \dots, \mathbb{1}(y(t) = K)]^T. \quad (3)$$

Algoritma 1 Ağırlıklı Ortalamalı Tahminci (AOT)

```

1: function FORECAST( $\mathbf{w}(t)$ ,  $\mathbf{f}(t)$ )
2:   for  $j = 1, 2, \dots, K$  do
3:      $p_j(t) \leftarrow \sum_{i=1}^N w_i(t) f_{i,j}(t)$ 
4:    $\hat{y}(t) \leftarrow \arg \max_j p_j(t)$ 
5:   return  $\hat{y}(t)$ 
6: function UPDATEWEIGHTS( $\eta_t$ ,  $\mathbf{f}(t)$ ,  $\hat{y}(t)$ ,  $y(t)$ ,  $\mathbf{L}(t-1)$ )
7:   for  $i=1, \dots, N$  do
8:      $\ell_i(t) \leftarrow \sum_{j \neq y(t)} f_{i,j}(t)$ 
9:      $L_i(t) = L_i(t-1) + \ell_i(t)$ 
10:  for  $i = 1, \dots, N$  do
11:     $w_i(t+1) = \frac{\exp(-\eta_t L_i(t))}{\sum_{r=1}^N \exp(-\eta_t L_r(t))}$ 
12:  return  $\mathbf{w}(t+1)$ ,  $\mathbf{L}(t)$ 
13: function WEIGHTEDAVERAGEFORECASTER( $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{e}$ )
14:  Init:
15:    for  $i = 1, \dots, N$  do
16:       $w_i(1) = 1/N$ 
17:       $L_i(0) = 0$ 
18:  for  $t=1, \dots, T$  do
19:     $\mathbf{f}(t) \leftarrow \mathbf{e}(\mathbf{x}(t))$ 
20:     $\hat{y}(t) \leftarrow \text{FORECAST}(\mathbf{w}(t), \mathbf{f}(t))$ 
21:     $\eta_t \leftarrow \sqrt{\frac{\alpha \ln(N)}{t}}$ 
22:     $\mathbf{w}(t+1)$ ,  $\mathbf{L}(t) \leftarrow$ 
23:      UPDATEWEIGHTS( $\eta_t$ ,  $\mathbf{f}(t)$ ,  $\hat{y}(t)$ ,  $y(t)$ ,  $\mathbf{L}(t-1)$ )

```

Uzmanların kaybı, önerdikleri olasılık dağılımının gerçek sınıfa olan mesafesinin bir fonksiyonu olarak tanımlanır ve

$$\ell_i(t) = 1 - \mathbf{f}_i(t)^T \cdot \mathbf{z}(t) \quad (4)$$

olarak hesaplanır.

Uzmanların ağırlıkları, bu kayıp fonksiyonu kullanılarak toplam kaybın bir üstel fonksiyonunun tüm uzmanlar üzerinde normalize edilmiş hali olarak hesaplanır:

$$L_i(t) = \sum_{s=1}^t \ell_i(s) \quad (5)$$

$$w_i(t+1) = \frac{\exp(-\eta_t L_i(t))}{\sum_{r=1}^N \exp(-\eta_t L_r(t))} \quad (6)$$

η_t öğrenme hızı parametresi ve α öğrenme katsayısı olmak üzere, $\eta_t = \sqrt{\frac{\alpha \ln(N)}{t}}$ olarak hesaplanır. $\alpha = 8$ seçildiğinde AOT algoritması için T anına kadar olan pişmanlık $\sqrt{2T \ln(N)}$ ile sınırlıdır [2]. Ağırlıklı Ortalamalı Tahminci, Algoritma 1'de verilmiştir.

C. Bağlamsal Ağırlıklı Ortalamalı Tahminci (BAOT)

Bağlam vektörü $\mathbf{x}_t$, t anındaki bağlamı ifade eder ve nitelik vektörü $\mathbf{x}(t)$ 'den farklıdır. Örneğin $\mathbf{x}_t$, $\mathbf{x}(t)$ 'nin niteliklerinin

bir alt kümesi veya $\mathbf{x}(t)$ 'nin doğrusal bir transformasyona uğramış hali olabilir. Bu makaledeki deneylerde bağlam, örneğin reel değerli niteliklerinden oluşturulmuştur. X , bağlamların alabileceği tüm değerleri içeren çok boyutlu bir uzayı; Y , eylemlerin alabileceği değerleri içeren sınırlı uzayı ifade eder. Slivkins, bağlamların ve eylemlerin birleşiminden oluşan $P \subset X \times Y$ benzerlik uzayının nasıl parçalara bölüneceğini açıklamıştır [8]. Ancak bu makalede sadece sınıflandırılma yapılacağı için, tüm eylemler arasındaki uzaklık eşit varsayılabilir. Bu nedenle P benzerlik uzayı yerine X bağlam uzayını parçalara bölmek yeterli olacaktır.

BAOT, Algoritma 2'de verilmiştir. Bağlam uzayının top şeklindeki parçalara bölümü Slivkins'in önerdiği yöntemle yapılmıştır [8]. $center(B)$, $r(B)$, B topunun merkezini ve yarıçapını, $rew(B)$, $\mu(B)$ ise B topunun toplam ve ortalama ödülleri ifade eder. D_X bağlam uzayındaki uzaklık fonksiyonu olmak üzere, B topunun kapladığı bağlamların kümesi

$$members(B) = \{\mathbf{x} : r(B) > D_X(center(B), \mathbf{x})\} \quad (7)$$

olur. $\mathbf{x}$ bağlam vektörü $c \times 1$ boyutlu ise, uzaklık fonksiyonu olarak $D_X(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\|\mathbf{x}_1 - \mathbf{x}_2\|^2 / c}$ kullanılabilir.

Bağlam $\mathbf{x}_t$ geldiğinde, bu bağlamın hangi topların etki alanında olduğu hesaplanır. $\mathcal{A}_t$, t anındaki topların kümesidir. $\mathcal{A}_t^B = \{B' \in \mathcal{A}_t : r(B') < r(B)\}$ olarak tanımlanır ve B topunun $\mathcal{A}_t$ kümesi içindeki etki alanı:

$$dom(B, \mathcal{A}_t) \triangleq members(B) - \bigcup_{B' \in \mathcal{A}_t^B} members(B') \quad (8)$$

ile gösterilir.

B topunun t anındaki güven yarıçapı $conf_t(B)$:

$$conf_t(B) \triangleq 4 \sqrt{\frac{\log T}{1 + n(B)}} \quad (9)$$

olarak tanımlanır. Burada $n(B)$, B topunun t anına kadar kaç kere seçildiğini gösterir.

$\mathbf{x}_t$ bağlamını etki alanı içerisine alan topların kümesi $\mathcal{B} = \{B \in \mathcal{A}_t : \mathbf{x}_t \in dom(B, \mathcal{A}_t)\}$ olur. Bağlam $\mathbf{x}_t$ için seçilecek top $B_t \triangleq \arg \max_{B \in \mathcal{B}} (I_t(B))$ olarak tanımlanır. $I_t(B)$, B topunun üst güven sınırını ifade eder ve aşağıdaki şekilde hesaplanır:

$$D(B, B') \triangleq D_X(center(B), center(B')) \quad (10)$$

$$I_t^{pre}(B) = \mu(B) + r(B) + conf_t(B) \quad (11)$$

$$I_t(B) = r(B) + \min_{B' \in \mathcal{A}_t} (I_t^{pre}(B') + D(B, B')) \quad (12)$$

$\mathbf{W}(B_t)$, $\mathbf{L}(B_t)$ sırasıyla uzmanların B_t topu içerisindeki ağırlıkları ve toplam kayıplarını ifade eden $N \times 1$ boyutlu vektörlerdir.

$$\mathbf{p}(t) = \mathbf{W}(B_t)^T \cdot \mathbf{f}(t) \quad (13)$$

hesaplandıktan sonra $\hat{y}(t)$ sınıf tahmini yapılır. Bu durumda ortaya çıkan ödül $\pi = \mathbb{1}(\hat{y}(t) = y(t))$ ile gösterilir. Sonrasında $\mathbf{W}(B_t)$ ve $\mathbf{L}(B_t)$ AOT algoritmasındaki gibi güncellenir.

Algoritma 2 Bağlamsal Ağırlıklı Ortalamalı Tahminci (BAOT)

```

1: function INITBALL(center, radius)
2:    $B \leftarrow B(\text{center}, \text{radius})$ 
3:    $n(B) = rew(B) = 0$ 
4:   for  $i=1, \dots, N$  do
5:      $W_i(B) = 1/N$ 
6:      $L_i(B) = 0$ 
7:   return  $B$ 
8: function ADAPTIVECONTEXTUALWAF( $\mathbf{x}, \mathbf{y}, \mathbf{e}$ )
9:   Init:
10:     $B \leftarrow \text{INITBALL}(\mathbf{x}_1, 1)$ 
11:     $\mathcal{A} \leftarrow \{B\}$ 
12:   for  $t=1 \dots T$  do
13:     Get  $\mathbf{x}_t$ 
14:      $B_t \leftarrow \arg \max_{B \in \mathcal{B}} I_t(B)$ 
15:      $\mathbf{f}(t) \leftarrow \mathbf{e}(\mathbf{x}(t))$ 
16:      $\hat{y}(t) \leftarrow \text{FORECAST}(\mathbf{W}(B_t), \mathbf{f}(t))$ 
17:      $\pi \leftarrow \mathbb{1}(\hat{y}(t) = y(t))$ 
18:      $n(B_t) \leftarrow n(B_t) + 1; rew(B_t) \leftarrow rew(B_t) + \pi$ 
19:      $\eta_t \leftarrow \sqrt{\frac{\alpha \ln(N)}{n(B_t)}}$ 
20:      $\mathbf{W}(B_t), \mathbf{L}(B_t) \leftarrow$ 
21:       UPDATEWEIGHTS( $\eta_t, \mathbf{f}(t), \hat{y}(t), y(t), \mathbf{L}(B_t)$ )
22:     if  $conf_t(B_t) \leq r(B_t)$  then
23:        $\hat{B} \leftarrow \text{INITBALL}(\mathbf{x}_t, \frac{r(B_t)}{2})$ 
24:        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\hat{B}\}$ 

```

IV. PERFORMANS DEĞERLENDİRMESİ

Bu bölümde, veri kümesinin detayları verilmiş ve önerilen yöntemin performansı diğer bilinen yöntemlerle karşılaştırılmıştır.

A. Simülasyon

Veri Kümesi: Bu sınıflandırma görevi için kullanılan veri kümesi, UCI (Center of Machine Learning and Intelligent Systems, University of California) Makine Öğrenimi Veri Deposu'ndan alınmıştır [7]. Veri kümesi sağlıklı sınıf lehine dengesizlik göstermektedir. Örneklerin %92.5'i sağlıklı gruba aitken %7.5'i hasta gruba aittir. Hasta grupta iki farklı hastalık vardır, veri kümesi toplam 3 sınıf içermektedir. Hastalara uygulanan testlerden 21 nitelik toplanmıştır; bunların 6'sı reel, 15'i ikilidir. Bu sınıflandırma görevinde eğitim için 3772, test için 3428 örnek bulunmaktadır.

Bilinen algoritmalar: Önerilen Bağlamsal Üstel Ağırlıklı sistemin performansı, farklı makine öğrenim algoritmalarıyla karşılaştırılmıştır. Bu algoritmalar Yardımcı Vektör Makineleri (SVM), k En Yakın Komşu (k-NN), CART Karar Ağacı, AOT ve eğitilen uzman havuzundan seçilen 3 uzmandır (En İyi, Ortalama ve En Kötü uzmanlar). Uzmanların iyilik sıralaması, test setindeki ortalama doğru tahmin oranlarına bakılarak yapılmıştır.

Deney Parametreleri: Bu deneyde, 3 parametre değerlendirilmiştir: tahmin hata oranı (PER), yanlış pozitif oranı (FPR) ve negatif hata oranı (FNR). PER, gerçekte hasta olup sağlıklı tahmin edilen veya gerçekte sağlıklı olup hasta tahmin

edilen örneklerin toplam örnek sayısına oranıdır. FPR negatif (sağlıklı) örneklerden pozitif (hasta) olarak sınıflandırılanların oranıdır. FNR ise pozitif örneklerden negatif olarak sınıflandırılanların oranıdır.

B. Deney Sonuçları

Karşılaştırılan tüm bilinen algoritmalar eğitim veri kümesinin tamamıyla eğitilmiştir. Uzman havuzunu oluşturan CART karar ağaçlarını üretmek için eğitim seti içerisinde birbirinden uzakta 10 veri kümesi seçilmiştir, bu kümelerdeki örnek sayıları büyüklükleri 10 ile 100 arasında rastgele atanmıştır. Sonra seçilen kümelerle CART karar ağaçları eğitilmiş ve uzmanların farklı alanlarda eğitilmesi sağlanmıştır. Sonrasında, sınıf dengesizliğini azaltacak şekilde rastgele seçimler yapılarak test setinden 3,000 örnek oluşturulmuştur ve algoritmalar bu örneklerle değerlendirilmiştir. Bu sayede problem zorlaştırılarak, eğitim ve test kümelerinin farklı dağılımlardan gelmesi sağlanmıştır. Deney 25 defa tekrarlanmış ve her deneyde seçilen örnekler ve sıraları değiştirilmiştir. Deneyler sırasında $\alpha = 8$ seçilmiştir.

Tablo I algoritmaların performans parametrelerini içermektedir. PER, FPR, FNR parametreleri için ortalama ve standart sapma değerleri verilmiştir. Önerilen yöntemin ortalama PER parametresi, kendini oluşturan uzmanların en iyisinden %0.63, AOT algoritmasından %0.52 daha iyidir. Deneylerde k-NN ve SVM'nin performansının düşük olduğu gözlemlenmiştir. Bunun nedeni eğitim kümesinde sınıf dağılımının dengesiz olmasıdır. Bağlamsal AOT performans açısından genellikle en iyi uzmandan daha üstündür.

Ağırlıklı Ortalamalı Tahminci algoritmasının bağlamsal varyantının, öğrenme eğrisi de orijinal versiyonuna göre daha üstündür. Bu eğriler Şekil 1'de gösterilmiştir.

V. SONUÇ

Bu bildiride sınıflar arası dengesiz dağılıma sahip eğitim veri kümelerinden öğrenme yapabilen bir algoritma sunulmuştur. Bu algoritma Ağırlıklı Ortalamalı Tahminci algoritmasının yeni bir varyantıdır. Bağlam uzayını uzaklıklara göre parçalara bölüp her bir parçada ayrı ağırlık tanımlayan bu varyantın kendini oluşturan uzmanlardan ve bilinen güncel sınıflandırma yöntemlerinden daha yüksek performansa sahip olduğu gösterilmiştir. Algoritma aynı zamanda bağlam kullanmayan orijinal versiyonuna göre de daha yüksek performans göstermektedir.

TABLO I: Algoritmaların Sınıflandırma Performanslarının Ortalama ve Standart Sapma Karşılaştırması

Ölçü Parameters	Ortalama			Standart Sapma		
	PER%	FPR%	FNR%	PER	FPR	FNR
Bağlamsal AOT	1.63	2.63	0.54	0.41	1.15	0.52
AOT	2.15	3.63	0.005	0.50	0.54	0.025
SVM	38.53	0.00	94.69	10.62	0.00	2.49
k-NN	24.95	2.99	56.89	7.19	0.37	9.66
Karar ağacı	2.78	0.51	6.15	1.08	0.21	2.85
En İyi Uzman	2.26	3.66	0.31	0.61	0.62	1.54
Ortalama Uzman	11.12	5.31	20.65	9.58	8.19	21.07
En Kötü Uzman	24.62	16.42	33.16	7.84	10.75	11.64

Şekil 1: BAOT ve AOT algoritmalarının öğrenim eğrileri

Medikal veri kümeleri genellikle dengesiz dağılıma sahip olduğu için, bu algoritmanın medikal veri kümelerinde kullanımı faydalı olacaktır.

KAYNAKLAR

- [1] Olshen Stone Breiman, Friedman. *Classification and regression trees*. Wadsworth Brooks, 1984.
- [2] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [3] Xiao Lin Chen, Yan Jiang, Min Jie Chen, Yong Yu, Hong Ping Nie, and Min Li. A dynamic cost sensitive support vector machine. In *Advanced Materials Research*, volume 424, pages 1342–1346. Trans Tech Publ, 2012.
- [4] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37. Springer, 1995.
- [5] Bartosz Krawczyk, Michał Woźniak, and Gerald Schaefer. Cost-sensitive decision tree ensembles for effective imbalanced classification. *Applied Soft Computing*, 14:554–562, 2014.
- [6] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. In *30th Annual Symposium on Foundations of Computer Science*, pages 256–261. IEEE, 1989.
- [7] Ross Quinlan. Thyroid Disease Data Set. <https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease>. [Çevrimiçi; en son erişim: 02 Şubat 2017].
- [8] Aleksandrs Slivkins. Contextual bandits with similarity information. *Journal of Machine Learning Research*, 15(1):2533–2568, 2014.
- [9] Cem Tekin, Jinsung Yoon, and Mihaela van der Schaar. Adaptive ensemble learning with confidence bounds. *IEEE Transactions on Signal Processing*, 65(4):888–903, 2016.