

Çok Yüzlü Konik Sınıflandırıcılarda Koni Tepe Noktası Tahmini

Cone Vertex Estimation in Polyhedral Conic Classifiers

Golara Ghorban Dordinejad, Hakan Çevikalp
Elektrik ve Elektronik Mühendisliği Bölümü
Eskişehir Osmangazi Üniversitesi
Eskişehir, Türkiye
{golaradordinejad, hakan.cevikalp}@gmail.com

Özetçe—Çok yüzlü konik sınıflandırıcılar son zamanlarda Destek Vektör Makinelerine oranla çok daha iyi sonuçlar verdiklerinden oldukça popüler hale gelmişlerdir. Bu tür sınıflandırıcıların önemli parametrelerinden biri de koni tepe noktasıdır. Literatürde genel olarak çok yüzlü konik sınıflandırıcılar için koni tepe noktası pozitif verilerin ortalaması olarak alınmaktadır. Bu bildiride, daha başarılı çok yüzlü konik sınıflandırıcı elde etmek için koni tepe noktasının optimal olarak tahmini üzerinde çalışmalar yapılmıştır. Geliştirilen yöntem, pozitif verilerin ortalamasını kullanan çok yüzlü konik sınıflandırıcılar, genişletilmiş çok yüzlü konik sınıflandırıcılar ve Destek Vektör Makineleri gibi sınıflandırıcılar ile karşılaştırılmıştır. Bu karşılaştırmalar için *UCI Machine Learning Repository*'deki çeşitli gerçek veri tabanları ve ayrıca oluşturulan sentetik bir veri kümesi kullanılmıştır. Deney sonuçları sunulan yaklaşımın veri sınıflandırma problemlerinin çözümünde etkili sonuçlar verdiğini göstermektedir.

Anahtar Kelimeler—*koni tepe noktası tahmini, çok yüzlü konik sınıflandırıcılar, genişletilmiş çok yüzlü konik sınıflandırıcılar, sınıflandırma.*

Abstract—Recently, polyhedral conic classifiers have become popular since they perform better compared to the Support Vector Machines (SVMs). Cone vertex of polyhedral conic classifiers is an important parameter and it is generally taken as the mean of positive data in literature. In this paper, we studied optimally estimating the cone vertex to improve the accuracy of the polyhedral conic classifiers. The proposed method has been compared with the linear SVMs and polyhedral conic classifiers and extended polyhedral conic classifiers that fix the cone vertex to the mean of the positive data. Various real databases in *UCI Machine Learning Repository* and a synthetic data set have been used for comparison of these classification methods. Experimental results show that the proposed approach has good results in solving data classification problems.

Keywords —*cone vertex estimation, polyhedral conic classifiers, extended polyhedral conic classifiers, classification.*

I. GİRİŞ

Sınıflandırma günümüzde örüntü tanıma, yapay zeka, istatistik, kavramsal psikoloji, görüntü analizi, tıp ve robotik gibi

bir çok bilim, endüstri, ticari ve askeri alanlarda kullanılan önemli tekniklerinden biridir. Bu alanlarda ortaya çıkan sınıflandırma sorunları karmaşık ve çoğunlukla çok kapsamlı verilerin kullanılması ile ilgilidir. Sınıflandırma konusu tüm alanlarda büyük önem taşıdığından her geçen gün daha gelişmiş sınıflandırma yöntemleri sunulmaktadır.

Son zamanlarda yapılan bir çok çalışmada geleneksel sınıflandırıcılardan vazgeçilip uygun kayıp fonksiyonu kullanarak daha sıkı bir pozitif sınıf modeli tasarlamak için tek-sınıf sınıflandırıcıları önerilmiştir. Bu sınıflandırıcılar pozitif sınıf modelini sadece pozitif sınıf örnekleri ile öğrenebilmektedir. Bu yönde bir çok yöntem önerilmiştir. Destek Vektör Veri Tanımı yöntemi [1] pozitif sınıf örneklerinin çoğunluğunu içeren kompakt bir hiperküre bulmaktadır. Bir başka çalışmada ardışık konveks model sınıflandırıcı kullanılarak negatif örneklerden kompakt pozitif bölge ayrılmaktadır [2]. Pozitif sınıf örneklerine en iyi yerleşen ve aynı zamanda diğer negatif örneklerden mümkün olduğunca uzak bir hiperdüzlem bulmak için Genelleştirilmiş Özdeğer Yakınsal Destek Vektör Makine sınıflandırıcı [3]'de önerilmiştir. [4]'teki çalışma ile pozitiflerin altgrup negatif örneklere karşı sınıflandırılması için hiperdüzlem sınıflandırıcı kümesi optimize edilerek çok yüzlü kabul bölgeleri öğrenilmiştir. Pozitif verileri yaklaşık olarak sınırlayabilen çok yüzlü oluşturmak için bir çok çeşitli yöntem vardır [5]-[7]. Bu çalışmaların içinde [6]'da verilen çok yüzlü konik sınıflandırıcılar basitliği ile dikkat çekmektedir. Bu yöntem [8]'de genişletilerek bir çok bilgisayarlı görü problemine başarı ile uygulanmış ve Destek Vektör Makinelerine oranla çok daha iyi sonuçlar alınmıştır.

Ayrıca, çok yüzlü konik sınıflandırıcılar için [9]-[11]'de farklı çalışmalar mevcuttur. [10] çalışmasında L1 normlu genişletilmiş çok yüzlü konik sınıflandırıcılar için iki aşamalı bir yaklaşım önerilmiştir ve [11]'deki çalışma ile çok yüzlü konik ayırmaya bağlı doğrusal parçalı bir sınıflandırıcı geliştirilmiştir. Bu çalışmadaki amaç, koni tepe noktası tahmini yapılarak daha başarılı çok yüzlü konik sınıflandırıcı oluşturmaktır. Koni tepe noktası tahmininde kullanılan çok yüzlü sınıflandırıcı modeli L1 koni vasıtasıyla hiperdüzlem bölümlerden ve onların genişletilmiş versiyonlarından oluşmaktadır ve pozitif veriler için kabul bölgesini tanımlamak için kullanılmaktadır. Bu sınıflandırıcının kullanma nedeni ise, pozitif sınıflarını

negatiflerden ayırmak için daha uygun ve kompakt konveks bölge şekillerini oluşturmasından kaynaklanmaktadır.

II. YÖNTEM

A. Çok yüzlü konik sınıflandırıcı

\mathbb{R}^n 'de iki ayrık kümeyi ayırmak için kullanılan çok yüzlü fonksiyonu *Polyhedral Conic Function (PCF)* $f_{(\omega,\gamma,c,b)} : \mathbb{R}^n \rightarrow \mathbb{R}$ (1)'de tanımlanmıştır ve bu fonksiyonu kullanan sınıflandırıcı *Polyhedral Conic Classifier (PCC)* olarak adlandırılmaktadır.

$$f_{(\omega,\gamma,c,b)}(x) = \omega^T(x-c) + \gamma \|x-c\|_1 - b \quad (1)$$

Burada $\omega, c \in \mathbb{R}^n$ ve $\gamma, b \in \mathbb{R}$ olarak tanımlanmıştır. Bu denklemde ω ve x 'in sayısal çarpımı $\omega^T x = \omega_1 x_1 + \dots + \omega_n x_n$ şeklinde ve $x \in \mathbb{R}^n$ vektörünün L1 normu ise $\|x\|_1 = |x_1| + \dots + |x_n|$ şeklinde verilmiştir. Denklem (1)'de tanımlanan $f_{(\omega,\gamma,c,b)}$ fonksiyon grafiği $(c, -b)$ tepe noktasına sahip çok yüzlü bir koni oluşturmaktadır [6].

B. Genişletilmiş çok yüzlü konik sınıflandırıcı

Bu çalışmada [8]'de önerilen genişletilmiş çok yüzlü konik sınıflandırıcılar *Extended Polyhedral Conic Classifier (EPCC)* kullanılmıştır. Bu sınıflandırıcı modeli [6]'da önerilen çok yüzlü konik fonksiyonundan - özellikle L1 konileri aracılığıyla hiperdüzlem bölümleri- ve pozitif kabul edilen bölgelerini tanımlamak için onların uzantısından yararlanarak oluşturulmuştur. Seçilen model, nispeten iyi lokalize edilmiş pozitif sınıfın daha kapsamlı negatif sınıftan ayırmak için uygun bir takım kompakt konveks bölge şekillerini sağlamaktadır. Genişletilmiş çok yüzlü konik fonksiyonu aşağıdaki gibi tanımlanır.

$$f_{(\omega,\gamma,c,b)}(x) = \omega^T(x-c) + \gamma^T |x-c| - b \quad (2)$$

Bu fonksiyonda $c \in \mathbb{R}^d$ koni tepe noktası, $\omega \in \mathbb{R}^d$ ağırlık vektörü ve b ise ofsettir. Genişletilmiş çok yüzlü konik fonksiyonunda $|u| = (|u_1|, \dots, |u_d|)^T$ bileşen yönlü modülü ve $\gamma \in \mathbb{R}^d$ ilgili ağırlık vektörünü belirtmektedir.

Belirtilen çok yüzlü konik fonksiyonlarında (1) ve (2), pozitifler için $f(x) < 0$ ve negatifler için $f(x) > 0$ karar bölgeleri kullanılmıştır. Benzer şekilde, boşluk tabanlı eğitim yöntemleri pozitifler için $f(x) \leq -1$ ve negatifler için ise $f(x) \geq +1$ uygulamaktadır. PCC yöntemi için $b > 0$, $\gamma > 0$, $\|\omega\|_\infty < \gamma$ ($\|\omega\|_\infty = \max_{i=1}^d |u_i|$, ∞ norm), $f(x) < \tau$ bölgesi konveks, \mathbb{R}^d 'de kompakttır ve c tepe noktasını içermektedir. Benzer olarak, EPCC için $b > 0$, $\gamma > 0$, $|\omega| < \gamma_i$, $i = 1, \dots, d$ ve her τ için $f(x) < \tau$ bölgesi tekrar konveks, kompakt ve c tepe noktasını içermektedir. PCC ve EPCC yapıları bu şekilde uygulanmıştır. PCC'de girdi olarak alınan öznelik vektörleri üzerinden boşluk tabanlı sınıflandırıcıları tanımlamak için öznelik vektörü $\tilde{x} \equiv \begin{pmatrix} x-c \\ \|x-c\|_1 \end{pmatrix} \in \mathbb{R}^{d+1}$ ve ağırlık vektörü ise $\tilde{\omega} \equiv \begin{pmatrix} -\omega \\ -\gamma \end{pmatrix} \in \mathbb{R}^{d+1}$ vektörüne arttırılmaktadır ve $\tilde{b} = b$ olarak kullanılmaktadır. Daha sonra, PCC karar fonksiyonu pozitifler

için bilinen doğrusal DVM formu $\tilde{\omega}^T \tilde{x} + \tilde{b} > 0$ ve negatifler için $\tilde{\omega}^T \tilde{x} + \tilde{b} < 0$ kullanılmaktadır.

Benzer şekilde EPCC yöntemi için öznelik vektörü ise

$$\tilde{x} \equiv \begin{pmatrix} x-c \\ |x-c| \end{pmatrix} \in \mathbb{R}^{2d} \quad \text{ve ağırlık vektörü için} \quad \tilde{\omega} \equiv \begin{pmatrix} -\omega \\ -\gamma \end{pmatrix} \in \mathbb{R}^{2d}$$

vektörüne arttırılmaktadır ve $\tilde{b} = b$ olarak alınmıştır. Pozitifler için yine aynı DVM formu $\tilde{\omega}^T \tilde{x} + \tilde{b} > 0$ ama bu sefer $2d$ boyutunda kullanılmaktadır. PCC ve EPCC için yukarıdaki ∓ 1 boşluklar bilinen ± 1 DVM boşluklarına aktarılır ve maksimum boşluk eğitimi için standart DVM kullanılmasına izin verir. Bu nedenle arttırılmış öznelik vektörlerinde bilinen DVM karesel programlamanın çalışması yeterli olacaktır.

$$\begin{aligned} \arg \min_{\tilde{\omega}, \tilde{b}} \quad & \frac{1}{2} \tilde{\omega}^T \tilde{\omega} + C_+ \sum_i \xi_i + C_- \sum_j \xi_j \\ \text{s.t.} \quad & \tilde{\omega}^T \tilde{x}_i + \tilde{b} + \xi_i \geq +1, \quad i \in I_+ \\ & \tilde{\omega}^T \tilde{x}_j + \tilde{b} - \xi_j \leq -1, \quad j \in J_- \end{aligned} \quad (3)$$

Yukarıdaki eşitlikte I_\pm pozitif ve negatif verilerin indekslerini, ξ kısıtları ihlal eden örnekler için arttıran yapay değişkenleri ve C_\pm ilgili ceza ağırlığını göstermektedir.

C. Koni tepe noktası tahmini

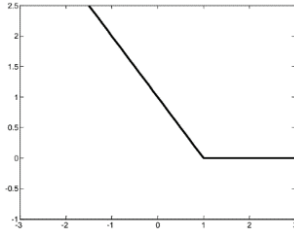
Genel olarak PCC ve EPCC yöntemlerinde tepe noktası (c parametresi) pozitif verilerin ortalaması olarak alınmıştır. Bu çalışmamızdaki amaç çok yüzlü konik sınıflandırıcılar ile daha başarılı sonuç elde etmek için koni tepe noktasını optimal şekilde belirlemektir. Çok yüzlü konik sınıflandırıcının koni tepe noktası tahmini için c parametresinin tahmin edilmesi gerekmektedir. Koni tepe noktası ($c = X_{pos} \alpha$) pozitif örneklerin doğrusal kombinasyonu olarak yazılabilir. Burada X_{pos} her sütununda pozitif örnekleri içeren bir matristir ve α ise doğrusal kombinasyon katsayılarını göstermektedir. Bu durumda değişken parametre α tahmin edilerek koni tepe noktası tahmin edilebilir. Diğer sınıflandırıcı parametreleri ile birlikte koni tepe noktasını da tahmin için eşitlik (4) kullanılabilir. Fakat bu fonksiyon tüm parametrelere göre konveks değildir. Bu sebeple bu problemin çözümü için sıra ile değişen optimizasyon (alternating optimization) yöntemi kullanılmıştır. Buna göre ilk olarak sabit bir tepe noktası kullanılarak DVM algoritması kullanılmak suretiyle ω, γ, b parametreleri bulunur. Daha sonra bulunan bu parametreler kullanılarak optimizasyon fonksiyonunun α parametresine göre türevi alınarak Stochastic Gradient yöntemi ile en iyi koni tepe noktası bulunur ve bu iteratif olarak devam ettirilir. Elde edilen güncel çok yüzlü koni parametreleri ile bir önceki döngüdeki değerlerinin arasındaki fark çok küçük olduğunda döngü sonlanır ve elde edilmiş olan güncel α parametreleri koni tepe noktasını belirler. Koni tepe noktası tahmini için L1 normlu genişletilmiş çok yüzlü konik sınıflandırıcı kullanılmıştır ve geliştirilen yeni yöntem EPCC-L1-C yöntemi olarak adlandırılmıştır. Tepe noktası sabit bir vektöre atandığında ω, γ, b parametreleri klasik DVM algoritması ile kolayca bulunmaktadır. Bu sebeple aşağıda sadece koni tepe noktasının optimizasyonu anlatılmıştır.

$$\min_{\omega, \gamma, \alpha} \frac{1}{2} \omega^T \omega + C_+ \sum_{i=1}^{n_+} H(y_i [\omega^T (x_i - X_{pos} \alpha) + \gamma^T |x_i - X_{pos} \alpha|_1] + b - 1) + C_- \sum_{i=1}^{n_-} H(y_i [\omega^T (x_i - X_{pos} \alpha) + \gamma^T |x_i - X_{pos} \alpha|_1] + b - 1) + \frac{1}{2} \alpha^T \alpha \quad (4)$$

TABLO I. OBJEKTİF FONKSİYON PARAMETRELERİ

Parametre	Boyut	Tanım
i	\mathbb{R}	İterasyon sayısı
n_{\pm}	\mathbb{R}	Pozitif ve negatif örnek sayısı
y_i	± 1	Sınıf etiketi
C_{\pm}	\mathbb{R}	Ağırlık çarpanı
ω, γ	\mathbb{R}^d	Ağırlık vektörleri
x_i	\mathbb{R}^d	Rastgele seçilmiş very örneği
X_{pos}	$\mathbb{R}^d \times \mathbb{R}^{n_+}$	Positif örnekleri içeren matris
α	\mathbb{R}^{n_+}	Doğrusal kombinasyon katsayıları vektörü
$c = X_{pos} \alpha$	\mathbb{R}^d	Koni tepe noktası
b	\mathbb{R}	Ofset (sapma)
$ u _1$	$[u_1 \dots u_d]$	L1 norm bileşen yönlü modülü

1) *Objektif fonksiyon:* EPCC-L1-C objektif fonksiyonu aşağıda verilmiş olup ve bu fonksiyonu F olarak gösterelim (4). Bu objektif fonksiyondaki parametrelerin boyutları ve açıklamaları Tablo (I)'de bulunmaktadır. Objektif fonksiyonunda verilen $H(t) = \max(0, 1-t)$ DVM'lerde kullanılan Hinge fonksiyonunu ifade etmektedir (Şekil 1). EPCC-L1-C objektif fonksiyonunda kullanılan Hinge kaybı için aşağıda verilen eşitsizlik her veri noktasına ayrı ayrı uygulanarak kontrol edilmektedir.



Şekil 1. Hinge kaybı.

$$\begin{cases} y_i [\omega^T (x_i - X_{pos} \alpha) + \gamma^T |x_i - X_{pos} \alpha|_1] < 1 & (1.durum) \\ y_i [\omega^T (x_i - X_{pos} \alpha) + \gamma^T |x_i - X_{pos} \alpha|_1] > 1 & (2.durum) \end{cases}$$

Eğer Hinge kaybı miktarı 1'den küçük ise (1.durum) objektif fonksiyonuna türev işlemi uygulanarak yöntem devam etmektedir. Eğer Hinge kaybı 1'den büyük ise (2.durum) kayıp olmadığından yöntem devam edilemez ve başka bir veri noktasına geçilir.

2) *Fonksiyon türevi:* EPCC-L1-C fonksiyonun türevini almak için öncelikle (4)'te verilen objektif fonksiyonun en önemli bölümü olan $X_{pos} \alpha$ çarpımını ele alarak başlayalım.

$$X_{pos} \alpha = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{n_{pos}1} \\ x_{12} & x_{22} & \dots & x_{n_{pos}2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1d} & x_{2d} & \dots & x_{n_{pos}d} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_{pos}} \end{bmatrix} = \begin{bmatrix} x_{11}\alpha_1 + x_{21}\alpha_2 + \dots + x_{n_{pos}1}\alpha_{n_{pos}} \\ x_{12}\alpha_1 + x_{22}\alpha_2 + \dots + x_{n_{pos}2}\alpha_{n_{pos}} \\ \vdots \\ x_{1d}\alpha_1 + x_{2d}\alpha_2 + \dots + x_{n_{pos}d}\alpha_{n_{pos}} \end{bmatrix}$$

Bir sonraki adım, rastgele seçilmiş x_i veri vektörünü $X_{pos} \alpha$ çarpım vektöründen çıkarma işlemidir. Elde edilen bu vektör objektif fonksiyonun L1 norm terimini oluşturur ve L1 norm türevinin uygulanması gereken bölümdür.

$$(x_i - X_{pos} \alpha) = \begin{bmatrix} x_{i1} - (x_{11}\alpha_1 + x_{21}\alpha_2 + \dots + x_{n_{pos}1}\alpha_{n_{pos}}) \\ x_{i2} - (x_{12}\alpha_1 + x_{22}\alpha_2 + \dots + x_{n_{pos}2}\alpha_{n_{pos}}) \\ \vdots \\ x_{id} - (x_{1d}\alpha_1 + x_{2d}\alpha_2 + \dots + x_{n_{pos}d}\alpha_{n_{pos}}) \end{bmatrix}$$

L1 norm bölümünün türevi aşağıdaki gibi olacaktır.

$$|x_i - X_{pos} \alpha|_1 = \text{sign}(x_i - X_{pos} \alpha) * (x_i - X_{pos} \alpha)' \quad (5)$$

Bu eşitlikte (*) çarpımı MATLAB programlama dilindeki gibi iki vektörün her bir teriminin çarpımını belirtmektedir. L1 norm türevindeki işaret belirleme (sign) için [13] çalışmasında verilen yöntem kullanılmıştır. Bu yöntemle göre elde edilen genel işaretlenme aşağıdaki gibi olmaktadır. Burada, $X_{pos}(i,:)$ terimi X_{pos} matrisinin i 'nci satır vektörüdür.

$$\text{sign}(x_i - X_{pos} \alpha) = \begin{cases} +(x_{i1} - x_{11}\alpha_1 - \dots - x_{n_{pos}1}\alpha_{n_{pos}}) & x_{ii} - X_{pos}(i,:)\alpha > 0 \\ -(x_{i1} - x_{11}\alpha_1 - \dots - x_{n_{pos}1}\alpha_{n_{pos}}) & x_{ii} - X_{pos}(i,:)\alpha < 0 \end{cases}$$

Bu durumda L1 norm bölümü için elde edilen türev aşağıdaki gibidir.

$$|x_i - X_{pos} \alpha|_1' = \begin{cases} +(X_{pos}(i,:)) & x_{ii} - X_{pos}(i,:)\alpha > 0 \\ -(X_{pos}(i,:)) & x_{ii} - X_{pos}(i,:)\alpha < 0 \end{cases}$$

Objektif fonksiyondaki diğer terimlerin türevi ise aşağıda belirtilmiştir.

$$\begin{aligned} \frac{\partial}{\partial \alpha} \left(\frac{1}{2} \omega^T \omega \right) &= 0, & \frac{\partial}{\partial \alpha} (b) &= 0, & \frac{\partial}{\partial \alpha} \left(\frac{1}{2} \alpha^T \alpha \right) &= 0 \\ \frac{\partial}{\partial \alpha} (\omega^T (x_i - X_{pos} \alpha)) &= -X_{pos}^T \omega \end{aligned}$$

1.durum: *Hinge* < 1 durumu için genel fonksiyon türevi aşağıda verilmiştir.

$$\nabla F = \begin{cases} C_+ \sum_{i=1}^{n_+} y_i [-X_{pos}^T \omega - \gamma^T (X_{pos}(i,:))] / n + & x_{ii} - X_{pos}(i,:)\alpha > 0 \\ C_- \sum_{i=1}^{n_-} y_i [-X_{pos}^T \omega - \gamma^T (X_{pos}(i,:))] / n + \alpha / n & \\ C_+ \sum_{i=1}^{n_+} y_i [-X_{pos}^T \omega + \gamma^T (X_{pos}(i,:))] / n + & \\ C_- \sum_{i=1}^{n_-} y_i [-X_{pos}^T \omega + \gamma^T (X_{pos}(i,:))] / n + \alpha / n & x_{ii} - X_{pos}(i,:)\alpha < 0 \end{cases}$$

2. durum: *Hinge* > 1 durumu için genel fonksiyon türevi aşağıda verilmiştir.

$$\nabla F = \alpha / n$$

3) *Stochastic Gradient Descent yöntemi:* Objektif fonksiyonun α 'ya göre türevi alındıktan sonra α 'yı minimize etmek gerekir. Bu minimizasyon işlemi için (6)'da verilen Stochastic Gradient Descent adımları kullanılır. Bu denklemde ε için sifıra yakın küçük bir değer kullanılmıştır. $\text{steplength}(s) = 0.000001$, $t = 750$ rastgele seçilmiş veri noktalarının döngü sayısı ve $\varepsilon = \text{steplength} / t$ olarak alınmıştır.

$$\alpha_{\text{yeni}} = \alpha_{\text{eski}} - \varepsilon \nabla F \quad (6)$$

III. DENEYLER

A. Sentetik veri tabanı üzerindeki deneyler

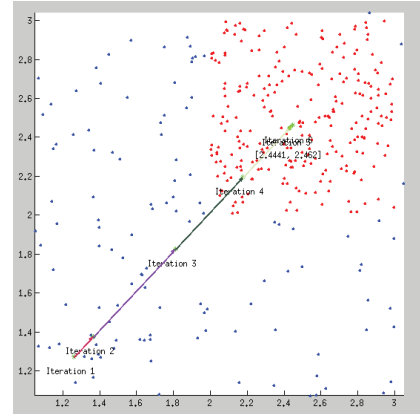
Önerilen yöntem ilk olarak sentetik veri tabanı üzerinde test edilmiştir. Sentetik veri tabanı için 250 pozitif örneğin kare şeklinde ortada ve 700 negatif örneğin ise pozitiflerin etrafında rastgele dağılacak şekilde 2 boyutlu bir veri kümesi oluşturulmuştur. Bu veri için optimal tepe noktası gerçekten de pozitif verilerin tam ortası olup, önerilen algoritmanın bu değere yakınsayıp yakınsamayacağı kontrol edilmiştir. Sentetik veri kümesinde pozitif veriler her iki x ve y koordinatında 2-3 koordinat değerleri arasında rastgele dağılmıştır ve ortalaması ise $(x_{mean}, y_{mean}) = (2.5, 2.5)$ 'dir. EPCC-L1-C yönteminin uygulanması ile birlikte beklenen sonuç her iterasyon sonucunda elde edilen değer giderek pozitif verilerin ortalamasına yaklaşmasıdır. Algoritmanın başında tepe noktası $(0,0)$ olarak verilmiştir. Sonuç olarak 6 iterasyon sonucunda elde edilen koordinat değeri $(x_{L1}, y_{L1}) = (2.4441, 2.4620)$ olmuştur ve hedef koordinatına yakındır (Şekil 2). Bu da önerilen sistemin doğru bir şekilde optimal tepe noktasını tahmin ettiğini göstermektedir.

B. Gerçek veri tabanları üzerindeki deneyler

PCC-L1, EPCC-L1 ve DVM gibi sınıflandırıcılar ile geliştirilen koni tepe noktası tahmin yöntemi EPCC-L1-C karşılaştırılmıştır. Bu karşılaştırma için *UCI Machine Learning Repository*'deki *IRIS*, *LR*, *VOC* ve *WDBC* veri tabanları kullanılmıştır. Bu veritabanları sırayla; 4, 16, 6 ve 30 öznitelikten oluşmaktadır. PCC-L1 ve EPCC-L1 yöntemlerinde koni tepe noktası olarak pozitif verilerin ortalaması alınırken EPCC-L1-C yönteminde koni tepe noktası tahmini yapılmıştır. Ayrıca, eğitim ve test kümeleri 10 kat çapraz geçerlilik kuralına uygun olarak oluşturulmuştur. Başlangıçta α için pozitif verilerin ortalaması referans alınmıştır. Bu deney için, DVM'deki C parametre değeri 10 olarak alınmıştır ve sonuçlar Tablo (II)'de verilmiştir. Bu veri tabanları için EPCC-L1-C yöntem sonucu ile diğer yöntem sonuçları karşılaştırıldığında (Tablo II), EPCC-L1-C yöntemi yüksek doğruluk oranını elde etmiştir. Ayrıca tüm çok yüzlü sınıflandırıcıları DVM sınıflandırıcılarına göre daha iyi sonuçlar vermiştir. Çalışma hızı açısından ise koni tepe noktası tahmini yapılarak elde edilen sınıflandırıcının karmaşık yapıya sahip olması ve bir çok optimizasyon işlemi içerdiğinden DVM sınıflandırıcı modeline göre yaklaşık 10 kat kadar daha yavaş sonuç verdiği gözlemlenmiştir.

IV. SONUÇ

Çok yüzlü konik sınıflandırıcılar, pozitif sınıf verilerini negatiflerden ayırmak için daha uygun ve kompakt konveks bölge şekillerini oluşturması, sınıflandırıcı modelin bütünsel optimum sonuç üretmesi, büyük problemlere uygun ölçekte olması ve gürbüz boşluk bazlı cost fonksiyonu kullanarak çakışmayı önlemesi gibi özelliklere sahip olması nedeni ile uygun bir sınıflandırıcı modelidir. Gürbüz çok yüzlü konik sınıflandırıcı elde etmek için koni tepe noktası tahmini büyük önem taşımaktadır. Bu tahmin algoritmasında kayıp fonksiyonu ve L1 normlu koniden yararlanılmıştır. Gerçek ve sentetik veri tabanları üzerinde yapılan deney sonuçları sunulan yaklaşımın veri sınıflandırma problemlerinin çözümünde etkili sonuçlar verdiğini göstermiştir.



Şekil 2. Sentetik veri kümesine uygulanan EPCC-L1-C yönteminin iteratif olarak optimal noktaya yakınsaması.

TABLO II. GERÇEK VERİ TABANLARINA UYGULANAN YÖNTEM SONUÇLARI

Yöntem	Veri Tabanı İçin Ortalama Doğruluk Oranı (%)			
	IRIS	LR	VOC	WDBC
PCC-L1	95.99	65.50	77.72	97.37
EPCC-L1	95.99	75.41	82.88	97.86
DVM	94.70	59.80	79.42	97.70
EPCC-L1-C	98.66	79.63	84.30	97.91

KAYNAKLAR

- [1] D. Tax, R. Duin, "Support vector data description", *Machine Learning*, 2004.
- [2] H. Cevikalp, B. Triggs, "Efficient object detection using cascades of nearest convex model classifiers", *CVPR*, 2012.
- [3] O. Mangasarian, E. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.
- [4] M. Dundar, vd. "Polyhedral classifier for target detection: a case study: colorectal cancer", *In Proceedings of the 25th International Conference on Machine Learning*, 2008.
- [5] A. Bagirov, J. Ugon, D. Webb, "An efficient algorithm for the incremental construction of a piecewise linear classifier", *Information Systems*, 2011.
- [6] R. Gasimov, G. Ozturk, "Separation via polyhedral conic functions", *Optimisation Methods and Software*, 2006.
- [7] N. Manwani, P. Sastry, "Learning Polyhedral Classifiers Using Logistic Function", *ACML*, 2010.
- [8] H. Cevikalp, B. Triggs, "Polyhedral conic classifiers for visual object detection and classification", *CVPR*, 2017.
- [9] A. Bagirov, vd., "A novel piecewise linear classifier based on polyhedral conic and max-min separabilities", *TOP*, 2013.
- [10] G. Ozturk, R. Kasimbeyli, "A Two Objective Classification Approach based on Conic Functions", *International Conference on Multiple Criteria Decision Making*, 2013.
- [11] G. Ozturk, A. Bagirov, R. Kasimbeyli, "An incremental piecewise linear classifier based on polyhedral conic separation", *Machine Learning*, 2015.
- [12] H. Cevikalp, M. Elmas, "Robust transductive support vector machines", *SIU*, 2016.
- [13] F. Bach, vd., "Convex optimization with sparsity-inducing norms", *Optimization for Machine Learning*, 2011.