



Multi center polyhedral conic classifiers that can classify complex data

Halil Sağlamlar*

Eskişehir Osmangazi University, Electrical Electronics Engineering, 26040 Odunpazarı/ Eskişehir/ Türkiye

Highlights:

- One class classification in machine learning
- Using multi centers instead of single center in conic classifiers
- Using single classifier for classification of positive data that clustered in different regions

Keywords:

- Polyhedral conic classifiers
- Multi center polyhedral conic classifiers
- Kernel SVM
- k-means
- Classification.

Article Info:

Research Article
Received: 24.09.2020
Accepted: 14.03.2021

DOI:

10.17341/gazimmfd.799556

Correspondence:

Author: Halil Sağlamlar
e-mail:
hsaglamlar@gmail.com
phone: +90 507 882 7226

Graphical/Tabular Abstract

Polyhedral conic classifiers (PCC), compared to support vector machines (SVM), stand out for their success while keeping simplicity. In these classifiers a conic function with a vertex point is used. Vertex point which is chosen well provides compact decision boundaries for positive samples that are clustered together. In cases where positive data belonging to the same class are clustered in different regions as seen in Figure A, a single classifier is not enough and more than one classifier is needed. In this study, a novel multicenter polyhedral conic classifier (MCPCC) is developed to use only one classifier to classify positive data clustered in different regions. Experiments are performed using proposed method and related methods for comparison. In the results, it was seen that the new MCPCC method gave successful and promising results.

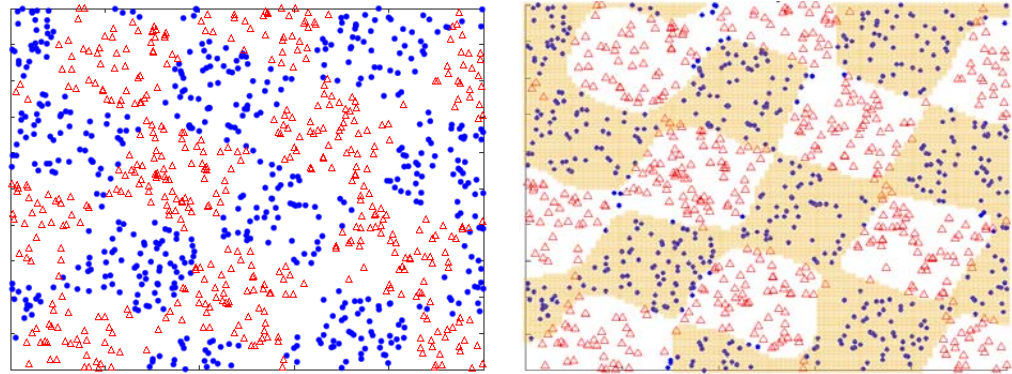


Figure A. (left) A data set whose layout is a rotated chess board. Blue dots are positive samples that are clustered in several separated regions. Red triangles are negatives. (right) Orange area is positive acceptance region using MCPCC classifier.

Purpose: This study focuses on creating a scalable and successful classifier that can construct non-linear decision boundaries while keeping SVM simplicity and fast run times when the positive data clustered different regions.

Theory and Methods:

In this new method we increased the number of vertex points instead of using multiple PCC classifiers and estimated highly nonlinear decision boundaries successfully using a single PCC classifier. As seen on Figure A, a very complex positive data is successfully discriminated from negatives with only one classifier. In MCPCC multiple centers are obtained using k-means algorithm. We used a separate validation data and computed the performance results by gradually increasing the value of number of centers, n . We set n to the value where the improvement in accuracy is negligible and the accuracy gets saturated. It is obvious that increase in n will increase the workload.

Results and Conclusion:

MCPCC method and related other methods were tested on several synthetic and real datasets (CIFAR-10, ESOGU-285, Liver, Splice and Svmguide). MCPCC got the best or second-best scores in these experiments. In ESOGU-285, MCPCC had third best score while EPCC had the best result. This shows that in ESOGU-285 positives are not clustered in separate regions. With tests performed we can put forward that MCPCC classifier can provide highly non-linear decision boundaries while keeping PCC simplicity and run time speed. It has also scalable with few parameters. Especially MCPCC is more effective than other methods when the positive data clustered separate regions.



Karmaşık verileri sınıflandırabilen çok merkezli çok yüzlü konik sınıflandırıcılar

Halil Sağlamlar*^{ID}

Eskişehir Osmangazi Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, 26040, Odunpazarı, Eskişehir

Ö N E Ç İ K A N L A R

- Makine öğreniminde tek sınıf sınıflandırma
- Konik sınıflandırıcılarda tek merkez yerine çok merkez kullanılması
- Farklı bölgelerde kümelenen pozitif verilerin tek bir sınıflandırıcı kullanılarak sınıflandırılması

Makale Bilgileri

Araştırma Makalesi

Geliş: 24.09.2020

Kabul: 14.03.2021

DOI:

10.17341/gazimmfd.799556

Anahtar Kelimeler:

Ok yüzlü konik sınıflandırıcılar, çok merkezli çok yüzlü konik sınıflandırıcılar, kernel SVM, k-means, sınıflandırma

ÖZ

Çok yüzlü konik sınıflandırıcılar destek vektör makineleriyle karşılaştırıldığında başarısıyla ve basitliği korumasıyla ön plana çıkmaktadır. Bu sınıflandırıcılarda bir tepe noktası olan koni fonksiyonu kullanılmaktadır. İyi seçilmiş bir tepe noktası bir bölgede toparlanan pozitif verilerin sıkıca çevrelenmesini sağlar. Aynı sınıfa ait pozitif verilerin farklı bölgelerde kümelendiği durumlarda ise tek bir sınıflandırıcı yeterli olmamakta, birden fazla sınıflandırıcıya ihtiyaç duyulmaktadır. Bu çalışmada tek bir sınıflandırıcı kullanarak farklı bölgelerde kümelenen pozitif verileri sınıflandırmak amacıyla çok merkezli çok yüzlü konik sınıflandırıcı (MCPCC) yöntemi geliştirilmiştir. Karşılaştırma yapabilmek için önerilen sınıflandırıcı ve benzer yöntemlerle deneyler gerçekleştirilmiştir. Elde edilen sonuçlarda MCPCC yönteminin başarılı sonuçlar verdiği gözlemlenmiştir.

Multi center polyhedral conic classifiers that can classify complex data

H I G H L I G H T S

- One class classification in machine learning
- Using multi centers instead of single center in conic classifiers
- Using single classifier for classification of positive data that clustered in different regions

Article Info

Research Article

Received: 24.09.2020

Accepted: 14.03.2021

DOI:

10.17341/gazimmfd.799556

Keywords:

Polyhedral conic classifiers, multi center polyhedral conic classifiers, kernel SVM, k-means, classification

ABSTRACT

Polyhedral conic classifiers, compared to support vector machines, stand out for their success while keeping simplicity. In these classifiers, a conic function with a vertex point is used. A carefully chosen vertex point provides compact decision boundaries for positive samples that are clustered together. In cases where positive data belonging to the same class are clustered in different regions, a single classifier is not enough, and more than one classifier is needed. In this study, a novel multicenter polyhedral conic classifier (MCPCC) is developed to use only one classifier to classify positive data clustered in different regions. Experiments are performed using proposed method and related methods for comparison. In the results, it is demonstrated that the new MCPCC method gives successful and promising results.

1. GİRİŞ (INTRODUCTION)

Makine öğrenmesi son yıllarda popülerliği giderek artan bir konu olmuştur. Çok farklı teknikler [1, 2] kullanılarak, farklı alanlarda uygulamalar [3, 4] geliştirilmiştir. Geniş marjlı sınıflandırıcılar (large margin classifiers) ise makine öğreniminde yaygın olarak kullanılmıştır [5, 6]. Son yıllarda derin öğrenme yöntemiyle daha başarılı sonuçlar elde edilse de bazı derin öğrenme algoritmaları geniş marjlı sınıflandırıcılar ile birlikte kullanılmaktadır. Geniş marjlı sınıflandırıcıların en bilineni Destek Vektör Makineleridir (Support Vector Machines, SVM) [7]. SVM yöntemi, öznitelik uzayında bir hiperdüzlem bularak iki sınıfın örneklerini birbirinden ayırmaktadır. Geniş marj tabanlı sınıflandırıcılarda bilinmeyen sınıfa ait örnekler, pozitif örneklerden çok uzakta olsa bile, başka bir karar sınırı olmadığından bir sınıfa dahil edilebilmektedir. Bu tarz sorunların üstesinden gelmek amacıyla tek sınıf sınıflandırıcılar ön plana çıkmıştır. Tek sınıf sınıflandırıcılar, negatif örnekler olmasa dahi, karar sınırlarıyla pozitif örnekleri sıkıca çevrelemeyi amaçlamaktadır. Bu sayede bilinmeyen sınıfa ait örnekler daha iyi dışlanabilmektedir.

Tek sınıf sınıflandırıcı olarak, [8]'de Destek Vektör Veri Açıklaması (Support Vector Data Description, SVDD) yöntemi önerilmiş olup bu yöntemde pozitif örnekleri çevreleyen bir hiperküre bulunması amaçlanmıştır. Diğer bir yöntemde yüz ve insan tespiti için, kademeli bir şekilde pozitif bölgeleri negatif bölgelerden ayıran dışbükey modeller önerilmektedir [9]. Genelleştirilmiş Özdeğer Proksimal Destek Vektör Makinesi (Generalized Eigenvalue Proximal Support Vector Machine, GEPSVM) yöntemi, pozitif örnekleri en iyi şekilde ayırırken, negatif örneklerden de olabildiğince uzak olan bir hiperdüzlem bulmaktadır [10]. Benzer şekilde, İkiz Destek Vektör Makinesi (Twin Support Vector Machine, TSVM) yöntemi önerilmiştir [11]. TWSM sınıflandırıcısı en iyi yerleşen bir hiperdüzlemi, kuadratik programlama probleminin çözümüyle bulmaktadır. En iyi yerleşen hiperdüzlemi bulmaya çalışan başka yöntemlerde önerilmiştir [12]. Ekleme Çekirdek (Additive Kernel) [13] ile Rastgele Öznitelik (Random Features) [14] yöntemleri, örnekleri daha büyük boyutlu uzaya dönüştürerek doğrusal olmayan bir ayırma imkân sağlamakta ve pozitif örnekleri çevrelemektedir. Benzer şekilde pozitif örnekleri çevreleyen çok yüzlülerin önerildiği yöntemler de mevcuttur [15, 16]. Fakat bunlar büyük eğitim kümesi boyutuyla zorlanabilmekte, lokal optimum ve aşırı öğrenmeye maruz kalabilmekte, ilave olarak kümeleme veya etiketlemeye ihtiyaç duyabilmektedir. Bu durum ise büyük ölçekli uygulamalar için uygun olmamaktadır.

Çok Yüzlü Konik Fonksiyonlar (Polyhedral Conic Functions, PCF) da tek sınıf sınıflandırıcılardan birisidir [17]. Burada pozitif örnekler, PCF ile verilen koninin seviye kümelerine yerleştirilmeye çalışılmaktadır. Bu sayede kapalı, pozitif örnekleri çevreleyen sıkı karar sınırı elde edilebilmektedir. Sıkı ve kapalı karar sınırı sayesinde hem açık küme hem kapalı küme problemlerinde SVM gibi yarı

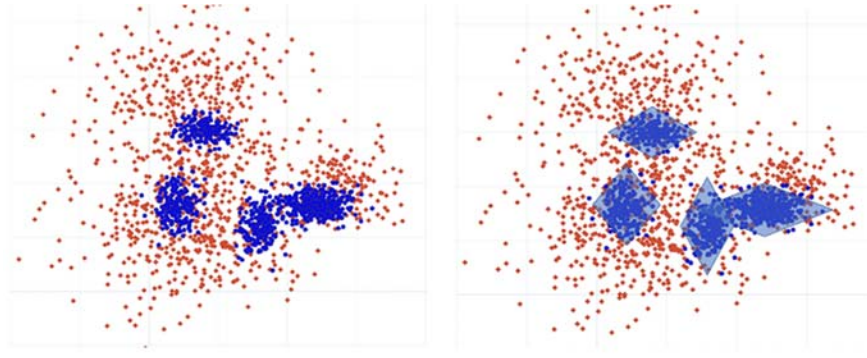
uzaylı karar sınırına sahip sınıflandırıcılardan daha yüksek başarımlar sağlanmaktadır. Burada kullanılan doğrusal programlama bu sınıflandırıcının büyük veri tabanlarında çalışmasını engellemektedir. Hata-ceza yaklaşımı yerine çok sınıflandırıcı bir yapıya sahip olması ise aşırı öğrenmeye neden olabilmektedir. Bu sorunların çözümü amacıyla Çok Yüzlü Konik Sınıflandırıcı (Polyhedral Conic Classifier, PCC) önerilmiştir [18, 19]. PCF'te yapılan değişiklikler ile de Genişletilmiş Çok Yüzlü Konik Fonksiyon (Extended Polyhedral Conic Function, EPCF) geliştirilmiş ve EPCF'i kullanan Genişletilmiş Çok Yüzlü Konik Sınıflandırıcı (Extended Polyhedral Conic Classifier, EPCC) önerilmiştir [18, 19]. PCC ve EPCC yöntemlerinin eğitim süreçleri doğrusal SVM gibi etkili bir konveks programlamayla yapılabilmekte olup test süreleri açısından da SVM kadar hızlı sonuçlar vermektedir.

PCF'te bir adet koni tepe noktası mevcuttur. Pozitif verilerin belli bir merkez etrafında toplanması durumunda, PCC yöntemi çok iyi bir sınıflandırma sağlamaktadır. Karşılaşılan bir problem ise Şekil 1'deki gibi pozitif örneklerin aynı uzayda farklı bölgelerde kümelenmesidir. Örneğin bir arabaya ait yandan, üstten, arkadan ve önden görünüm öznitelik uzayında farklı bölgelerde kümelenmektedir. Bu durumda her bir küme için ayrı bir sınıflandırıcı kullanılması gerekir. Fakat birden fazla sınıflandırıcı, eğitim ve test sürecini hem yavaşlatabilmekte hem de karmaşık hale getirebilmektedir. Bu çalışmada bu kapsamdaki problemleri çözmek amacıyla hem PCF basitliğini koruyan hem de tek bir sınıflandırıcı kullanan Çok Merkezli Çok Yüzlü Konik Sınıflandırıcı (Multi Center Polyhedral Conic Classifier, MCPCC) geliştirilmiştir. Çalışmamızda sadece bir sınıflandırıcı ile farklı merkezlerde kümelenen pozitif verilerin sınıflandırılması amaçlanmıştır. Bu kapsamda PCF'te bir tepe noktası yerine birden fazla tepe noktasının kullanılması önerilmiş ve yeni sınıflandırıcı MCPCC oluşturulmuştur. Bu kapsamda MCPCC yöntemlerinin başarımlarını incelemek amacıyla büyük ve küçük ölçekli veri tabanlarında deneyler yapılmıştır. Bu deneylerde farklı yöntemler de kullanılarak elde edilen sonuçlarla yöntemlerimizin başarımlarını incelenmiştir.

Makalenin geri kalan kısmı şu şekilde formatlandırılmıştır: 2. Bölümde benzer yöntemler hakkındaki literatür araştırması, 3. bölümde yöntemlerimiz ve teorik altyapısı belirtilmiştir. 4. bölümde yöntemlerimiz farklı veri tabanlarında ve farklı yöntemlerle karşılaştırılarak başarımlarını incelenmiştir. Çalışma 6. bölüm ile sonuçlandırılmıştır.

2. LİTERATÜR ARAŞTIRMASI (LITERATURE RESEARCH)

PCF fonksiyonunun ilk önerildiği çalışmadaki sınıflandırma yönteminde rastgele bir koni tepe noktası, c , seçilmekte ve doğrusal programlama ile PCF parametreleri bulunmuştur [17]. Bulunan parametrelerle oluşan koni, bütün pozitif örnekleri karar sınırı içine almadıysa, dışarıda kalan pozitif örneklerle yeni bir PCF için yukarıdaki adımlar



Şekil 1. Pozitif sınıf verilerinin farklı bölgelerde kümelendiği karmaşık yapıya sahip bir veri (mavi: pozitif örnekler, kırmızı: negatif örnekler)

(A complex dataset that positive samples are clustered in different regions; blue: positive, red: negative samples)

tekrarlanmaktadır. Dışarıda herhangi bir pozitif kalmadıysa yöntem durmaktadır. Diğer bir ifadeyle sınıfların tam olarak ayrıştırılmadığı durumlarda bu örnekler için SVM'deki gibi arttıran yapay değişkenler (slack variables) kullanmak yerine, ayrı PCF sınıflandırıcı eğitilmiştir. Bu sebeple sınıflandırıcı aşırı öğrenmeye (over-fitting) maruz kalmış ve test başarımı oldukça düşmüştür. Ayrıca bir sınıflandırıcı yerine aynı amaçla çok sayıda PCF sınıflandırıcı kullanılması test hızını etkilemektedir. Diğer bir problem de doğrusal programlama kullanan bu sınıflandırıcıyla çok veri sayısına sahip ve öznelik boyutu büyük bazı veri tabanlarında hafıza problemiyle karşılaşılmasıdır.

Diğer bir PCF yönteminde veri tabanındaki tüm sınıflar bire-tümü yöntemiyle ayrıştırılmıştır [20]. Her bir sınıfa ait örnekler k -merkezli kümeleme yöntemiyle k adet kümeye ayrılmış ve her bir küme için de bir PCF elde edilmiştir. Toplamda $k \times \text{sınıf sayısı}$ kadar PCF bulunmaktadır. Doğrusal programlamada her bir örnek kullanıldığından, $k \times \text{sınıf sayısı}$ kadar PCF eğitimi için her bir noktanın tekrar tekrar kullanılması gerekmektedir. Bu ise eğitim sürecini uzatmaktadır. Bundan dolayı büyük veri tabanları için doğrusal programlama yönteminin uygulanması kolay olmamaktadır. Bu sorunu aşmak için yeni bir yöntem (Incremental Conic Functions, ICF) önerilmiştir [21]. Buradaki yöntemde [20]'deki yönteminin sıkıntılına çözüm aranmıştır. PCF oluştururken kullanılmasına gerek olmayan örnekler çıkarılmış ve doğrusal programlamanın yükü azaltılmaya çalışılmıştır. Ayrıca küme sayısı k sabit alınmamış ve algoritmanın içinde en uygun değeri verecek şekilde hesaplanmıştır. Bu yeni yöntemle [20]'deki sonuçlara yakın sonuçlar elde edilmekle birlikte %64 oranında bir hızlanma sağlanmış ayrıca k değerinin önceden belirlenmesine gerek kalmamıştır.

Başka bir PCF yöntemi de (One-Class Polyhedral Conic Functions, O-PCF) [22]'de önerilmiştir. Tek sınıf bir sınıflandırıcıda eğitim sırasında sadece pozitif sınıfa ait örnekler kullanılmaktadır. Pozitif örnekler dışında başka bir veri bulunmamaktadır. Tek sınıflı sınıflandırıcı kullanan bu yöntemde, PCF'in karar sınırının büyüklüğü sınıflandırma hatasına göre belirlenerek en uygun PCF'in bulunması amaçlanmıştır.

Bir başka PCF uygulamasında ise iki aşamalı bir yaklaşım izlenmiştir [23]. İlk aşamada veri tabanındaki örnekler öznelik eklenmiş, ikinci aşamada genişletilmiş öznelikler ile veri tabanı SVM kullanılarak eğitilmiştir. İlk aşamada öznelik bulunmadan önce [20]'deki yöntem gibi her bir sınıf için k -merkezli kümeleme yöntemiyle k adet küme bulunmaktadır. Her bir örnek için, her sınıfın k adet PCF'inden aldığı en düşük değer, o örneğe öznelik olarak eklenir. Bu işlem sonundan tüm örneklerin öznelik boyutu, sınıf sayısı kadar artmış olur. Sonrasında ise ikinci aşamaya geçilir ve genişletilmiş öznelikler ile veri tabanı SVM yöntemiyle eğitilir. [24]'teki PCF çalışmasında [13]'te kullanılan algoritma [18, 19]'daki hata ceza yaklaşımı ile doğrusal programlama kullanılarak uygulanmıştır. [25]'teki PCF uygulamasında yarı gözetimli öğrenme yöntemiyle çalışma yapılmıştır. Bu çalışmada üyelik fonksiyonları ve PCF, sınıflandırma performansını iyileştirmek için birlikte kullanılmıştır. [26]'daki çalışmada aşırı öğrenmeye karşı rastgele alt uzay metoduyla PCF'i birleştiren bir yöntem önerilmiştir. [27]'de ise hata-ceza yaklaşımı kullanan doğrusal programlama tabanlı P-PCF algoritması önerilmiştir. Yukarıda, PCF fonksiyonu kullanılarak geliştirilen yöntemlerden bahsedilmiştir. Bu yöntemlerin çoğu doğrusal programlama yöntemiyle eğitilen modeller üzerinden hazırlanmıştır. Doğrusal programlamada tüm örnekler eğitim için kullanılmaktadır. Bundan dolayı veri tabanının boyutlarının artmasıyla doğrusal programlamanın hesap yükü artmaktadır. Ayrıca birden fazla PCF kullanımı ile de hesaplama yükü artmakta ve aşırı öğrenmeye maruz kalılabilmektedir. Birden fazla PCF kullanımının eğitim sürecine olumsuz katkısı olduğu gibi test süresini de uzatmaktadır. Ayrıca birden fazla PCF kullanımı eğitim sürecinin karmaşıklaşmasına da neden olabilmektedir. Bu çalışmada basit PCF yapısı kullanılarak tek bir PCF sınıflandırıcı ile hızlı, başarılı ve aşırı öğrenmeye maruz kalmayan MCPCC yöntemi önerilmektedir.

3. YÖNTEM (METHOD)

3.1. Çok Yüzlü Konik Fonksiyon (Polyhedral Conic Function)

Çok yüzlü konik fonksiyon (PCF) ilk olarak [17]'de literatüre girmiştir. Eş. 1'de verilen ve iki ayrık kümeyi

birbirinden ayıran PCF $R^d \rightarrow R$ 'ye tanımlıdır. Bu fonksiyonla $(\mathbf{c}, -b)$ tepe noktasına sahip bir koni oluşmaktadır. Şekil 2'de PCF fonksiyonu ile oluşturulmuş bir koni ve bu koninin seviye kümeleri kullanılarak iki sınıfın ayrılması gösterilmiştir.

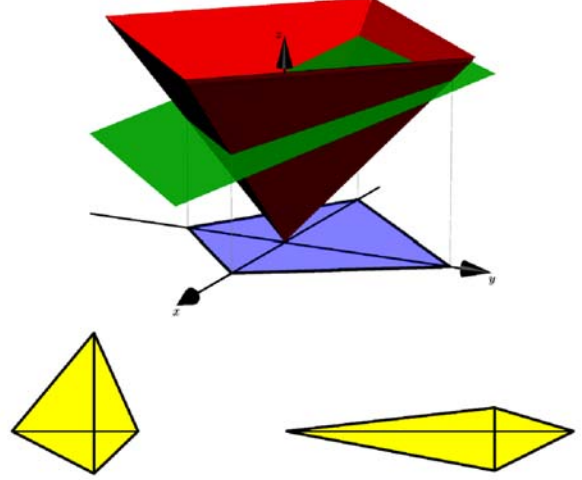
$$f_{\mathbf{w},\gamma,\mathbf{c},b}(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{c}) + \gamma\|\mathbf{x} - \mathbf{c}\|_1 - b \quad (1)$$

Eş. 1'de $\mathbf{x} \in R^d$ test noktası, $\mathbf{c} \in R^d$ koninin tepe noktası, $\mathbf{w} \in R^d$ ağırlık vektörü, $\gamma \in R^+$ ağırlık katsayısı, $b \in R^+$ ofset değer, $\|\mathbf{u}\|_1 = |u_1| + \dots + |u_d|$ L_1 normu ifade etmektedir. İki ayrı sınıfa ait örnekler (\mathbf{x}_i, y_i) ; pozitif örnekler $y_i = +1$, negatif örnekler $y_i = -1$ olarak tanımlanırsa, pozitif ve negatif örnekler $f(\mathbf{x}_i) < 0$, $y_i = +1$ ve $f(\mathbf{x}_i) \geq 0$, $y_i = -1$ şartlarını sağlar. Bu ayırımı, SVM'deki ayırmadan farklılık göstermektedir. SVM'de pozitifler için $\mathbf{w}^T\mathbf{x}_i + b \geq 1$, negatifler için $\mathbf{w}^T\mathbf{x}_i + b \leq -1$ şartları sağlanmaktadır. Halbuki PCF'te pozitif örnekler için negatif, negatifler için pozitif PCF değeri elde edilmektedir. Eş. 1 ile belirtilen en uygun koninin bulunması amacıyla \mathbf{w}, γ, b parametreleri eğitim sürecinde hesaplanmaktadır. Koninin tepe noktası bileşeni, \mathbf{c} , veri yapısına göre model eğitiminden önce belirlenmesi gereken bir parametredir.

3.2. Çok Yüzlü Konik Sınıflandırıcı (Polyhedral Conic Classifier)

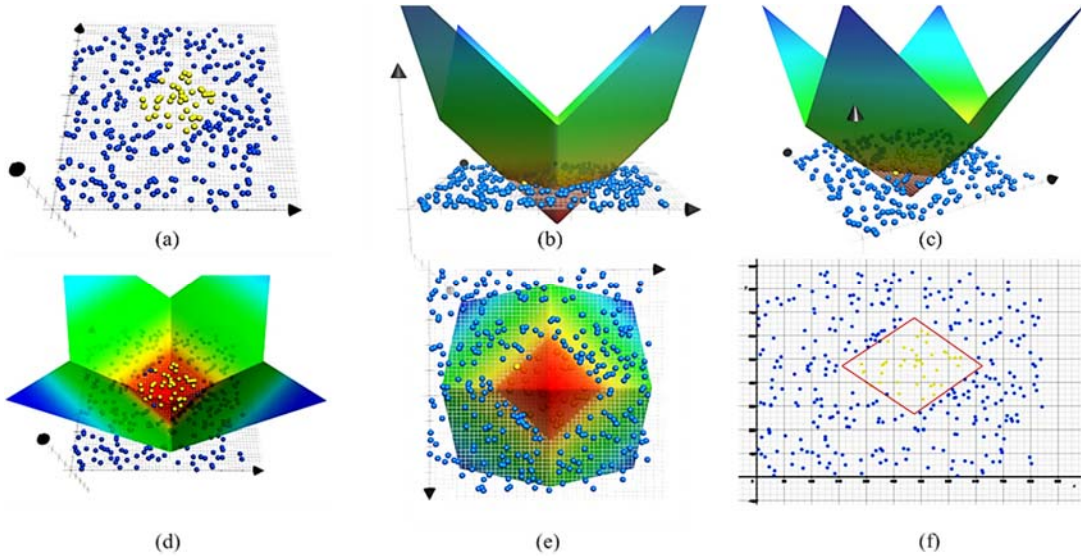
Çok yüzlü konik sınıflandırıcı (PCC), Eş. 1'deki PCF fonksiyonu temel olarak önerilmiştir [17]. Eş. 1 fonksiyonu, Şekil 3'te gösterilen \mathbf{c} merkezli bir koniden ($z = \gamma\|\mathbf{x} - \mathbf{c}\|_1$) geçen hiperdüzlemin ($z = \mathbf{w}^T(\mathbf{x} - \mathbf{c}) - b$) koni içinde kalan kesit alanını ifade etmektedir. Bu kesit alanın öznelik uzayındaki iz düşümü pozitif karar bölgesini oluşturmaktadır. Bu iz düşüm ile oluşan karar sınırı, bir arada kümelennmiş pozitif örnekleri diğer tüm negatif örneklerden

ayıran kompakt ve dışbükey alandır. Ayrıca bu alan marj tabanlı öğrenmeye de uygundur. Makul sayıda serbest değişkenler ile de aşırı öğrenme ve hız kontrolü de kolaylıkla yapılabilmektedir.



Şekil 3. PCF, bir koniden geçen hiperdüzlemin öznelik uzayındaki iz düşümüdür [18]. Sol alt sarı dörtgen PCF ile sağ alt sarı dörtgen EPCF ile elde edilen karar bölgeleridir. (PCF is the projection in the feature space of the hyperplane section through a cone. Bottom left yellow quadrilateral is obtained by PCF, bottom right is obtained by EPCF.)

PCC'de marj tabanlı sınıflandırıcıyı tanımlarken basit bir dönüşüm kullanarak sınıflandırıcımızı standart SVM sınıflandırıcısına benzetebiliriz. Bu dönüşüm, $(\mathbf{x} - \mathbf{c})$ vektörüne $\|\mathbf{x} - \mathbf{c}\|_1$ ekleyerek $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} - \mathbf{c} \\ \|\mathbf{x} - \mathbf{c}\|_1 \end{pmatrix} \in R^{d+1}$, $-\mathbf{w}$ vektörüne $-\gamma$ ekleyerek $\tilde{\mathbf{w}} = \begin{pmatrix} -\mathbf{w} \\ -\gamma \end{pmatrix} \in R^{d+1}$ ve $\tilde{b} = b$



Şekil 2. İki sınıfın ayrılması amacıyla PCF fonksiyonu ile koni oluşturulmaktadır. Bu koninin farklı açılardan görünümü (b)-(e) ile gösterilmiştir. (f)'te, pozitif (sarı) örnekler, negatif (mavi) örneklerden PCF seviye kümesinden elde edilmiş karar sınırı ile ayrılmıştır [19]. (A cone is formed using PCF to classify two classes. Views of the cone from different angles are shown at (b)-(e). Positive samples (yellow) are separated from negatives (blue) by acceptance region obtained from PCF level sets)

ile yapılır. Bu sayede PCC karar fonksiyonu, SVM yöntemindeki gibi pozitifler için $\tilde{\mathbf{w}}^T \tilde{\mathbf{x}} + \tilde{b} \geq 0$, negatifler için $\tilde{\mathbf{w}}^T \tilde{\mathbf{x}} + \tilde{b} < 0$ şartlarını sağlamış olur. Yaptığımız bu dönüşümle PCF'in ∓ 1 marjı da SVM'de kullanılan ± 1 marjına dönüşür. Bu dönüşüm sayesinde Eş. 1'deki PCF fonksiyonu, SVM'deki marj tabanlı karar fonksiyonuna benzetilmiş olur. Bu da bize SVM için kullanılan matematiksel çözümlerin ve yaygın yazılımların PCC sınıflandırıcı için de kullanılabilmesine olanak sağlar. Çok bilinen SVM kuadratik programlamayı PCC için uygularsak optimizasyon problemimiz Eş. 2'deki gibi tanımlanır:

$$\begin{aligned} \arg \min_{\tilde{\mathbf{w}}, \tilde{b}} \quad & \frac{1}{2} \tilde{\mathbf{w}}^T \tilde{\mathbf{w}} + C_+ \sum_i \xi_i + C_- \sum_j \xi_j \\ \text{s. t.} \quad & \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i + \tilde{b} + \xi_i \geq 1, i \in I_+, \\ & \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_j + \tilde{b} - \xi_j \leq -1, j \in I_-, \\ & \xi_i, \xi_j \geq 0, \end{aligned} \quad (2)$$

Eş. 2'de de $I_{+(-)}$ pozitif (negatif) sınıfa ait örneklerin indislerini, $C_{+(-)}$ ceza katsayısını, $\xi_{i(j)}$ ise kısıtları sağlamayan örnekler karşılık gelen arttıran yapay değişkenleri ifade etmektedir. Bu problemi çözmek için SVM eğitiminde kullanılan herhangi büyük veri ile uyumlu bir algoritma kullanılabilir. Eğer pozitif sınıfa ait veriler öznitelik uzayında kompakt bir bölgede toplanmışsa (Bkz. Şekil 3), sınıflandırıcı pozitif örnekleri etrafındaki negatif örneklerden başarı ile ayıran kompakt konveks karar bölgeleri oluşturmaktadır. Eğitim sürecinden sonra bulunan modelde, herhangi bir \mathbf{x}_{test} örneği, eğer $\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_{test} + \tilde{b} \geq 0$ şartını sağlıyorsa pozitif sınıfa, $\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_{test} + \tilde{b} < 0$ şartını sağlıyorsa negatif sınıfa atanmaktadır.

PCC'de kullanılan L_1 normu yerine L_2 norm yaklaşımı da kullanılabilir. Bu sayede bazı veri tabanlarında daha iyi sonuçlar elde edilebilmektedir. Eklenmiş öznitelik vektörü L_2 normunun karesine göre oluşturduğumuzda $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} - \mathbf{c} \\ \|\mathbf{x} - \mathbf{c}\| \end{pmatrix}$, ($\|\mathbf{u}\| = u_1^2 + \dots + u_d^2$), şeklinde olur. Oluşan karar bölgesi 3 boyutlu uzayda elipsoit bir yapıya sahip olur.

Eş. 2'deki kuadratik programlama $\tilde{\mathbf{w}}$ ve \tilde{b} parametrelerinin bulunmasını amaçlar. Koninin tepe noktası olan \mathbf{c} parametresi ise burada hesaplanmamaktadır. \mathbf{c} parametresi yerel olarak optimizasyonla hesaplanabilir [28]. Fakat bu hesaplama sonuca katkı sağlasa da işlem yükünü oldukça arttırmaktadır [28]. Bu yöntem dışında, \mathbf{c} koni merkezi pozitiflerin ortalaması veya orta değeri olarak da seçebilir. Deneylerimizde \mathbf{c} parametresi için pozitiflerin ortalaması tercih edilmiştir.

3.3. Genişletilmiş Çok Yüzlü Konik Sınıflandırıcı (Extended Polyhedral Conic Classifier)

Genişletilmiş çok yüzlü konik sınıflandırıcıda (EPCC), PCC sınıflandırıcısından farklı olarak $\gamma \|\mathbf{x} - \mathbf{c}\|_1$ L_1 norm ifadesi yerine $\gamma^T |\mathbf{x} - \mathbf{c}|$ ifadesi kullanılmaktadır. $\gamma \in R^d$ ağırlık vektörünü ve $\|\mathbf{u}\| = (|u_1|, \dots, |u_d|)^T$ her bir bileşenin mutlak değeri olarak tanımlanmaktadır. Bu farklılıklar dışındaki kalan şartlar Eş. 1 ile aynıdır. Yeni tanımlanan Genişletilmiş

Çok Yüzlü Konik Fonksiyon (EPCF) fonksiyonu Eş. 3'teki gibidir:

$$f_{w,\gamma,c,b}(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{c}) + \gamma^T |\mathbf{x} - \mathbf{c}| - b \quad (3)$$

EPCF fonksiyonu da aynı PCF gibi $z = \mathbf{w}^T(\mathbf{x} - \mathbf{c}) - b$ hiperdüzleminin $z = \gamma^T |\mathbf{x} - \mathbf{c}| = \|\text{diag}(\gamma)(\mathbf{x} - \mathbf{c})\|_1$ konisinin içinde kalan kesitidir. Buradaki koninin PCF konisinden farkı, diyagonal olarak ölçeklenmiş bir koni olmasıdır. PCF ve EPCF ile elde edilen karar bölgeleri Şekil 3'te altta gösterilen sarı dörtgenler ile belirtilmiştir. Soldaki PCF, sağdaki EPCF ile elde edilmiştir. EPCF ile elde edilen karar bölgesi ölçeklenebilir bir dörtgendir. Bu sayede pozitif örneklerin daha iyi ve sıkı çevrelenmesi sağlanabilmektedir.

EPCC eğitimi için PCC sınıflandırıcısında yapılan dönüşüm yaklaşımı aynı şekilde uygulanmaktadır. Bu kapsamda $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} - \mathbf{c} \\ |\mathbf{x} - \mathbf{c}| \end{pmatrix} \in R^{2d}$, $\tilde{\mathbf{w}} = \begin{pmatrix} -\mathbf{w} \\ -\gamma \end{pmatrix} \in R^{2d}$ ve $\tilde{b} = b$ dönüşümü yapılır. Burada dikkat çeken husus PCC'de $(d + 1)$ olan öznitelik boyut büyüklüğü EPCC'de $2d$ 'ye çıkmıştır. Bu dönüşümle de Eş. 2'deki SVM kuadratik programlamasını kullanabiliriz. Bu dönüşümden sonra PCC için geçerli eğitim ve test süreçleri EPCC için de aynıdır.

PCC'de kullanılan L_2 norm karesi yöntemi EPCC içinde uyarlanabilir. Her ne kadar EPCC yönteminde L_2 norm elde edilemese de öznitelik vektörü $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} - \mathbf{c} \\ \mathbf{x}_e \end{pmatrix} \in R^{2d}$, $\mathbf{x}_e = \begin{bmatrix} (\mathbf{x} - \mathbf{c}_1)^2 \\ \vdots \\ (\mathbf{x} - \mathbf{c}_d)^2 \end{bmatrix} \in R^d$ şeklinde oluşturularak, L_2 normunun karesine benzetilebilir. Fakat burada yöntemin adı L_2 norm yerine, norm ifadesi kullanılmadan L_2 olarak adlandırılmaktadır. Yapılan bazı deneylerde L_2 yöntemiyle elde edilen sonuçların L_1 normundan daha iyi olduğu gözlemlenmiştir.

3.4. Çok Merkezli Çok Yüzlü Konik Sınıflandırıcı (Multi Center Polyhedral Conic Classifier)

Çok yüzlü konik sınıflandırıcılarda (PCC, EPCC) bir merkez (\mathbf{c}) içeren PCF (Eş. 1 ve 3) kullanılmaktadır. Bu merkez, koninin tepe noktasını temsil etmektedir. Pozitif örnekler, bu merkez etrafında toplandığında PCC ve EPCC pozitif örnekleri sıkı şekilde çevrelemekte ve sınıflandırma performansı artmaktadır. Bu bilgi ışığında pozitif örneklerin birden farklı bölgelerde kümelendiği durumlarda, daha fazla merkez kullanarak çok yüzlü konik sınıflandırıcıların başarımının artırılabilir. Bu düşünce üzerine Eş. 4'teki çok merkezli çok yüzlü konik fonksiyon (MCPCF) ve bu fonksiyonu kullanan çok merkezli çok yüzlü konik sınıflandırıcı (MCPC) oluşturulmuştur.

$$f_{w,\gamma,c,b,n}(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{c}_1) + \gamma^T \begin{bmatrix} \|\mathbf{x} - \mathbf{c}_1\|_1 \\ \vdots \\ \|\mathbf{x} - \mathbf{c}_n\|_1 \end{bmatrix} - b \quad (4)$$

Eş. 1'deki PCF'ten farklı olarak Eş. 4'te bir merkez yerine n adet $\mathbf{c}_i \in R^d, i = 1, \dots, n$ merkez kullanılmaktadır. \mathbf{x} test

örneği için, bu merkezlerden uzaklıklar L_1 normuna göre hesaplanmakta ve $\boldsymbol{\gamma} \in R^n$ ağırlık katsayısıyla f fonksiyonuna eklenmektedir. Diğer çok yüzlü konik sınıflandırıcılardan farklı olarak eğer merkezler pozitif örneklerin dağılım gösterdiği bölgelerden seçilirse yöntem doğrusal olmayan karar sınırları verebilmektedir. PCF'ten diğer bir fark ise ilk terimde yer almaktadır. PCF'te tek merkez olduğundan, ilk terimde bu tek merkez, \mathbf{x} 'ten çıkarılmaktadır ($\mathbf{w}^T(\mathbf{x} - \mathbf{c})$). MCPCF'te n adet merkez olduğundan, pozitif eğitim örneklerinin ortalaması olan ve ayrı bir merkez olarak da eklenen \mathbf{c}_1 , \mathbf{x} test örneğinden ($\mathbf{w}^T(\mathbf{x} - \mathbf{c}_1)$) çıkarılmıştır. Buradaki amaç diğer çok yüzlü konik sınıflandırıcıların kullandığı \mathbf{c}_1 merkezinin avantajlarını da MCPCF'te kullanmaktır.

MCPCC'in eğitimi de aynı PCC eğitimi gibidir. PCC eğitimindeki aynı prosedürler kullanılıp doğrusal SVM karar fonksiyonuna dönüşüm yapılmaktadır. Bu sayede SVM için kullanılan matematiksel çözümlerin ve yazılımların MCPCC sınıflandırıcımız için de kullanılabilmesi sağlanır. Bu kapsamda $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} - \mathbf{c} \\ \mathbf{x}_e \end{pmatrix} \in R^{d+n}$, $\mathbf{x}_e = \begin{pmatrix} \|\mathbf{x} - \mathbf{c}_1\|_1 \\ \vdots \\ \|\mathbf{x} - \mathbf{c}_n\|_1 \end{pmatrix} \in R^n$, $-\mathbf{w}$ vektörüne $-\boldsymbol{\gamma}^T$ ekleyerek $\tilde{\mathbf{w}} = \begin{pmatrix} -\mathbf{w} \\ -\boldsymbol{\gamma}^T \end{pmatrix} \in R^{d+n}$ ve $\tilde{b} = b$ dönüşümü ile Eş. 2 kullanılarak eğitim süreci gerçekleştirilir. Burada dikkat çeken husus öznelik vektörü, $(d + n)$ boyutlu yeni öznelik vektörüne transfer edilmiştir. Eğer $n = 1$ seçilirse bir adet koni tepe noktası kullanılmış olur. Bu da PCC sınıflandırıcısını vermektedir. Bu açıdan MCPCC sınıflandırıcısı PCC sınıflandırıcısının genelleştirilmiş halidir denilebilir. Fakat burada dikkat edilmesi gereken diğer bir husus da artan n değeri ile hesaplama yükünün de artmasıdır. $\mathbf{x}_i, i = (1, \dots, m)$, m adet eğitim verisi, y_i etiket, n koni merkez sayısı olmak üzere MCPCC eğitim algoritması aşağıdaki gibidir:

Adım-1: Pozitif eğitim örneklerinin ortalamasını, \mathbf{c}_1 , bul.

Adım-2: k -merkezli kümeleme ile $n - 1$ adet küme merkezini, $\mathbf{c}_i, i = 2, \dots, n$, hesapla.

Adım-3: Öznelik eklenmiş örnekleri, $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} - \mathbf{c} \\ \|\mathbf{x} - \mathbf{c}_1\|_1 \\ \vdots \\ \|\mathbf{x} - \mathbf{c}_n\|_1 \end{pmatrix}$,

oluştur.

Adım-4: SVM sınıflandırıcısı kullanarak MCPCF parametrelerini, $\mathbf{w}, \boldsymbol{\gamma}, b$, hesapla.

Merkez sayısı olan n , MCPCC algoritması dışında belirlenmesi gereken bir parametredir. n değeri kademeli olarak artırılarak performans sonuçları gözlemlenmiş ve sonuçların değişiminin azaldığı ve doyum noktasına ulaştığı andaki n değeri seçilmiştir. n parametresi belirlenirken daha küçük ayrı bir veri kümesi kullanılarak daha hızlı sonuçlar elde edilmiştir. Diğer bir MCPCC sınıflandırıcısında ise L_1 norm yerine L_2 (Öklid) norm kullanılmaktadır. Bu sınıflandırıcıda Eş. 5'te tanımlanan L_2 Çok Merkezli Çok Yüzlü Konik Fonksiyonu (L_2 Multi Center Polyhedral Conic Function, MCPCF- L_2) kullanılmaktadır.

Eş. 5'teki $\|\cdot\|$ işlemi, $\|\mathbf{u}\| = \sqrt{u_1^2 + \dots + u_d^2}$, L_2 normunu temsil etmektedir. Bu fonksiyonun kullanıldığı sınıflandırıcı ise L_2 Çok Merkezli Çok Yüzlü Konik Sınıflandırıcı (MCPCC- L_2) olarak adlandırılmıştır. MCPCC- L_2 eğitim sürecindeki dönüşüm ise $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} - \mathbf{c} \\ \mathbf{x}_e \end{pmatrix} \in R^{d+n}$,

$$\mathbf{x}_e = \begin{pmatrix} \|\mathbf{x} - \mathbf{c}_1\| \\ \vdots \\ \|\mathbf{x} - \mathbf{c}_n\| \end{pmatrix} \in R^n, \tilde{\mathbf{w}} = \begin{pmatrix} -\mathbf{w} \\ -\boldsymbol{\gamma}^T \end{pmatrix} \in R^{d+n} \text{ ve } \tilde{b} = b$$

olarak gerçekleştirilir.

$$f_{w,\boldsymbol{\gamma},c,b,n}(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{c}_1) + \boldsymbol{\gamma}^T \begin{pmatrix} \|\mathbf{x} - \mathbf{c}_1\| \\ \vdots \\ \|\mathbf{x} - \mathbf{c}_n\| \end{pmatrix} - b \quad (5)$$

4. DENEYLER (EXPERIMENTS)

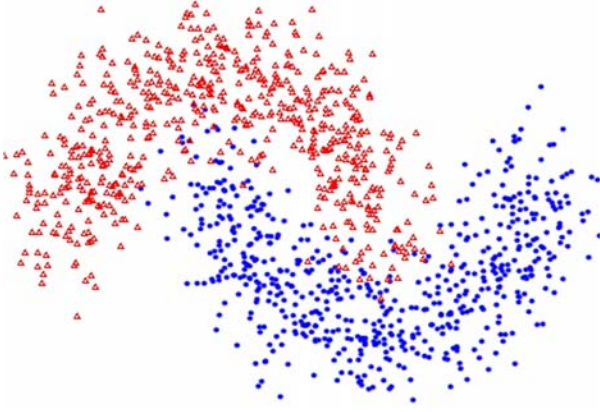
Çalışmada önerilen yöntemler hem sentetik hem de gerçek veri tabanlarında test edilmiştir. Kullanılan veri tabanlarında diğer benzer yöntemlerle de sonuçlar elde edilerek karşılaştırma yapılmıştır. Bu yöntemler aşağıda listelenmiştir:

- 2. derece polinom ve Gauss Kernel SVM (KSVM)
- Additive Kernel [13]
- PCF+Doğrusal SVM [23]

Sonuçlar ikili veri tabanı deneylerinde kullanılan Ortalama Hassasiyet (Average Precision, AP [29]), çok sınıflı veri tabanlarında ise doğru sınıflandırma oranı olarak hesaplanmıştır. Pozitif örneklerin öznelik uzayında farklı bölgelerde kümelendiği durumlarda, MCPCC sınıflandırıcısının kümelenen pozitifleri standart PCC sınıflandırıcısına göre daha iyi sınıflandırması beklendiğinden, bu tarz veri yapısına sahip olduğu değerlendirilen veri tabanları kullanılarak deneyler gerçekleştirilmiştir. Bu bölümde iki adet sentetik veri kümesinde görsel deneyler ile iki sınıflı, ESOGU-285 ve CIFAR-10 veri tabanlarında deneyler yapılmıştır.

4.1. İç İç Geçmiş Veri Kümesi (Nested Dataset)

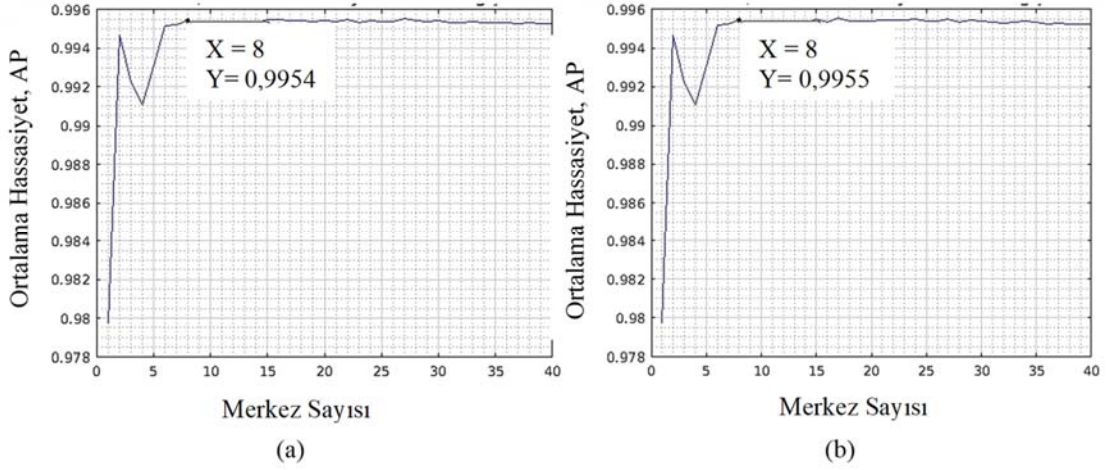
MCPCC sınıflandırıcısı iki adet yapay veri kümesinde incelenmiştir. İlk veri kümesi, iç içe geçmiş iki yarım çember şeklinde bir yapıdır (Şekil 4). Her biri 600 adet pozitif (mavi) ve 600 adet negatif (kırmızı) örnekten oluşan eğitim ve test verisini içermektedir. Öncelikle merkez sayısının sınıflandırma performansı üzerindeki etkisini incelemek ve en uygun merkez sayısını belirlemek amacıyla farklı merkez sayılarıyla elde edilen performans değerleri bulunmuştur. Elde edilen sonuçlarla oluşturulan grafik Şekil 5'te sunulmuştur. Hem MCPCC- L_1 hem MCPCC- L_2 yönteminde 8 adet merkez ile %99,5'lik AP skoru elde edilmiş ve merkez sayısı arttıkça AP değerinde fazla bir artış gözlenmemiştir. Bundan dolayı merkez sayısının 8 olarak belirlenmesi uygun olacaktır. Merkez sayısının değişiminin pozitif karar bölgesinde nasıl bir etki yaptığı Şekil 6'da sunulmuştur. Merkez sayının artmasıyla pozitif karar bölgesinin daha belirleyici olduğu ve daha iyi ayırım yapıldığı görülmektedir.



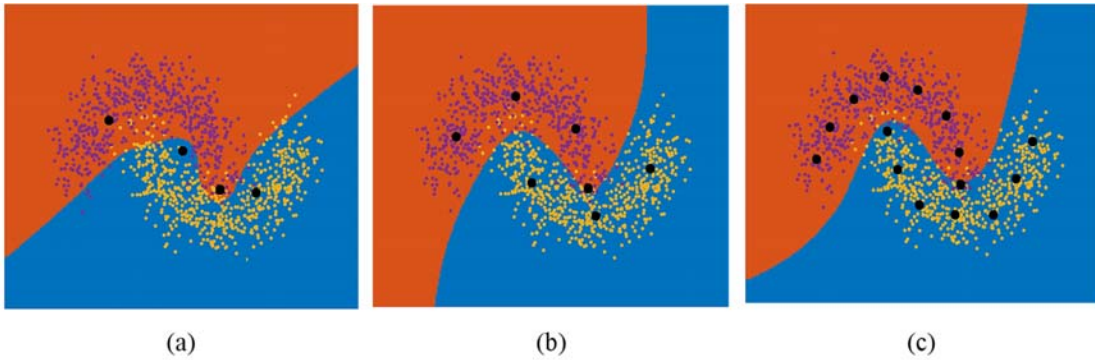
Şekil 4. İç içe geçmiş sentetik veri kümesi (Mavi noktalar: pozitif örnekler, Kırmızı üçgenler: negatif örnekler) (Nested synthetic dataset; blue dots: positive, red triangles: negative samples)

MCPCC yönteminin başarımını karşılaştırmak için mevcut PCC, EPCC yöntemleri ile KSVM (Kernel SVM), Additive Kernel ve PCF+Doğrusal SVM [23] yöntemi ile deneyler

gerçekleştirilmiştir. Her bir yöntem için farklı parametreler ile deneyler tekrarlanmış ve elde edilen en iyi sonuçlar Tablo 1’de sunulmuştur. Sonuçlar incelendiğinde KSVM (Gauss) MCPCC yöntemlerimize çok yakın sonuçlar olsa da MCPCC- L_2 %0,02, MCPCC- L_1 %0,01 daha iyi sonuç almıştır. MCPCC yöntemine benzeyen PCF+Doğrusal SVM ise oldukça iyi skor alarak MCPCC’ye yakın sonuç göstermiştir. Her bir yöntemin kullandığı öznelilik vektörünün boyutu, Tablo 1’de yöntem isminin yanında parantez içinde gösterilmiştir. KSVM yöntemleri için bu değer, kullandığı destek vektör adedi olarak seçilmiştir. KSVM (Gauss) 92 adet destek vektörü kullanırken, MCPCC- L_1 ve MCPCC- L_2 sadece 11 boyut kullanmıştır. MCPCC yöntemlerimizin eğitim süresi ise PCC ve EPCC yöntemlerinden daha uzundur, fakat diğer yöntemlerden çok daha kısadır. Test süresi açısından, PCC- L_1 en hızlı yöntem olurken MCPCC yöntemleri ise PCC ve EPCC ile aynı sonuçları vermektedir. Test esnasında MCPCC- L_1 yönteminin KSVM (Gauss)’den 3,5 kat daha hızlı çalıştığı dikkat çekmektedir. Test süresi açısından en yavaş yöntem ise KSVM (Polinom) olmuştur. Deneydeki sonuçlar göz



Şekil 5. İç içe geçmiş veri kümesinde MCPCC ile farklı merkez sayıları ile AP değişimi, a) MCPCC- L_1 yöntemiyle, b) MCPCC- L_2 yöntemiyle (AP change with different number of centers using MCPCC in nested dataset, a) MCPCC- L_1 method, b) MCPCC- L_2 method)



Şekil 6. Farklı merkez sayıları ile MCPCC- L_2 karar bölgesi değişimi; merkez sayıları a) 4, b) 7, c) 15; siyah noktalar merkezleri, kırmızı bölge negatif alanı, mavi bölge pozitif alanı temsil etmektedir. (The change in acceptance region with different number of centers in MCPCC- L_2 ; number of centers a) 4, b) 7, c) 15; black points are centers, red region is negative area, blue region is positive area.)

önüne alındığında, MCPCC yöntemimiz hem hızı hem de basitliği ile ön plana çıkmaktadır. PCF+Doğrusal SVM yönteminin iki aşamalı algoritmasından dolayı sürelerin karşılaştırmaya uygun olmadığı değerlendirilmiştir.

Şekil 7’deki karar bölgeleri incelendiğinde, MCPCC, KSVM (Gauss), PCF+Doğrusal SVM ve Additive Kernel yöntemlerinin pozitif ve negatif verileri çok başarılı bir şekilde sınıflandırabildiği görülmektedir. Diğer yöntemler bu veri tabanındaki pozitif ve negatifleri ayırmada güçlük çekmektedir.

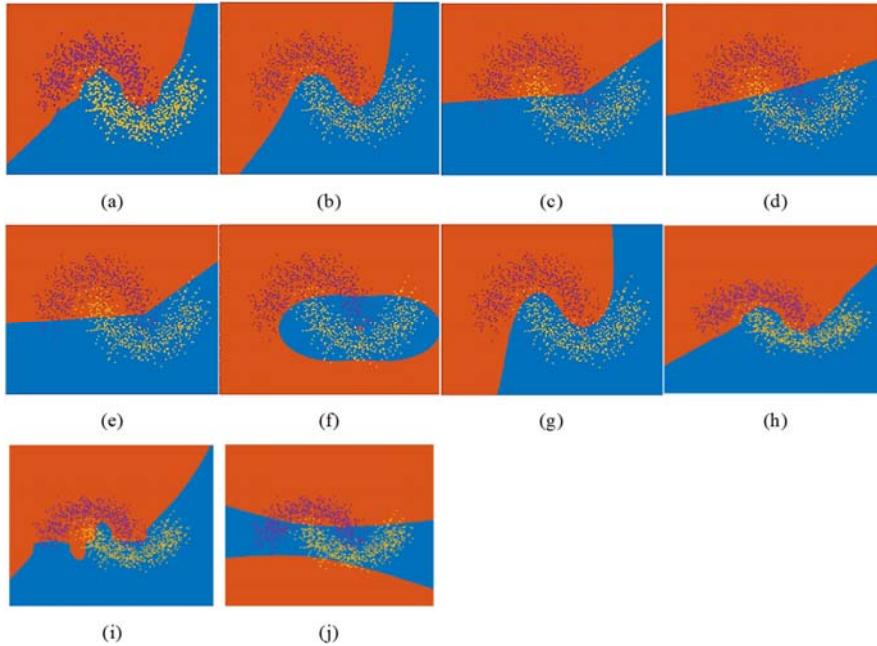
4.2. Döndürülmüş Satranç Tahtası Veri Tabanı (Rotated Chess Board Dataset)

İkinci deneyde $\pi/8$ derece döndürülmüş satranç tahtası şeklinde yerleşime sahip sentetik veriler seçilmiştir (Şekil 8). Eğitim ve test kümeleri rastgele dağılım gösteren 1000 (pozitif + negatif) örnek içermektedir. MCPCC için en uygun

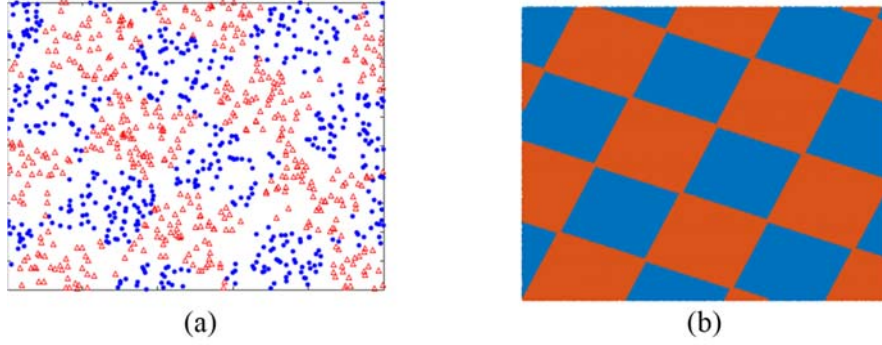
merkez sayısını belirlemek için farklı merkez sayılarıyla elde edilen sonuçlar hesaplanmıştır (Şekil 9). MCPCC- L_2 yönteminde 85 adet merkezle %98,69’luk bir AP değeri elde edilmiştir. MCPCC- L_1 yönteminde ise 12 merkez en iyi sonucu vermiştir. MCPCC- L_2 kullanılarak merkez sayısının artmasıyla pozitif karar bölgesindeki değişim Şekil 10’da gösterilmiştir. Merkez sayısının artmasıyla pozitif verinin daha iyi ayrıştırılabildiği görülmektedir. Deneyde her bir yöntem için farklı parametrelerle elde edilen sonuçların iyileri seçilerek Tablo 2’de sunulmuştur. %99,50 AP değeri ile KSVM (Gauss) en yüksek skoru elde etmiştir. Sonra %99,07 AP değeri ile MCPCC- L_2 yöntemimiz gelmiştir. Diğer yöntemler ise iyi bir başarıyı sağlayamamıştır. Bunun en önemli nedeni pozitif örneklerin dağınık yerleşmesi ve karar bölgelerinin oldukça kompleks yapıya sahip olmasıdır. Kümeleme özneliklerini kullanan PCF+Doğrusal SVM sınıflandırıcısı önerilen yöntemlere oranla kötü sonuçlar vermekle birlikte PCC, EPCC ve KSVM (Polinom) yöntemlerinden daha iyi sonuç vermiştir.

Tablo 1. İç içe geçmiş veri kümesi deneyindeki sonuçlar (Results in nested dataset experiment)

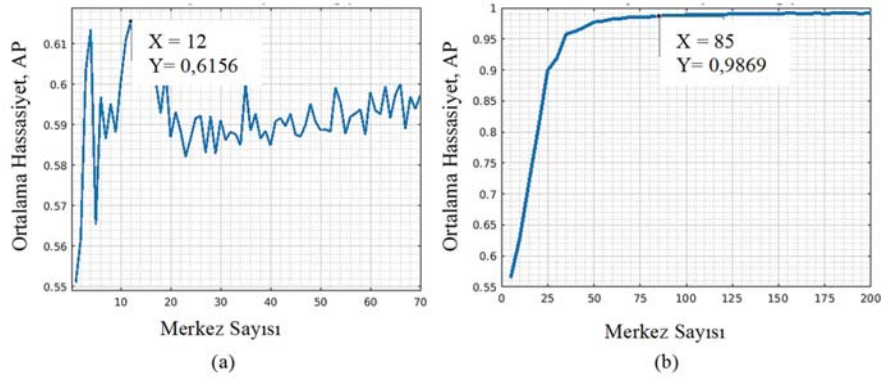
Yöntem	Skor (AP, %)	Eğitim Süresi (ms)	Test Süresi (ms)
MCPCC- L_1 (11)	99,55	15,2	1,03
MCPCC- L_2 (11)	99,56	16,1	1,04
EPCC- L_1 (4)	95,85	7,3	1,11
EPCC- L_2 (4)	95,17	9,1	1,04
PCC- L_1 (3)	96,17	7,5	1,02
PCC- L_2 (3)	95,85	8,0	1,11
KSVM (Gauss) (92)	99,54	113,2	3,80
PCF+Doğrusal SVM (22)	99,42	-	-
Additive Kernel (5)	96,17	238	1,71
KSVM (Polinom) (668)	82,25	22,0	29



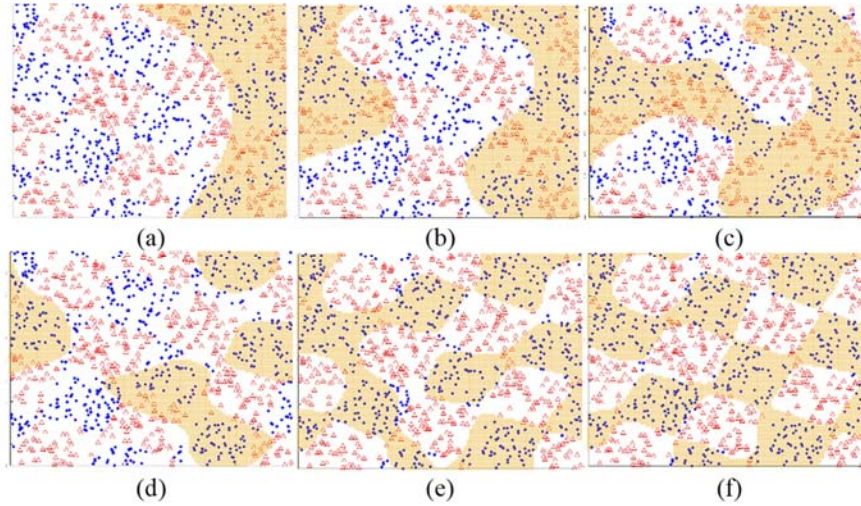
Şekil 7. İç içe geçmiş veri kümesi için pozitif karar bölgeleri: a) MCPCC- L_1 , b) MCPCC- L_2 , c) EPCC- L_1 , d) EPCC- L_2 , e) PCC- L_1 , f) PCC- L_2 , g) KSVM (Gauss), h) PCF+Doğrusal SVM, i) Additive Kernel, j) KSVM (Polinom)
(Positive acceptance regions in nested dataset: a) MCPCC- L_1 , b) MCPCC- L_2 , c) EPCC- L_1 , d) EPCC- L_2 , e) PCC- L_1 , f) PCC- L_2 , g) KSVM (Gauss), h) PCF+Linear SVM, i) Additive Kernel, j) KSVM (Polynomial))



Şekil 8. Döndürülmüş satranç tahtası yerleşimine sahip sentetik veri kümesi: a) gerçek veriler, b) verilerin temsili yerleşimi (Synthetic dataset that has a rotated chess board layout: a) samples b) dataset replacement)



Şekil 9. Farklı merkez sayıları ile ortalama hassasiyet değişimi: a) MCPCC- L_1 , b) MCPCC- L_2 (AP change with different number of centers using MCPCC, a) MCPCC- L_1 , b) MCPCC- L_2)



Şekil 10. Farklı merkez sayıları ile MCPCC- L_2 yönteminde pozitif karar bölgeleri (Mavi noktalar pozitif verileri, kırmızı üçgenler negatif verileri, turuncu bölgeler pozitif karar bölgeleri temsil eder.) Merkez sayısı: a) 1, b) 5, c) 10, d) 20, e) 50, f) 100.

(The change in acceptance region with different number of centers in MCPCC- L_2 (blue points are positive samples, red triangles are negative samples, orange area is positive acceptance region.) Number of centers: a) 1, b) 5, c) 10, d) 20, e) 50, f) 100.)

Her bir yöntemin kullandığı öznelilik vektörünün boyutu, Tablo 2’de yöntem isminin yanında parantez içinde gösterilmiştir. KSVM (Gauss) yöntemi için bu değer, kullandığı destek vektör adedi olarak seçilmiştir. MCPCC-

L_2 yöntemimiz KSVM’e (Gauss) çok yakın bir başarımla sağlamakla birlikte KSVM (Gauss) 190 destek vektörü kullanırken MCPCC- L_2 yöntemi yarısından az (88) boyut kullanmıştır. Burada dikkat çeken diğer bir husus da

Tablo 2. Satranç tahtası deneyindeki sonuçlar (Results in rotated chess board experiment)

Yöntem	Skor (AP, %)	Eğitim Süresi (ms)	Test Süresi (ms)
MCPCC- L_1 (15)	62,61	49	1,341
MCPCC- L_2 (88)	99,07	4384	2,326
EPCC- L_1 (4)	54,93	6,71	0,737
EPCC- L_2 (4)	55,33	6,57	0,803
PCC- L_1 (3)	55,08	8,02	0,717
PCC- L_2 (3)	56,30	5,49	0,822
KSVM (Gauss) (190)	99,50	226	2,233
PCF+Doğrusal SVM (6)	57,30	-	-
Additive Kernel (5)	57,41	24,73	1,8
KSVM (Polinom) (3)	53,92	48,38	29,56

MCPCC- L_2 yöntemi için eğitim süresinin diğer yöntemlerden çok daha fazla olmasıdır. İlk deneyde bu kadar fark mevcut değildi. Bunun nedeni ise merkez sayısının artmasıyla Eş. 5'e göre her noktanın merkezlerden uzaklık hesaplamasının da artmasıdır. İlk deneyde MCPCC'de 8 merkez kullanılırken eğitim süresi açısından diğer yöntemlere çok yakındır. Fakat bu deneyde merkez sayısının 85 seçilmesi hesaplama yükünü ve eğitim/test sürelerini artırmaktadır. Şekil 11'de sınıflandırıcıların nasıl bir pozitif karar bölgesi meydana getirdiği sunulmuştur. MCPCC- L_2 ve KSVM (Gauss) yöntemlerinde oluşan pozitif karar bölgelerinin satranç tahtasına oldukça benzer olduğu görülmektedir. Diğer yöntemler ise istenen karar bölgelerini doğru bir şekilde belirleyememektedir.

4.3. İki Sınıflı Veri Tabanları (Binary Classification Datasets)

Bu deneyde MCPCC'ye benzer yaklaşım kullanan [23]'ün kullandığı aynı prosedür (10 çapraz katman doğrulama) ve veri tabanları (Liver, Splice ve Svmguide1) kullanılmıştır. Bu veri tabanları genellikle küçük ölçekli olup Tablo 3'te özellikleri belirtilmiştir [30]. [23]'ten farklı olarak skorlar hesaplanırken sınıflandırma oranı yerine AP metriği kullanılmıştır. PCF+Doğrusal SVM yazılımı, yöntemin yazarlarından temin edilmiş olup, yöntemin ikinci aşamasında diğer deneylerimizde kullanılan SVM yazılımı kullanılmıştır.

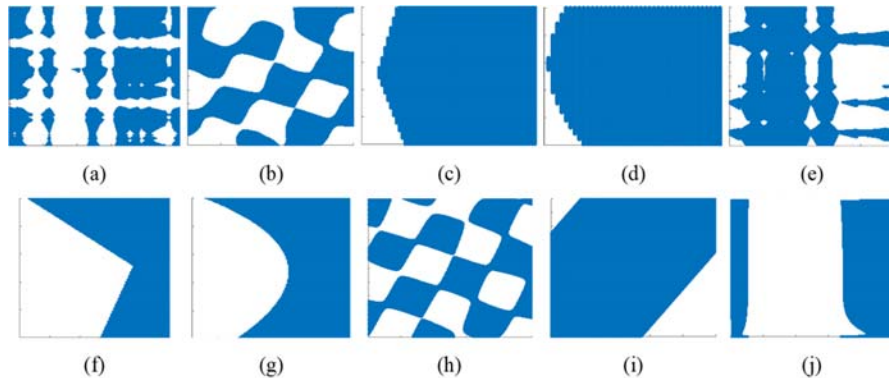
Tablo 3. İkili veri tabanlarının özellikleri (Properties of binary datasets)

Veri tabanı	Liver	Splice	Svmguide
Eğitim örnek sayısı	145	1000	3098
Test örnek sayısı	200	2175	4000
Öznitelik sayısı	5	60	4
Sınıf sayısı	2	2	2

Tablo 4'te sonuçlar sunulmuştur. MCPCC- L_1 yöntemi her üç veri tabanında en iyi veya en iyi ikinci sonucu elde etmiştir. MCPCC- L_2 yöntemi de MCPCC- L_1 yöntemine yakın sonuçlar vermiştir. Tek bir koni merkezi içeren EPCC- L_1 Liver veri tabanında, Additive Kernel yöntemi ise Splice veri tabanında en iyi sonucu almıştır.

Tablo 4. İkili veri tabanları ile deney sonuçları (AP, %) (Results in binary datasets experiment)

Yöntem	Liver	Splice	Svmguide1
MCPCC- L_1	75,34	95,97	99,77
MCPCC- L_2	73,03	92,55	99,73
EPCC- L_1	77,67	92,80	99,67
EPCC- L_2	74,13	95,05	99,51
PCC- L_1	69,85	90,87	99,61
PCC- L_2	68,41	90,53	99,59
PCF+Doğrusal SVM	71,69	91,08	99,38
Additive Kernel	75,12	96,24	99,68
Polinom Kernel SVM	68,88	89,73	99,61



Şekil 11. Döndürülmüş satranç tahtasında pozitif karar bölgeleri, a) MCPCC- L_1 , b) MCPCC- L_2 , c) EPCC- L_1 , d) EPCC- L_2 , e) PCF+Doğrusal SVM, f) PCC- L_1 , g) PCC- L_2 , h) KSVM (Gauss), i) KSVM (Polinom), j) Additive Kernel (Positive acceptance regions in rotated chess board dataset, a) MCPCC- L_1 , b) MCPCC- L_2 , c) EPCC- L_1 , d) EPCC- L_2 , e) PCF+Linear SVM, f) PCC- L_1 , g) PCC- L_2 , h) KSVM (Gauss), i) KSVM (Polynomial), j) Additive Kernel)

4.4. ESOGU-285 Yüz Veri Tabanı (ESOGU-285 Face Dataset)

ESOGU-285 yüz veri tabanı, 285 farklı insana ait yüz videosunu içermektedir [31]. Her bir insanın 4 farklı senaryoda 2 farklı zamanda çekilmiş videosu bulunmaktadır. En kısa video 100, en uzununu 1360 resimden oluşmaktadır. Bu deney için seçilen senaryoda, kişilerden Şekil 12'deki gibi serbest doğal baş hareketleri yapması istenmiştir. Veri tabanından rastgele 10 adet insan seçilerek resimler kırpılmış ve 120×90 boyutuna sabitlenmiştir. Problemi zorlaştırmak için herhangi bir ön işlem veya öznetelik çıkarma uygulamadan resmin salt piksel değerleri kullanılmıştır. Böylece her bir resim için $120 \times 90 = 10\,800$ boyutlu öznetelik vektörü elde edilmiştir. Çok sınıflı bir veri tabanı olduğu için bire-tümü yaklaşımıyla eğitim gerçekleştirilmiştir. Yöntemlerin sınıflandırma oranları Tablo 5'te sunulmuştur.

Tablo 5. ESOGU Yüz veri tabanındaki deney sonuçları (Results in ESOGU face dataset experiment)

Yöntem	Skor (Sınıflandırma Oranı, %)
MCPCC- L_1	68,47
MCPCC- L_2	67,42
EPCC- L_1	72,44
EPCC- L_2	72,44
PCC- L_1	59,84
PCC- L_2	59,87
PCF+Doğrusal SVM	52,68
Additive Kernel	69,05
KSVM (Polinom)	64,84

EPCC- L_1 ve EPCC- L_2 en iyi sonucu göstermiştir. Hemen arkasından Additive Kernel ve MCPCC- L_1 yöntemleri gelmektedir. EPCC orijinal öznetelik boyutunun 2 katı (2×10800), Additive Kernel 3 katı (3×10800) öznetelik boyutuna sahip olmakla birlikte, MCPCC- L_1 orijinal öznetelik boyutu + merkez sayısı ($10800 + 401 = 11\,201$) kadar öznetelik boyutuna sahiptir. Görüleceği üzere MCPCC ile en iyi sonucu veren EPCC sınıflandırıcısı arasında %4 fark

bulunmasına karşın MCPCC'nin öznetelik boyutu EPCC'nin yaklaşık yarısı kadardır.

4.5. CIFAR-10 Veri Tabanı (CIFAR-10 Dataset)

Son olarak CIFAR-10 (<https://www.cs.toronto.edu/~kriz/cifar.html>) veri tabanı kullanılarak deney yapılmıştır. CIFAR-10 veri tabanı 32×32 boyutlu renkli resimleri içeren, her bir sınıfta 6000 resim bulunan, 10 sınıflı bir veri tabanıdır. Toplamda 60 000 resmin 50 000 tanesi eğitim, 10 000 tanesi test amacıyla kullanılmaktadır. Öncelikle AlexNet kullanılarak örneklerin 4096 boyutlu CNN öznetelikleri çıkarılmıştır. Farklı parametrelerle deneyler tekrar edilmiş ve elde edilen en iyi sonuçlar Tablo 6'da sunulmuştur. MCPCC- L_2 yöntemi en iyi sonucu göstermiştir. Bu yöntemi KSVM (2. derece polinom kernel) ve EPCC- L_1 takip etmektedir. Burada dikkati çeken bir husus da MCPCC- L_2 4897 öznetelik boyutunu kullanırken; Kernel SVM 8 386 560, EPCC- L_1 8192 öznetelik boyutu kullanmaktadır. PCF+Doğrusal SVM yönteminde ise doğrusal programlama kullanan ilk aşamada, öznetelik vektörü veri tabanının boyutundan dolayı hesaplanamadığından sonuçlar elde edilememiştir.

Tablo 6. CIFAR-10 veri tabanı deneyindeki sonuçlar (Results in CIFAR-10 dataset experiment)

Yöntem	Skor (Sınıflandırma Oranı, %)
MCPCC- L_1	75,25
MCPCC- L_2	78,81
EPCC- L_1	75,97
EPCC- L_2	75,22
PCC- L_1	75,66
PCC- L_2	75,69
PCF+Doğrusal SVM	-
Additive Kernel	75,45
KSVM (Polinom)	77,16



Şekil 12. ESOGU-285 yüz veri tabanında bulunan bir videodan alınmış resim örnekleri

5. SONUÇLAR (CONCLUSIONS)

Görsel nesne algılama görevlerinde sınıflandırma problemi pozitif ve negatif örnekler açısından her zaman dengeli değildir. Pozitif veriler negatiflerden sayıca daha az olmakla birlikte, pozitifler geometrik olarak kompakt ve görsel olarak tutarlı dağılım gösterirken negatifler daha dağınık ve çok çeşitli bir dağılıma sahiptir. Bu tarz problemlerde, pozitif örnekleri sıkıca çevreleyen, bu esnada iki sınıfın örtüştüğü bölgelerdeki olumsuzlukları da dikkate alarak karar sınırlarını oluşturan bir sınıflandırıcıya ihtiyaç duyulur. Bu amaçla çalışmada kompakt karar sınırları bulan PCC ve EPCC çokyüzlü konik sınıflandırıcıları önerilmiştir. Bu sınıflandırıcılar güçlü ve ölçeklenebilir geniş marj tabanlı yöntemler olup, uygun parametrelerle pozitif karar bölgeleri kompakt sıkı dışbükey bir yapıya sahip olmaktadır. Fakat pozitif örneklerin aynı uzayda farklı bölgelerde kümelenmesi durumunda bu sınıflandırıcılar etkili olamamakta ve sınıflandırmanın yapılabilmesi için birden fazla sınıflandırıcı kullanılması gerekmektedir. Birden fazla sınıflandırıcı ise eğitim/test sürecini yavaşlatabilmekte, daha karmaşık hale getirebilmekte ve aşırı öğrenmeye maruz bırakabilmektedir. Bu tarz problemleri çözmek amacıyla hem PCF basitliğini koruyan hem de tek bir sınıflandırıcı kullanan MCPCC yöntemi geliştirilmiştir. MCPCC yönteminde farklı merkezleri içeren MCPCF fonksiyonu oluşturulmuştur. Bu sayede farklı bölgelerde kümelenen pozitif verilerin tek bir sınıflandırıcı ile etkili bir şekilde sınıflandırılması sağlanmıştır. Yapılan deneylerde de MCPCC sınıflandırıcısının diğer yöntemlerden daha başarılı olduğu görülmüştür.

KAYNAKLAR (REFERENCES)

- Hanbay K., Hyperspectral image classification using convolutional neural network and two dimensional complex Gabor transform, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 35 (1), 443-456, 2020.
- Doğan M., Orman A., Örcü M., Örcü H., A new approach based on regression analysis and mathematical programming to multigroup classification problems, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34 (4), 1939-1956, 2019.
- Aktaş A., Doğan B., Demir Ö., Tactile paving surface detection with deep learning methods, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 35 (3), 1685-1700, 2020.
- Toraman S., Türkoğlu İ., A new method for classifying colon cancer patients and healthy people from FTIR signals using wavelet transform and machine learning techniques, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 35 (2), 33-942, 2020.
- Cevikalp, H., Triggs, B., Large margin classifiers based on convex class models, *IEEE 12th international conference on computer vision workshops*, Kyoto, 101-108, 2009.
- Cevikalp, H., Triggs, B., Hyperdisk based large margin classifier, *Pattern recognition*, 46 (6), 1523-1531, 2013.
- Cortes, C., Vapnik, V., Support vector networks, *Machine learning*, 20, 273-297, 1995.
- Tax D.M.J., Duin, R.P.W., Support vector data description, *Machine Learning*, 54 (1), 45-66, 2004.
- Cevikalp, H., Triggs, B., Visual object detection using cascades of binary and one-class classifiers, *International journal of computer vision*, 123 (3), 334-349, 2017.
- Mangasarian O.L., Wild, E.W., Multisurface proximal support vector machine classification via generalized eigenvalues, *IEEE Transactions on pattern analysis and machine intelligence*, 28 (1), 69-74, 2006.
- Jayadeva, Khemchandani, R., Chandra, S., Twin support vector machines for pattern classification, *IEEE transactions on pattern analysis and machine intelligence*, 29 (5), 905-910, 2007.
- Cevikalp H., Best fitting hyperplanes for classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 (6), 1076-1088, 2016.
- Vedaldi, A., Zisserman, A., Efficient additive kernels via explicit feature maps, *IEEE transactions on pattern analysis and machine intelligence*, 34 (3), 480-492, 2012.
- Rahimi, A., Recht, B., Random features for large-scale kernel machines, *advances in neural information processing systems*, Vancouver, 1177-1184, 2007.
- Dundar, MM., Wolf, M., Lakare, S., Salganicoff, M., Raykar, VC, Polyhedral classifier for target detection: A case study: Colorectal cancer, *25th International Conference on Machine Learning*, Helsinki, 288-295, 2008.
- Bagirov, A.M., Ugon, J., Webb, D., Ozturk, G., Kasimbeyli, R., A novel piecewise linear classifier based on polyhedral conic and max-min separabilities, *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 21 (1), 3-24, 2013.
- Gasimov, R., Ozturk, G., Separation via polyhedral conic functions, *Optimization methods & software*, 21 (4), 527-540, 2006.
- Cevikalp, H., Triggs, B., Polyhedral conic classifiers for visual object detection and classification, *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 4114-4122, 2017.
- Cevikalp H., Sağlamlar H., Polyhedral conic classifiers for computer vision applications and open set recognition, *IEEE transactions on pattern analysis and machine intelligence*, 43 (2), 608-622, 2021.
- Ozturk, G., Ciftci, T., Clustering based polyhedral conic functions algorithm in classification, *Journal of Industrial & Management Optimization*, 11 (3), 921-932, 2015.
- Cimen, E., Ozturk, G., Gerek, O., Incremental conic functions algorithm for large scale classification problems, *Digital Signal Processing*, 77, 187-194, 2018.
- Cimen, E., Ozturk, G., O-PCF algorithm for one-class classification, *Optimization Methods and Software*, 35 (6), 1065-1079, 2019.
- Ozturk, G., Cimen, E., Polyhedral conic kernel-like functions for SVMs, *Turkish Journal of Electrical*

- Engineering & Computer Sciences, 27 (2), 1172-1180, 2019.
24. Ceylan, G., Ozturk, G., Revised polyhedral conic functions algorithm for supervised classification, Turkish Journal of Electrical Engineering & Computer Sciences, 28 (5), 2735-2749, 2020.
 25. Sati Uylas, N., A novel semisupervised classification method via membership and polyhedral conic functions, Turkish Journal of Electrical Engineering & Computer Sciences, 28 (1), 80-92, 2020.
 26. Cimen, E., A Random Subspace Based Conic Functions Ensemble Classifier, Turkish Journal of Electrical Engineering & Computer Sciences, 28 (4), 2165-2182, 2020.
 27. Acar, M., Kasimbeyli, R., A polyhedral conic functions based classification method for noisy data, Journal of Industrial & Management Optimization, 13, doi: 10.3934/jimo.2020129, 2020.
 28. Dordinejad, G. G., Çevikalp, H., Cone vertex estimation in polyhedral conic classifiers, 25th Signal Processing and Communications Applications Conference, Antalya, 1-4, 2017.
 29. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A., The PASCAL visual object classes challenge, International journal of computer vision, 88 (2), 303-338, 2010.
 30. Chang, C., Lin, C., Libsvm: A library for support vector machines, ACM Trans Intell Syst Technol, 2 (3), 1-27, 2011.
 31. Yalcin, M., Cevikalp, H., Yavuz, H.S., Towards large-scale face recognition based on videos, International Conference on Computer Vision Workshops, Santiago, 1078-1085, 2015.