# ÇOK YÜZLÜ KONİK SINIFLANDIRICILAR İÇİN MARJ EN BÜYÜKLENMESİ

**Yüksek Lisans Tezi**

**Gürhan CEYLAN**

**Eskişehir 2017**

# ÇOK YÜZLÜ KONİK SINIFLANDIRICILAR İÇİN MARJ EN BÜYÜKLENMESİ

**Gürhan CEYLAN**

**YÜKSEK LİSANS TEZİ**

**Endüstri Mühendisliği Anabilim Dalı**
**Danışman: Doç. Dr. Gürkan ÖZTÜRK**

**Eskişehir**
**Anadolu Üniversitesi**
**Fen Bilimleri Enstitüsü**
**Mayıs, 2017**

*Bu Tez Çalışması BAP Komisyonunca kabul edilen 1607F673 nolu proje kapsamında desteklenmiştir.*

## JÜRİ VE ENSTİTÜ ONAYI

Gürhan CEYLAN'ın "Çok Yüzlü Konik Sınıflandırıcılar için Marj En Büyüklenmesi" başlıklı tezi 29/05/2016 tarihinde, aşağıdaki jüri tarafından "Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği" nin ilgili maddeleri uyarınca Endüstri Mühendisliği Anabilim dalında Yüksek Lisans tezi olarak kabul edilmiştir.

| | | Unvanı-Adı-Soyadı | İmza |
|---|---|---|---|
| Üye (Tez Danışmanı) | : | Doç. Dr. Gürkan ÖZTÜRK | …….. |
| Üye | : | Pof. Dr. Nihal ERGİNEL | …….. |
| Üye | : | Yrd. Doç. Dr. Hasan Serhan YAVUZ | …….. |

.....

Enstitü Müdürü

# ÖZET

ÇOK YÜZLÜ KONİK SINIFLANDIRICILAR İÇİN MARJ EN BÜYÜKLENMESİ

Gürhan CEYLAN
Endüstri Mühendisliği Anabilim Dalı
Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Mayıs, 2017
Danışman: Doç. Dr. Gürkan ÖZTÜRK

Genelleştirme yeteneği, sınıflandırma algoritmalarının başarılı tahminleme yapabilmelerinde önemli bir yere sahiptir. Literatürde iyi bilinen destek vektör makineleri, farklı veri kümelerinin birbirlerine en yakın noktalarından geçen hiperdüzlemler arasındaki mesafe olarak tariflenen marj değerini en büyükleyerek genelleştirme yeteneğini artırmaya çalışmaktadır. Yapılan araştırma kapsamında çok yüzlü konik fonksiyonlar, destek vektör makinelerinden elde edilen en büyük marj değerine sahip ayırıcı hiperdüzlem olarak tariflenerek, marj en büyüklenmesi yaklaşımı konik yüzeylere uygulanmıştır. Bu uygulama ile çok yüzlü konik sınıflandırıcılar için marj değerini en büyüklemek üzere desktek vektör makineleri temelli iki yeni yaklaşım önerilmiştir. Birinci yaklaşımda, çok yüzlü konik fonksiyonlar, temel formda bir kernel fonksiyonu gibi kullanılarak hem hiperdüzlem hem de konik sınıflandırıcı yüzeyler oluşturulmuştur. İkinci yaklaşımda ise genelleştirilmiş özdeğer problemi çözülerek, konik fonksiyonlar ile uzaklık değerine dayanan konik sınıflandırıcılar elde edilmiştir. Bunlara ek olarak, çok yüzlü konik fonksiyonlar algoritmasının aşırı uyum sorununu gidermek amacıyla ceza parametreli bir yaklaşım da önerilmiştir.

**Anahtar Kelimeler:** Marj En Büyüklenmesi, Çok Yüzlü Konik Sınıflandırıcılar, Destek Vektör Makineleri, Optimizasyon

## ABSTRACT

MARGIN MAXIMIZATION FOR POLYHEDRAL CONIC CLASSIFIERS

Gürhan CEYLAN

Department of Industrial Engineering

Anadolu University, Graduate School of Science, May, 2017

Supervisor: Assoc. Prof. Dr. Gürkan ÖZTÜRK

Generalization ability has a key role for successful prediction for classification algorithms. In literature, well known support vector machines tries to increase generalization ability via maximizing the value called margin, which is the largest distance between two parallel hyperplanes on the closest points of different data sets. In this research, polyhedral conic functions are reformulated as maximum margin separating hyperplane and, idea of margin maximization is adapted to conic surfaces. Based on this idea, two new approaches are proposed to maximize margin value for conic classifiers. In the first approach, conic functions are used in a same manner with kernel functions to obtain both hyperplane and conic classifiers. In the second approach, a distance based conic classifier is obtained by solving generalized eigen value problem. In addition to these, a penalized approach is also proposed to overcome overfitting problem of the polyhedral conic functions algorithm.



**Keywords:** Margin Maximization, Polyhedral Conic Classifiers, Support Vector Machines, Optimization.

## TEŞEKKÜR

Tarih boyunca özgürlükleri, hayatları pahasına dahi olsa kararlı duruşlarını koruyarak bilimin itaatkar olmadığını gösteren bütün güzel insanlara teşekkürü borç bilirim.

Gürhan CEYLAN

MAYIS 2017

27/05/2017

## ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilemeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Anadolu Üniversitesi tarafından kullanılan "bilimsel intihal tespit programı"yla tarandığını ve hiçbir şekilde "intihal içermediğini" beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

.....

Gürhan CEYLAN

# İÇİNDEKİLER

**Sayfa**

**Sayfa**

## TABLOLAR DİZİNİ

**Sayfa**

## ŞEKİLLER DİZİNİ

**Sayfa**

## SİMGELER VE KISALTMALAR DİZİNİ

| | |
|---|---|
| DVM | : Destek vektör makineleri |
| ÇKF | : Çok yüzlü konik fonksiyonlar |
| $k$-ort ÇKF | : Kümeleme temelli çok yüzlü konik fonksiyonlar |
| GÖPDVM | : Genelleştirilmiş özdeğer problemi destek vektör makineleri |
| ÇKFDVM | : Destek vektör makineleri yaklaşımı |
| GÖPÇKF | : En iyi koni yaklaşımı |
| $c$-ÇKF | : Ceza parametreli koni yaklaşımı. |
| Enk | : En küçüklemek |

# 1. GİRİŞ

İnsan türü, zekasını kullanmaya başladığından bu yana çevresindeki olan biteni gözlemlemeye ve gözlemlerinden belirli bir örüntü/düzen/anlam bulmaya çalışa gelmiştir. Başlangıç olarak ay, güneş ve yıldızların konumları gözlemlenerek çeşitli düzenler çıkarılmış, mevsimsel döngülerden ekip dikme işlerinin düzenlenmesi sağlanmış, takvimler oluşturulmuştur. Zaman içerisinde gelişen gözlem yapma ve örüntü bulma yöntemleri sayesinde (bilimsel yöntem), makro boyuttaki yasalardan atom altı parçacıkların işleyişine kadar çok karmaşık sistemlere ait düzenler bilinebilir hale gelmiştir. Artan veri üretimi ve işleme gücüne paralel olarak gelişen, veri yığınlarından yararlı bilgilerin çıkarılmasını amaçlayan veri madenciliği çalışmaları, ilkel insandan beri süregelen insan aklı ile gerçekleştirilen örüntü arama işlemlerinin, bilgisayarlar tarafından yapılmasını amaçlayan çalışmalar olarak ifade edilebilir.

Büyük veri yığınlarından, klasik istatistiksel yöntemlerle çıkarılamayan yararlı bilgilerin elde edilmesi amacıyla geliştirilen veri madenciliği yöntemleri, sınıflandırma/tahminleme ve kümeleme/tanımlama olarak iki ana başlıkta toplanabilmektedir. Sınıflandırma çalışmalarında elde edilmek istenen yararlı bilgi önceden belirlenmiş olduğundan ve geliştirilen yöntemler bu amaç doğrultusunda oluşturulduğundan güdümlü/denetimli öğrenme olarak, kümeleme çalışmalarında ise elde edilmek istenen bilgi önceden bilinmediğinden ve geliştirilen yöntemler özel bir amaç doğrultusunda geliştirilmediğinden dolayı güdümsüz/denetimsiz öğrenme olarak isimlendirilmektedir.

Sınıflandırma, sınıf bilgileri bilinen veriler ile oluşturulan modeller yardımıyla, hangi sınıfa ait olduğu bilinmeyen verilerin sınıf değerlerinin tahminlenmesi olarak ifade edilebilir. Sınıflandırma yöntemleri sağlık, genetik, astronomi, eğitim vb. gibi geniş bir araştırma alanında tariflenen problemleri çözmeyi amaçlamaktadır. Örneğin, belirli özelliklere dayanarak, hasta ve sağlıklı kişilerin belirlenmesi, bir gen diziliminin sorumlu olduğu proteinin tahminlenmesi ya da bir öğrencinin aldığı bir dersi başarılı bir şekilde geçip geçemeyeceğinin tahminlenmesi bu tür problemlere örnek olarak verilebilir. Sınıflandırma yöntemleri, farklı alanlardaki problemlerin etkin bir şekilde çözülebilmesini amaçladığından, tahminleme ve genelleştirme başarılarının yüksek olması istenmektedir. Tahminleme başarısı, tahminlenmek istenen değerin doğru belirlenmesinin bir ölçüsü iken, genelleştirme başarısı ise sınıflandırma yönteminin veri ile olan uyumunun bir ölçüsüdür.

Tez kapsamında, Vapnik ve Cortes (1995) [1] tarafından ortaya atılan Destek Vektör Makineleri (DVM) yönteminde kullanılan marj en büyüklemesi yaklaşımını, Gasimov ve Öztürk (2006) [2,3]. tarafından ortaya konulan Çok Yüzlü Konik Fonksiyonlar (ÇKF) ile birleştirmeye yönelik çalışmalar yapılmıştır. Yapılan çalışmalarda, ÇKF formulasyonu, DVM yönteminde bulunması amaçlanan en büyük marj değerine sahip ayırıcı yüzey formulasyonuna uygun bir şekilde tarif edilmiştir. Yapılan tarifleme ile, DVM yaklaşımı ve en iyi koni yaklaşımı geliştirilerek, marj en büyüklenmesi konik yüzeylere uygulanmıştır. Bu yaklaşımlara ek olarak, ÇKF algoritmasının ceza parametreli bir versiyonu da ortaya konulmuştur.

İkinci bölümde sınıflandırma problemi ve yöntemleriyle ilgili genel bilgiler verilmiş, literatürde çokça bahsedilen yöntemler kısaca anlatılmıştır. Ayrıca yöntem değerlendirme ölçütleri ve model seçim yöntemleri ifade edilmiştir.

Bölüm 3'te, ÇKF, ÇKF algoritması ve ayırıca Öztürk ve Çiftci (2011) [4] tarafından ortaya konulan kümeleme temelli ÇKF (*k*-ort ÇKF) algoritması, ayrıntılı bir şekilde anlatılmıştır. Bölüm 4'te ise, DVM yöntemi ele alınmış ve bu yöntemin bir varyasyonu olan Mangasarian ve Wild (2006) [5] tarafından ortaya konulan Genelleştirilmiş Özdeğer Problemi Destek Vektör Makineleri (GÖPDVM) yöntemi incelenmiştir.

Bölüm 5'te, aşırı uyum sorunu ele alınmış, ÇKF algoritmasının bir analizi yapılarak algoritma özelinde sorun sebepleri ortaya konulmuştur. Geliştirilen yöntemler, destek vektör makineleri yaklaşımı, en iyi koni yaklaşımı ve ceza parametleri yaklaşım sırasıyla sunulmuştur. Son olarak Bölüm 6'da ise, geliştirilen yöntemlerin implementasyonu ve parametre araması ile ilgili bilgiler verilerek, yöntemler elde edilen sonuçlar üzerinden karşılaştırılmıştır. Çalışma, genel bir değerlendirme ve ileride yapılabilecek araştırmalar ifade edilerek sonlandırılmıştır.

## 2. SINIFLANDIRMA

Bu bölümde sınıflandırma yöntemlerinin genel olarak nasıl çalıştığı anlatılmış, sınıflandırma yöntemlerinin karşılaştırılmaları için kullanılan, ölçüm değerlerinden bahsedilmiştir. Ayrıca model seçimi için literatürde kullanılan yöntemlere kısaca değinilmiştir.

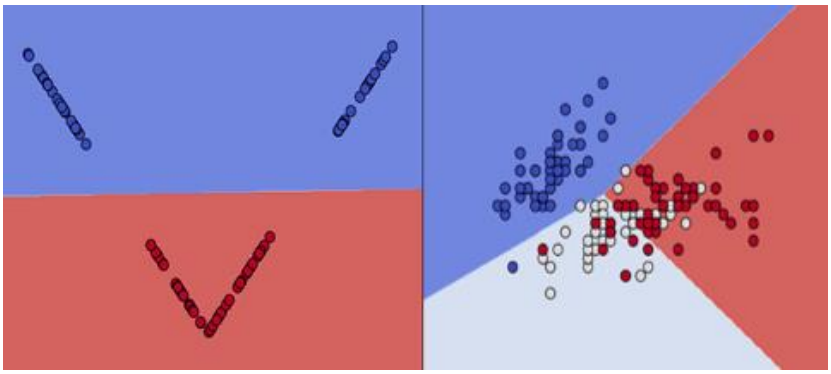
Bilimsel çalışmaların pek çoğu, yapılan gözlemlere uygun olarak tariflenen modeller aracılığıyla, mevcut durumun ifade edilmesi ya da gelecekte olacak olan olayların tahminlenmesini amaçlamaktadır. Veri işleme ve depolama gücüne paralel olarak gelişen, büyük boyutlu verilerin analiz edilerek yararlı bilgilere dönüştürülmesi işlemi olarak tariflenen veri madenciliğinde kullanılan, sınıflandırma yöntemleri de bu tip çalışmaların içerisindedir.

Sınıflandırma, bir verinin ait olduğu sınıfın ya da kategorik bir değerin, elde bulunan veriler yardımıyla oluşturulan modeller kullanılarak tahminlenmesidir. Örneğin: yapılan kan tahlillerine göre bir kişinin hasta olup olmadığı, bir çiçeğin yaprak boyutu, rengi gibi özelliklerine dayanarak hangi türe ait olduğunun belirlenmesi, sınıflandırma problemi olarak ifade edilebilir. Sınıflandırma problemleri farklı disiplinlerde, çeşitli çözüm gereksinimleri olan problemleri kapsadığından bu alanda geniş bir literatür bulunmaktadır. Karar ağaçları, bayes sınıflandırma, en yakın komşu yöntemi, literatürde çokça bahsedilen sınıflandırma yöntemlerindendir. Bunların yanı sıra DVM yöntemi ve matematiksel modelleme temelli yöntemler de bulunmaktadır. Bu yöntemler ilerleyen bölümlerde ayrıntılı olarak ele alınacaktır.

Tahminleme modellerinin oluşturulması için, önceden hangi sınıfa ait oldukları bilinen bir veri kümesine ihtiyaç duyulur. Literatürde eğitim kümesi olarak isimlendirilen bu kümeden seçilen verilerle oluşturulan modeller, mevcut ve daha önce karşılaşılmamış olan test kümesindeki verilerin hangi sınıfa ait olduklarını ifade edebilmektedirler. Şekil 2.1'de ikili ve çoklu sınıflandırıcı yüzey örnekleri görülmektedir. Görülen ayırıcı yüzeyler/karar kuralları/fonksiyonları farklı sınıflara ait olan verileri ayırabilmekte, daha önce karşılaşılmamış bir veriyi ise yüzeyin hangi tarafında kaldığına göre sınıflayabilmektedirler.

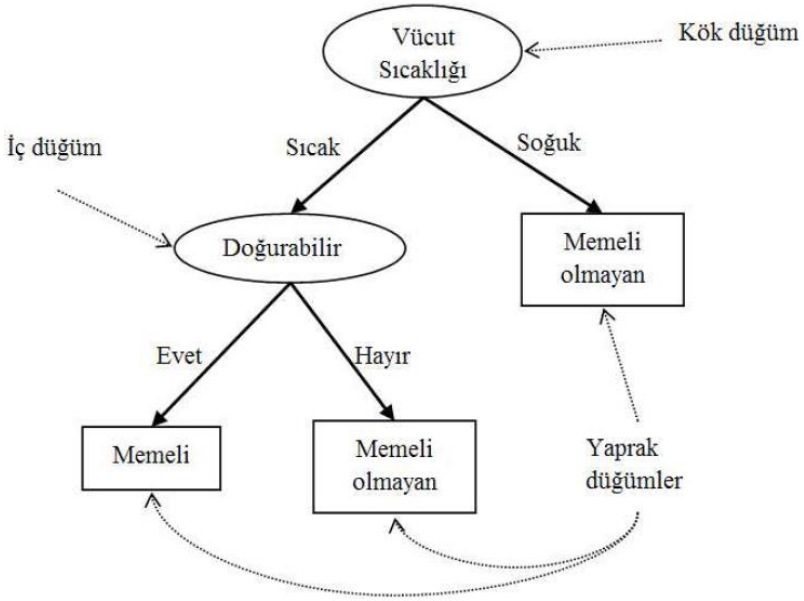
Eğitim kümesindeki verilerin sınıf değerlerinin/etiketlerinin bilindiği çalışmalar denetimli öğrenme olarak isimlendirilirken, verilerin etiketlerinin bir kısmının bilindiği ya da hiç bilinmediği senaryolar da oluşabilmektedir. Etiket bilgilerinin bir kısmının

bilinerek yapılan çalışmalar, yarı denetimli öğrenme olarak adlandırılırken, etiket bilgilerinin bilinmediği çalışmalar denetimsiz öğrenme olarak adlandırılmaktadır.



**Şekil 2.1** *İkili ve çoklu sınıflandırıcı yüzeyler.*

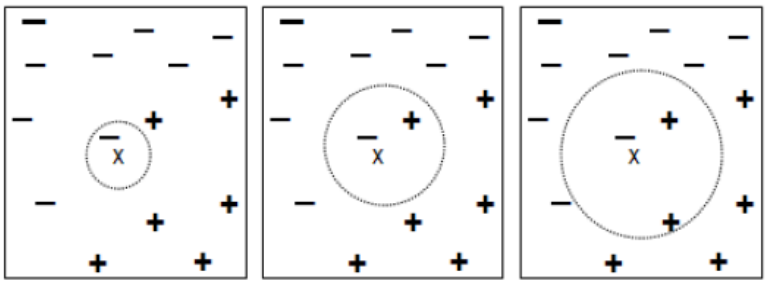
Adını oluşturduğu ağaç yapısından alan karar ağaçları, kök düğümle başlar, iç düğümlerle devam eder ve yaprak düğüm ile sonlanır. Genellikle her düğümde bir özelliğe ait soru taşıyan bu yapı bir veriyi genelden özele giderek sınıflandırır. İlk düğümde veriyi en iyi ayırdığı düşünülen özelliğe ait sorudan elde edilen cevapla bir sonraki soru sorulur, bu süreç yaprak düğüme kadar devam eder. Bu yöntemin avantajları gürültüye karşı duyarlı olmaması, gereksiz özellikleri kullanmaması olarak ifade edilebilir [3]. Karar ağaçlarının yapısını gösteren görsel Şekil 2.2'de verilmiştir.




**Şekil 2.2** *Bir karar ağacı yapısı*

**Kaynak:** *Öztürk, 2007, s.25*

Tembel bir sınıflandırma yöntemi olan en yakın komşu yönteminde, test aşamasına kadar model oluşturulmaz. Test aşamasında belirli özelliklere göre komşuluk değeri hesaplanarak, veriler en yakın komşunun sınfına atanırlar. Her test sırasında, model baştan oluşturulduğundan ve eğitim kümesindeki bütün veriler kullanıldığından, depolama ve hesaplama maliyeti yüksek olabilmektedir. Bu yöntemin düzgün çalışabilmesi için, komşuluk değerinin hesaplandığı özelliklerin normalleştirilmesi gerekmektedir [3]. Şekil 2.3'te farklı komuşuluk değerlerine, göre oluşan sınıflandırma yüzeylerini göstermektedir.




**Şekil 2.3** *Komşuluk değerine göre değişen sınıf değerleri.*

**Kaynak:** *Öztürk, 2007, s. 21*

Thomas Bayes'in ortaya koyduğu Bayes teoremine dayanan, Bayes sınıflandırma yöntemlerinde, $X$ veri kümesi bilindiğinde $Y$ sınıf değişkeninin olasılığı $P(Y|X)$ hesaplanarak, sınıflandırma yapılmaktadır. Bu yöntemin düzgün çalışabilmesi için eğitim kümesindeki örnek sayısının fazla olması gerekmektedir. Gerekli sayıda örneğin olmadığı durumlarda, Monte Carlo yöntemleri ile ilgili verinin olasılık dağılımı ya da kısmi olasılık dağılımı kullanılarak, gerçek veriyi taklit eden yapay örnekler oluşturulur. Naif Bayes yönteminde, $X$ veri kümesinde bulunan özelliklerin birbirinden bağımsız oldukları varsayılarak, koşullu olasılık hesaplamadan kaynaklanan zorluklar giderilmiştir [3].

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{2.1}$$

Sınıflandırma yöntemlerinin zengin bir literatüre ve çeşitliliğe sahip olması, bu yöntemlerin karşılaştırılmasını ve başarılı olanların belirlenmesini gerektirmektedir. Bu amaçla kullanılan ölçüt değerleri, tahminleme oranı, kesinlik, duyarlılık ve $f$ değerleridir.

## 2.1 Değerlendirme Ölçütleri

Bir sınıflandırma işlemi sonucu Tablo 2.1'de görülen hatalı sınıflandırma matrisi oluşturulur. Bu matris yapılan tahminler ile gerçek sınıf değerlerini karşılaştırarak yukarıda bahsedilen ölçüt değerlerinin hesaplanmasında kullanılır.

**Tablo 2.1** *Hatalı sınıflandırma matrisi*

| | | **Gerçek Sınıf** | |
|---|---|---|---|
| | | **A** | **B** |
| **Tahmin Edilen Sınıf** | **A** | gerçek pozitif (gp) | yanlış pozitif (yp) |
| | **B** | yanlış negatif (yn) | gerçek negatif (gn) |

***Tahminleme başarısı:*** Gerçek pozitif tahmin sayısının tüm tahminlere oranı ile bulunan bu değer çoğu durumda kullanışlı olmakla beraber, dengesiz verilerden oluşan problemlerde yöntem başarısını yanlış ifade edebilmektedir.

$$Tahminleme\ Başarısı = \frac{gp}{yp + yn + gn}$$

***Kesinlik:*** Dengesiz problemlerde kullanılan bu değer, gerçek pozitif tahmin sayısının, gerçek pozitif ve yanlış negatif sayılarına oranları ile hesaplanır.

$$Kesinlik = \frac{gp}{gp + yp}$$

***Duyarlılık:*** Dengesiz problemlerde kullanılan bir diğer ölçüt olan duyarlılık ise, gerçek pozitif tahmin sayısının, gerçek pozitif ve yanlış pozitif sayılarına oranları ile hesaplanır.

$$Duyarlılık = \frac{gp}{gp + yn}$$

***f değeri*:** Çoğu durumda kesinlik ve duyarlılık ölçütleri birbiri ile çelişen amaçlar oluşturduğundan, sınıflandırma yöntemlerinin nihai karşılaştırmaları için genel bir değerlendirme ölçütü olarak $f$ değeri kullanılmaktadır. Bu ölçük kesinlik ile duyarlılığın harmonik bir ortalamasıdır.

$$f = \frac{2 \times duyarlılık \times kesinlik}{duyarlılık + kesinlik}$$

## 2.2 Model Seçme Yöntemleri

Bir sınıflandırma yönteminin, önceki bölümde tariflenen ölçüt değerlerinde iyi sonuçlar verebilmesi için, tahminleme modelinin ve eğer var ise yönteme ait parametrelerin başarılı bir şekilde seçilmesi gerekmektedir. Bu gereksinimi karşılamak için en çok kullanılan yöntemler, bir eğitim bir test kümesi(Holdout), bir biri dışarıda kalsın(Leave one out) yöntemi, k-kere çapraz doğrulama(*k*-Fold Cross Validation) ve katmanlı *k*-kere çapraz doğrulama (Stratified *k*-Fold) yöntemi olarak ifade edilebilir.

***Bir eğitim bir test kümesi:*** Bu yöntemde, mevcut veriler iki bölüme ayrılarak, bir kısmı eğitim kümesi olarak kullanılırken diğer kısmı, test kümesi olarak kullanılmaktadır. Verinin hangi oranda ayrılacağı analizi yapan kişinin inisiyatifine kalmış olmakla birlikte, eğitim kümesi daha büyük olacak şekilde seçilir. Bu yöntemin temel kısıtları, verinin sadece belirli bir kısmının test için kullanılması dolayısıyla eğitim için kullanılan veri sayısını azaltması ve seçilen küme boyutlarına bağlı sonuçlar verebilmesidir [3].

***Biri dışarıda kalsın:*** Bu yöntemde *n* elemanlı bir veri seti için *n* adet model oluşturulur. Oluşturulan her modelin eğitimi için *n-1* eleman kullanıldığından hesaplama açısından maliyetli bir yöntemdir. Ayrıca bu yöntem ile tahminleme oranı varyans değeri yüksek tahminleyiciler ortaya çıkar [3].

***k-kere çapraz doğrulama:*** Bu yöntemde ise *n* elemanlı bir veri setinden, olabildiğince eleman sayıları birbirine yakın olabilecek şekilde *k* adet alt küme ve *k* adet model oluşturulur. Oluşturulan her modelin eğitiminde *k-1* adet altküme kullanılırken, geriye kalan bir küme ise ölçüt değerlerinin hesaplanması amacıyla test kümesi olarak kullanılır. Bu yöntemde başarı ölçütleri, *k* adet test sonucundan elde edilen değerlerin ortalaması alınarak hesaplanır [3].

***Katmanlı k-kere çapraz doğrulama:*** *k*-kere çapraz doğrulama yönteminin bir varyasyonu olan bu yöntemde, her altküme, asıl eğitim kümesindeki farklı sınıflara ait oran değerleri korunarak oluşturulur. Bu sayede, yapılan testlerde verinin gerçek yapısına olan uygunluğun korunması amaçlanmaktadır. Yöntemin daha gerçekçi alt örnekler oluşturması dolayısıyla, çalışmada model seçim yöntemi olarak tercih edilmiştir.

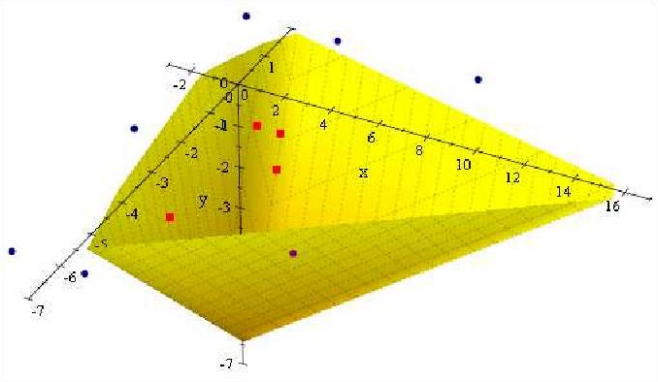
# 3. ÇOK YÜZLÜ KONİK FONKSİYONLAR

İki ya da daha fazla kümenin ayrılması sorunu bir en iyileme problemi olarak tariflenebildiğinden, literatürde matematiksel programlama temelli sınıflandırma yöntemleri de bulunmaktadır. Bu yöntemlerin başlıcaları, Bennet ve Mangasarian'ın (1992) [7] gürbüz (robust) doğrusal programlama algoritması, Astorino ve Gaudiosso'nun (2002) [8] h-çokyüzlü yöntemi ve Bagirov'un (2005) [9] en büyük-en küçük ayırma yöntemi olarak ifade edilebilir.

Çok yüzlü konik yüzeyler yardımıyla bir veri kümesini en iyi şekilde ayırmayı amaçlayan çok yüzlü konik fonksiyonlar algoritması da matematiksel programlama temelli bir sınıflandırma yöntemidir. İlgili çalışmada öne sürülen çok yüzlü ayırma fonksiyonu $g_{(w,\xi,\gamma,\alpha)} : R^n \rightarrow R$ aşağıdaki gibi tanımlanmaktadır. Aynı çalışmada, ortaya konulan ayırıcı fonksiyonun grafiğinin bir koni olduğu ve her alt seviye kümesinin de bir dışbükey çok yüzlü bir küme olduğu ispatlanmıştır. İspat detayları için ilgili makaleye bakılabilir [2]. Şekil 3.1. bir ÇKF grafiğini göstermektedir.

$$g_{(w,\xi,\gamma,\alpha)}(x) = \omega(x-a) + \xi\|x-a\|_1 - \gamma \tag{3.1}$$

$$\omega \in R^n, \xi \in R_+ = [0,+\infty], \gamma \geq 1, 1\ normu := \|.\|_1$$




**Şekil 3.1** *Çok yüzlü konik fonksiyon grafiği*

Bu yöntemin bir diğer varyasyonu da Öztürk ve Çiftçi (2011) çalışmasında ortaya konulan kümeleme temelli bir yaklaşımıdır. Bir sonraki bölümde, çok yüzlü konik fonksiyonlar ve kümeleme temelli çok yüzlü konik fonksiyonların işleyişi ayrıntılı bir şekilde anlatılmıştır.

## 3.1 Çok Yüzlü Konik Fonksiyonlar Algoritması

Bu bölümde ÇKF algoritmasının işleyiş yöntemi anlatılmıştır. $A$ ve $B$ gibi iki kümeyi ayırmaya çalışan algoritma, $A$ kümesine ait rastgele bir noktayı tepe noktası olarak belirleyerek başlar. Seçilen tepe noktasına göre, $A$ kümesine ait verileri olabildiğince fazla içeren ve $B$ kümesinden ise hiçbir veriyi içermeyen koniler oluşturulur. Elde edilen her koni sonrasında, bulunan koni ile doğru ifade edilebilen, yani koni içerisinde kalan $A$ kümesine ait veriler, kümeden çıkarılır. Tepe noktası seçme adımına dönülür ve $A$ kümesindeki bütün veriler çıkarılana kadar algoritma adımları tekrarlanır. Bir noktanın hangi sınıfa ait olduğu ise, o noktanın elde edilen konik fonksiyonlardaki en küçük değerine göre belirlenir. $A$ ve $B$ kümleri $R^n$' de verilmiş iki küme olsun:

$$A = \{a^i \in R^n : i \in I\} \; A = \{1, \dots, m\}$$

$$B = \{b^j \in R^n : i \in J\} \; J = \{1, \dots, p\}$$

***Başlangıç Adımı:*** $l = 1, l_l = I, A_\iota = A$ atamalarını yap. Adım 1'e git.

***Adım 1:*** $a^l$ noktası $A_l$ kümesinin herhangi bir noktası olsun. $P_l$ problemini çöz.

$$\omega(a^i - a^l) + \xi \left\| a^i - a^l \right\|_1 + \gamma + 1 \leq y_i, \quad \forall_i \in I_l \tag{3.2}$$

$$-\omega(b^i - a^l) - \xi \left\| b^i - a^l \right\|_1 + \gamma + 1 \leq 0, \quad \forall_j \in J_l \tag{3.3}$$

$$y = (y_1, \dots, y_m) \in R_+^m, \omega \in R^n, \xi \in R, \gamma \geq 1$$

kısıtları altında;

$$P(l) \; Enk \left( \frac{ye_m}{m} \right) \tag{3.4}$$

$P_l$ probleminin bir çözümünü bul, $\omega^l, \xi^l, \gamma^l, y^l$. Bu çözüme karşılık gelen çok yüzlü konik fonksiyonu aşağıdaki gibi oluştur.

$$g_1(x) = g_{(\omega^l, \xi^l, \gamma^l, y^l, a^l)}(x) \tag{3.5}$$

***Adım 2:*** $I_{l+1} = \{i \in I_l : g_l(a^i) + 1 > 0\}, A_{l+1} = A_l : i \in I_{l+1}\}$, $l = l + 1$ güncellemesini yap. Eğer $A_l \neq \emptyset$ ise Adım 1'e git.

***Adım 3:*** $A$ ve $B$ kümelerini ayıran $g(x)$ fonksiyonunu aşağıdaki gibi tanımla ve dur.

$$g(x) = Enk_l \; g_l(x) \tag{3.6}$$

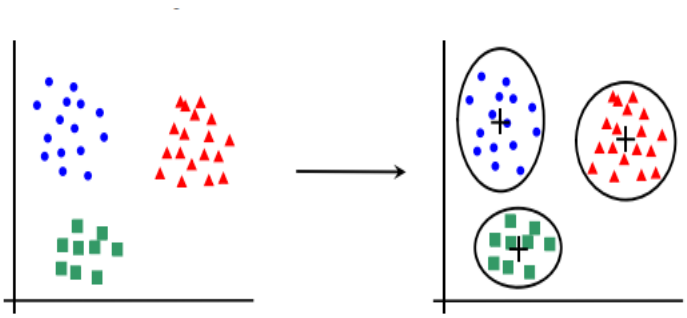
Algoritmadaki (3.2) kısıtı, amaç fonksiyonuyla beraber düşünüldüğünde $l$. adımda oluşturulan çok yüzlü konik fonksiyonun içerisinde kalan $A$ kümesine ait verileri olabildiğice arttırmaya çalışırken, (3.3) kısıtı, aynı çok yüzlü konik fonksiyonun içerisinde $B$ kümesine ait hiçbir verinin olmamasını sağlar. Bir verinin hangi sınıfa ait olduğuna karar verilirken, ilgili verinin elde edilen sonlu sayıdaki çok yüzlü konik fonksiyon değerlerine bakılır. Eğer herhangi bir fonksiyon değeri sıfır ya da negatif ise bu verinin $A$ kümesine ait olduğu, diğer durumda $B$ kümesine ait olduğu söylenir.

ÇKF algoritmasını ortaya konulduğu, Gasimov ve Öztürk (2006), çalışmada sunulan sonuçlar, yöntemin sınıflandırma problemlerini başarılı bir şekilde çözdüğünü göstermektedir. Ayrıca yöntemin seçilen tepe noktasına olan duyarlılığı ve aşırı uyum sorunu da ifade edilmiştir. Yöntemin ortaya konulmasından bu yana, ÇKF algoritması ile ilgili çalışmalar yapılmakta olup, yöntemin görüntü işleme ve nesne tanıma alanlarında başarılı sonuçlar verdiği Çimen (2013) [10], Çevikalp (2017) [11] çalışmalarında rapor edilmiştir.

## 3.2 Kümeleme Temelli Çok Yüzlü Konik Fonksiyonlar Algoritması

Kümeleme yöntemlerinde amaç, hangi sınıfa ait olduğu bilgisi önceden bilinmeyen verileri, tariflenmiş bir benzerlik değeri üzerinden, örneğin birbirlerine olan öklid uzaklıklarını kullanarak kümelere ayırmaktır. Bu sayede benzer özelliklere sahip veriler olabildiğince aynı kümede toplanırken, birbirlerine benzerlikleri düşük olan veriler farklı kümelerde toplanmaktadır. Şekil 3.2'de bir kümeleme örneği görülmektedir.

Kümeleme yöntemlerine ait literatürde, MacQueen [12] tarafından sunulmuş olan $k$-ortalamalar algoritması en ünlü yöntemlerden birisidir. $N$ elemanlı bir veri kümesi üzerinde yöntemin işleyişi aşağıdaki gibidir.



**Şekil 3.2** *Kümeleme örneği*

***Başlangıç adımı:*** $k \leq N$ olacak şekilde, $k$ küme sayısını belirle ve rassal olarak her küme için bir merkez $c_k$ belirle.

***Adım 1:*** Veri noktalarının, Öklid uzaklıklarını hesaplayarak kendilerine en yakın olan küme merkezine, $c_k$ 'ya ata.

***Adım2:*** Bütün küme merkezlerini, kendilerine atanmış veri noktalarının aritmetik ortalamalarını alarak tekrar hesapla. Bütün veri noktalarının atandığı küme merkezleri sabit kalana kadar Adım 1'e dön.

*k*-ort-ÇKF algoritması, ÇKF algoritmasının tepe noktası seçimine olan duyarlılığını azaltmak ve büyük boyutlu problemlerde uygulanabilirliğini sağlamak amacıyla sunulmuştur. Kümeleme temelli ÇKF yönteminin işleyişi aşağıda gibidir. $p$ sınıfa ait verilerin birleşiminden oluşan $A$ veri kümesi $R^n$'de verilmiş olsun:

***Başlangıç Adımı:*** (3.7)'ye göre $p$ adet $B_j$ kümesi oluşturulur, $j = 0$ ataması yapılır ve $k$ küme sayısı belirlenir.

$$B_j = \bigcup_{l=1, l \neq j}^{p} A_l \qquad j = 1, \dots, p \tag{3.7}$$

***Adım 1:*** $j = j + 1$ ataması yapılır ve $A_j$ ve $B_j$ kümeleri seçilir.

***Adım 2:*** $A_j$ veri kümesine $k$-ortalamalar algoritması uygulanır. İlgili kümeye ait $k$ adet alt küme $A_{jr}$ ve bu altkümelerin merkezleri olan $c_{jr} \in R^n, r = 1, \dots, k$ değerleri elde edilir.

***Adım 3:*** Adım 2'de bulunan her $A_{jr}, r = 1, \dots, k$, kümesi için $(P_{jr})$ problemi çözülür. Çözümden elde edilen, $w_{jr}, \xi_{jr}, \gamma_{jr}$ parametreleri ile $g_{jr}, r = 1, \dots, k$ çok yüzlü konik fonksiyonu oluşturulur.

$$\omega_{jr}\left(a_i - c_{jr}\right) + \xi_{jr}\left\|a_i - c_{jr}\right\|_1 - \gamma_{jr} \leq y_i, \qquad \forall_i \in I_{jr} \tag{3.8}$$

$$-\omega_{jr}\left(a_l - c_{jr}\right) + \xi_{jr}\left\|a_l - c_{jr}\right\|_1 - \gamma_{jr} \leq z_{i,} \ \forall_l \in I_B \tag{3.9}$$

$$I_{jr} = \{i : \ a_i \in A_{jr}\} \quad I_B = \{i : \ a_i \in B_j\}$$

kısıtları altında:

$$\left(P_{jr}\right) Enk \tag{3.10}$$

***Adım 4:*** Sınıflandırıcı fonksiyon (3.11)'e göre oluşturulur.

***Adım 5:*** Eğer $j < p$ ise Adım 1'e dön, diğer durumda algoritmayı sonlandır.

$$g_j(x) = Enk_{r=1,\dots,k} g_{jr}(x) \tag{3.11}$$

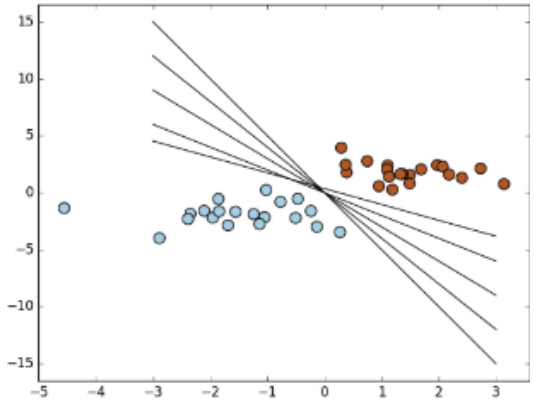
Yapılan çalışmada, ortaya konulan yöntemin, beklenildiği gibi büyük boyutlu problemlerin çözümünde kullanılabileceği test sonuçları ile gösterilmiştir. Ayrıca yöntemin seçilen küme merkezlerine bağımlı olmadığı ancak seçilen $k$ küme sayısına bağımlı olduğu tespit edilmiştir. Elde edilen sonuçların başarılı olması dolayısıyla, kullanılan kümeleme yöntemlerinin *k*-ort-ÇKF algoritmasına olan etkisini Çimen (2013) çalışmasında incelenmiştir. Çalışmada ele alınan kümeleme yöntemlerinin, algoritma hızını etkilediği raporlanmıştır.

# 4. DESTEK VEKTÖR MAKİNELERİ

Literatürde çokça bahsedilen, başarısını ispatlanmış sınıflandırma yöntemlerinden birisi de Vapnik tarafından ortaya atılan DVM'dir. Bu yöntemin başarısının, iki ana unsuru bulunmaktadır. Birincisi, yöntemin eğitim kümesindeki hatayı en küçüklemek yerine, tahminleme yapılırken olabilecek hataları en küçükleyen amaç fonksiyonu kullanması. İkincisi ise verilen sınıflandırma problemini, düşük bir maliyet daha büyük boyutlu uzaylara taşıyabilmesidir.

## 4.1 Doğrusal Ayrılabilir Durum

Şekil 4.1'de olduğu gibi doğrusal olarak birbirinden ayrılabilen iki veri kümesi düşünüldüğünde, bu iki kümeyi ayırıan sonsuz sayıda yüzey bulunduğu görülecektir. Örneğin, Bennet ve Mangasarian'ın (1992) çalışmalarında ortaya koydukları yöntem, iki kümeyi tam olarak ayıran kümelere belirli bir uzaklıktaki, herhangi bir yüzeyi bulmaktadır. DVM ise, iki veri kümesine de olabildiğince uzak olan ayırıcı yüzeyi bulmayı amaçlamaktadır. Bu sayede ileride yapılabilecek olan hatalı tahminlerin enküçüklenmesi sağlanmaktadır. Bu amaç literatürde deneysel risk en küçüklenmesi olarak ifade edilmektedir [13].



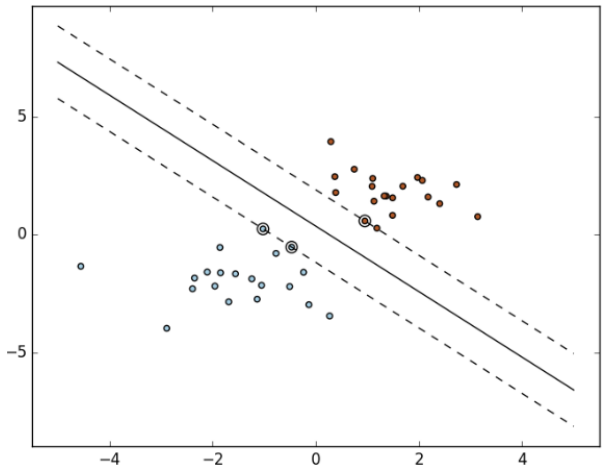

**Şekil 4.1** *Doğrusal olarak ayırabilen yüzeyler.*

DVM, iki kümeyi en iyi ayıran yüzeyi, marj olarak isimledirilen, iki kümenin birbirlerine yakın yüzeylerinde bulunan uç noktaların üzerinden geçen hiper düzlemler arasındaki mesafeyi, en büyükleyerek bulmaktadır. Şekil 4.2'de görülen kesikli çizgilerin üzerinden geçtiği noktalar destek vektörleri, kesikli çizgilerle gösterilen hiper düzlemler arasındaki mesafe ise marj değeridir. Ortadaki sürekli çizge ise bulunması amaçlanan en büyük marj değerine sahip ayrıcı yüzeydir.

Sınıf bilgileri, $y = \{+1, -1\}^m$ olan, $A_{m_1 \times n}$ ve $B_{m_2 \times n}$ veri kümleri verilerek $X = A \cup B$, $x_i \in X, i = \{1, \dots, m\}$ , olarak tanımla tanımlandığında, DVM tarfından bulunmak istenen yüzey (4.1) gibi ifade edilir ve işlem kolaylığı açısından, (4.2) olarak düzenlenir.

$$\omega x_i + b \geq +1 \; eğer \; y_i = +1, \qquad \omega x_i + b \leq -1 \; eğer \; y_i = -1 \tag{4.1}$$

$$y_i(\omega x_i + b) \geq +1 \tag{4.2}$$




**Şekil 4.2** *En büyük marj değerine sahip ayırıcı yüzey*

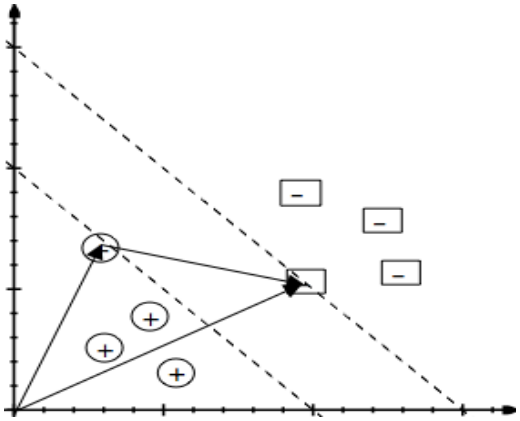
Devam eden kısımlarda, ifade kolaylığı açısından $x_+ \in A$, $x_- \in B$ olarak, ve $||.||$ iki normu olarak kullanılmıştır. Şekil 4.3 üzerinden de görülebileceği üzere, marj genişliği değeri denklem (4.3) kullanılarak hesaplanabilmektedir. (4.3) ve (4.2) beraber ele alındığında (4.4)'de gösterilen değerler elde edilir ve marj genişliği değeri, (4.5)'de görüldüğü gibi düzenlenir.

$$Marj \; genişliği = (x_+ - x_-) \times \frac{\omega}{\|\omega\|} \tag{4.3}$$

$$\omega x_+ = 1 - b, \qquad \omega x_- = 1 + b \tag{4.4}$$

$$Marj \; genişliği = \frac{2}{\|\omega\|} \tag{4.5}$$

$$y_i(\omega x_i + b) \leq +1 \; \forall_i \; k.a. \;\; Enk = \frac{1}{2} \|\omega\|^2 \tag{4.6}$$



**Şekil 4.3** *Marj değeri hesaplanması*

Tariflenen kısıtlı fonksiyon (4.6), Lagrange çarpanları kullanılarak kısıtsız hale getirilir. Karush-Kuhn Tucker koşulları kullanılarak, denklem (4.10) da görüldüğü gibi tekrar düzenlenir.

$$L_p = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{m} \alpha_i \left[y_i(\omega x_i + b) - 1\right] \tag{4.7}$$

$$\frac{dL}{d\omega} = \omega - \sum_{i=1}^{m} \alpha_i \, y_i x_i = 0 \quad \sum_{i=1}^{m} \alpha_i \, y_i x_i = \omega \tag{4.8}$$

$$\frac{dL}{db} = -\sum_{i=1}^{m} \alpha_i \, y_i = 0 \qquad \sum_{i=1}^{m} \alpha_i \, y_i = 0 \tag{4.9}$$

$$L_p = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i} \sum_{j} \alpha_i \alpha_j y_i y_j x_i x_j \tag{4.10}$$

$$\sum_{i=1}^{m} \alpha_i \, y_i x_i u > 0 \;\; ise + 1, değil \; ise - 1 \tag{4.11}$$

Denklem (4.10)'un çözümde, $\alpha_i > 0$ koşulunu sağlayan noktalar destek vektörleri olarak belirlenir ve ayırıcı yüzeyleri tarifleyen denklemler yardımıyla, bulunmak istenen hiper düzlemin $b$ parameteresi hesaplanır. Karar fonksiyonu (4.11) denklemine göre düzenlendiğinde, $u \in R^n$ gibi bilinmeyen bir noktanın sınıf değerinin, kendisi ile destek vektörlerinin noktasal çarpımına eşit olduğu görülür. Bu sonuç sayesinde, çekirdek-hilesi (kernel-trick) olarak isimlendirilen yöntem kullanılarak, veriler düşük bir maliyet ile daha yüksek boyutlu bir uzayda hesaplanıyormuşçasına çözülür [5].

## 4.2 Çekirdek Hilesi ve Doğrusal Ayrılamayan Durum

Çoğu problem bulundukları $n$ boyutlu uzayda doğrusal olarak ayrılamayıp, $n+k$ gibi daha yüksek boyutlu bir uzaya taşınarak burada doğrusal olarak ayrılabilmektedirler. Boyut artırma işleminde $k$ değerinin belirli olmamasının yanı sıra, ciddi bir hesaplama yükü getirmektedir. Denklem (4.10)'da görüldüğü üzere destek vektörleri, verilerin iç çarpımlarının ağırlıklı toplamı ile hesaplanabildiğinden, verileri $n+k$ boyutuna taşıyarak ilgili uzayda işlem yapmak yerine, verilerin $(n+k)$ boyutlu uzaydaki iç çarpım değerlerini veren çekirdek fonksiyonları kullanarak yapılması mümkün olmaktadır. Bize verilen $n=2$ boyutlu $x, x'$ noktalarını $k=4$ olacak şekilde 6 boyutlu bir uzaya taşıyarak iç çarpımlarını bulmak istediğimizi ve $z, z'$ noktalarının verilen $x, x'$ noktalarının istenilen uzaydaki karşılıkları olduğunu düşünelim.

$$x = (x_1, x_2) \quad z^T z' = K(x, x') \tag{4.12}$$

$$z = \phi(x) = \left(1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2\right) \tag{4.13}$$

$$K(x, x') = (1 + x_1 x_1' + x_2 x_2' + 2x_1^2 x_1'^2 + 2x_2^2 x_2'^2 + 2x_1 x_1' x_2 x_2') \tag{4.14}$$

Çekirdek hilesi kullanılmadığı durumda, noktaların istenilen uzaydaki çarpımlarını bulabilmek için, $\phi(x)$ fonksiyonu yardımı ile noktalar, denklem (4.13)'de görüldüğü gibi istenilen uzaya taşınarak ilgili uzaydaki çarpımları hesaplanmalıdır (4.14).

$$K(x, x') = (1 + x^T x')^2 = (1 + x_1 x_1' + x_2 x_2')^2 \tag{4.15}$$

$K(x, x')$ fonksiyonu, (4.15)'de görüldüğü gibi tanımlandığında, buradan elde edilen sonucun, $x, x'$ noktalarının (4.13)'de gösterildiğibi istenilen uzaya taşındıktan sonra (4.14)'de elde edilen iç çarpım değerine eşit olduğu görülecektir. Buradaki dikkate değer olan durum, denklem (4.15)'de verilerin başka bir boyuta taşınmasına gerek olmadığıdır.

$$L = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left[y_i(\omega x_i + b) - 1 + \xi_i\right] + \sum_{i=1}^{m}\mu_i\xi_i \tag{4.16}$$

Boyut artırımı yapılsa dahi, gerçek hayat problemlerinin pek çoğu ayrılamayan iç içe geçmiş veri kümelerinden oluşmaktadır. Bu durumda yanlış sınıflandırma yapmadan iki kümeyi birbirinden ayıran yüzey bulunamamaktadır. Bu sorunun üstesinden gelebilmek için, (4.10)'da ifade edilen denkleme bir ceza parametresi $C$ eklenerek, yapılan hatalı sınıflandırmalar $\xi$ ile marj genişliği arasında bir ödünleşim sağlanır (4.16). $\mu$ değeri ilgili $\xi$ ye ait Lagrange çarpanıdır. Bu çalışmada, karşılaştırmalarda

kullanılan DVM yöntemi, (4.16) ile ifade edilen, çekirdek hilesinin kullanılmadığı formulasyonu ifade etmektedir.

## 4.3 Genelleştirilmiş Öz Değer Problemi Destek Vektör Makineleri

Mangasarian ve Wild (2006), tarafından ortaya konulan GÖPDVM yöntemi Fung ve Mangasarian (2001) [14] tarafından ortaya atılan, Yaklaşık Destek Vektör Makineleri (YDVM) yönteminin bir uzantısı olarak ifeda edilebilir. DVM yöntemine karşılık, karmaşık karesel optimizasyon problemi çözdürülmesini gerektirmeyen yöntemin farklı varsyasyonları da bulunmaktadır [15,16,17].

Bir önceki bölümde anlatılan klasik DVM, YDVM ve GÖPDVM arasındaki fark, marj değerinin tariflenmesine dayanmaktadır. Bu farklar kısıtlar üzerinden şu şekilde açıkalanabilir. DVM yönteminde denklem (4.1,4.2) ile verilen kısıtlar, YDVM yönteminde (4.13) olarak tariflenir.

$$x'\omega_1 - \gamma_1 = 0, x'\omega_2 - \gamma_2 = 0 \tag{4.17}$$

Bu sayede elde edilen destek vektörleri uç noktalardan daha içerideki noktalara taşınmış olur ve $A, B$ veri kümelerini en iyi şekilde tarifleyen birbirlerinden olabildiğince uzak paralel yüzeyler elde edilir. GEPSVM, yönteminde ise marj değeri hesaplanırken paralellik şartı gözetilmez. Şekil 4.4'de farklı marj değeri yaklaşımlarının görsel bir karşılaştırması görülmektedir.

Bu bölümde, bir üst bölümde kullanılan notasyonlara sadık kalınarak, ilgili boyutlara uygun birlerden oluşan sütun vektörü $e$ ve birim matrisi $I$ kullanılmıştır. GÖPDVM yöntemi, $A, B$ gibi iki küme verildiğinde, $\omega_1, \gamma_1$ parametrelerine sahip $A$ kümesine en yakın $B$ kümesine en uzak bir düzlem ile, $\omega_2, \gamma_2$ parametrelerine sahip $B$ kümesine en yakın $A$ kümesine en uzak bir diğer düzlem oluşturmayı amaçlamaktadır. Bu amaçlar doğrultusunda, problem (4.18) olarak tariflenir.

$$\underset{(w,\gamma)\neq 0}{\text{Enk}} \frac{\|A\omega_1 - eb_1\|^2}{\|B\omega_1 - eb_1\|^2} \tag{4.18}$$

Daha sonra amaç fonksiyonuna düzlem parametrelerinin normunu azaltan Tikhonov düzenleme terimi $\delta > 0$, eklenerek (4.19) denklemi elde edilir.

$$\underset{(w,\gamma)\neq 0}{\text{Enk}} \frac{\|A\omega_1 - eb_1\|^2 + \delta \left\| \begin{bmatrix} \omega_1 \\ b_1 \end{bmatrix} \right\|^2}{\|B\omega_1 - eb_1\|^2} \tag{4.19}$$

Yukarıda ifade edilen optimizasyon problemi, $R^{(n+1)\times(n+1)}$ $G$ ve $H$ simetrik matrislerinin (4.20), (4.21) olarak tanımlanması ile, (4.22)'de görülen şekilde düzenlenir.

$$G \coloneqq [A-e]^T[A-e] + \delta I \tag{4.20}$$

$$H \coloneqq [B-e]^T[B-e], z_1 \coloneqq \begin{bmatrix}\omega_1 \\ b_1\end{bmatrix} \tag{4.21}$$

Denklem (4.22) ile ifade edilen optimizasyon probleminin en iyi değeri, (4.23)'de gösterilen ve Rayleigh Quotient özellikleriyle çözülebilen genelleştirilmiş özdeğer probleminin çözümünden elde edilir.

$$\underset{z\neq 0}{\text{Enk}}\, r(z_1) \coloneqq \frac{z_1^T G z_1}{z_1^T H z_1} \tag{4.22}$$

$$Gz_1 = \lambda H z_1, z_1 \neq 0 \tag{4.23}$$

Denklem (4.22) ile tariflenen en küçükleme probleminin, en iyi değeri, denklem (4.23) ile ifade edilen problemin, en küçük özdeğerine karşılık gelen, özdeğer vektörü $\lambda_1$ kullanılarak elde edilir, $z_1 = [\omega_1\ b_1]^T$. Daha sonra $B$ kümesine en yakın $A$ kümesine olabildiğince uzak olan, yüzeyi bulabilmek amacıyla aşağıdaki düzenlemeler yapılır ve (4.27) problemi çözülerek $z_2 = [\omega_2\ b_2]^T$, parametleri elde edilir.

$$\underset{(w,\gamma)\neq 0}{\text{Enk}} \frac{\|B\omega_2 - eb_2\|^2 + \delta \left\|\begin{bmatrix}\omega_2 \\ b_2\end{bmatrix}\right\|^2}{\|B\omega_2 - eb_2\|^2} \tag{4.24}$$

$$L \coloneqq [B-e]^T[B-e] + \delta I, M \coloneqq [A-e]^T[A-e] \tag{4.25}$$

$$\underset{z\neq 0}{\text{Enk}}\, s(z_2) \coloneqq \frac{z_2^T L z_2}{z_2^T M z_2} \tag{4.26}$$

$$Lz_2 = \lambda M z_2, z_2 \neq 0 \tag{4.27}$$

Elde edilen düzlem parametreleri ile, karar fonksiyonu (4.28)'de gösterildiği gibi tariflenerek, hangi sınıfa ait olduğu bilinmeyen bir noktanın sınıf bilgisi, kendisine en yakın düzleme göre belirlenir.

$$\underset{h=1,2}{\text{Enk}}(x) = |\omega_h x - b_h\,| \tag{4.28}$$

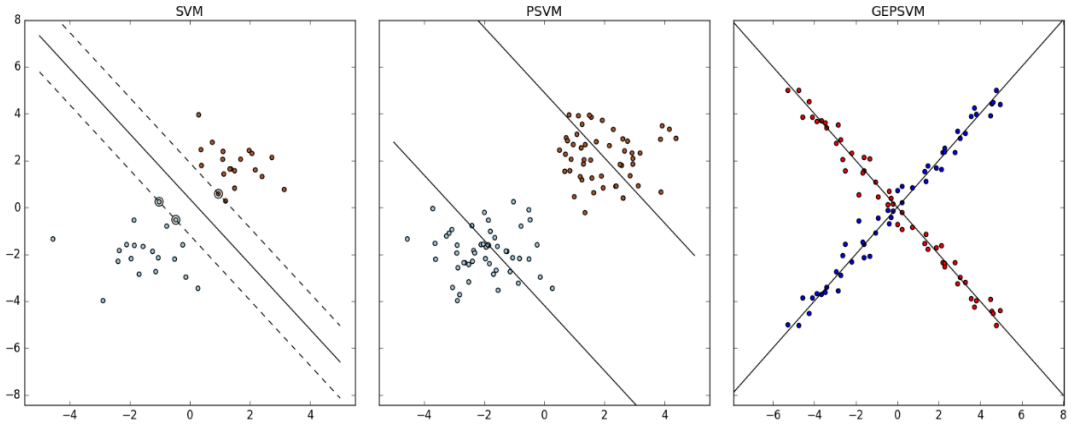
Çalışmada öne sürülen, GÖPDVM yönteminin, çekirdek fonksiyonu kullanılarak geliştirilen bir versiyonu da sunulmuştur. GÖPDVM yönteminde çözdürülmesi gereken genelleştirilmiş özdeğer probleminin karmaşıklığı $O(n^3)$ iken, klasik DVM yönteminde çözdürülmesi gereken karesel programlama probleminin karmaşıklığı $O(n^{3.5})$'dur [5]. Bu durum GÖPDVM yönteminin daha hızlı çalışmasını sağlamaktadır. Ayrıca özdeğer probleminin çözümü tek satır kod ile Matlab, Scilab, Scipy gibi

ortamlarda çözdürülebilirken, karesel programlama probleminin çözümü için özelleşmiş karmaşık kodlara ve çözücülere ihtiyaç duyulmaktadır. Bahsedilen avantajlarının yanı sıra, GÖPDVM yönteminin diğer sınıflandırma yöntemleri ile aynı başarı ile problemlere çözüm getirdiği, yapılan deneyler ile gösterilmiştir.

# 5. MARJ ENBÜYÜKLENMESİ

Sınıflandırma yöntemleri, mevcut sınırlı sayıda verilere dayanarak ilgilenilen durumun bütün uzayını en iyi şekilde ifade etmek isterler. Bir yöntemin eğitim kümesine dayanarak geri kalan uzayı tarif edebilme yeteneği genelleştirme başarısı denmektedir. Eğitim süresi arttıkça, eğitim aşamasında ortaya çıkan hatanın giderek azalırken, test aşamasında ortaya çıkan hatanın giderek artması olarak tariflenebilen aşırı uyum sorunu ise genelleştirme başarısının düşük olduğunu gösteren bir ölçüdür. Aşırı uyum sorunu eğitim ve test başarıları arasındaki farka dayanarak söylenebilmektedir.

Şekil 5.1'de aşırı uyum sorunu görselleştirilmiştir. Şekilde noktalarla gösterilen veriler eğitim kümesi olup, asıl veriyi ifade eden gerçek fonksiyon düz yeşil çizgi ile ifade edilmiştir. Tahmin için kullanılan model fonksiyon ise düz mavi çizgi ile gösterilmiştir.

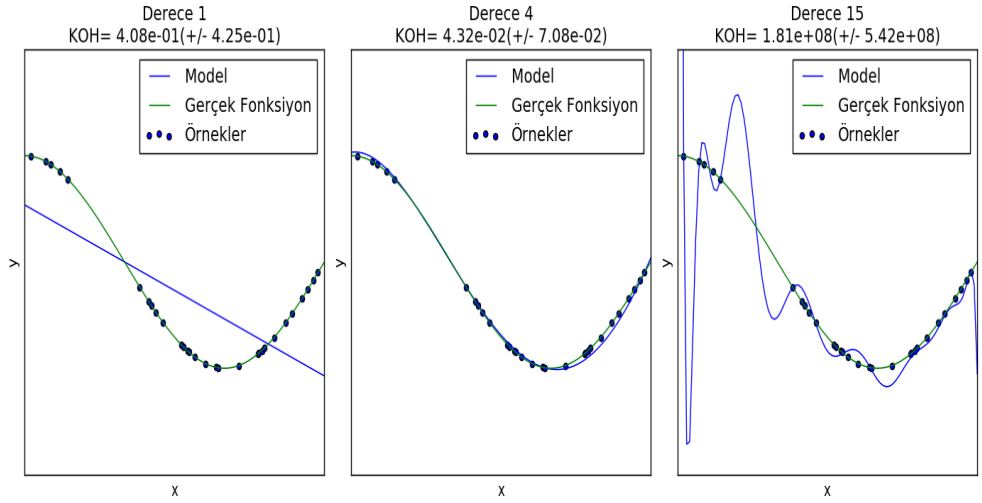



**Şekil 5.1** *Farklı marj değeri yorumları*

Şekil 5.2'de tahminleme modeli olarak sırasıyla, 1.,4. ve 15. dereceden polinomlar kullanılmıştır. Şekiller incelendiğinde eğitim verilerine en uygun modelin 15. dereceden polinom kullanarak elde edildiği görülmektedir. Ancak gerçek fonksiyonun değerlerine göre hesaplanan test hatalarına bakıldığında, en büyük hatanın en uyumlu olduğu düşünülen modelde oluştuğu görülmektedir.

Şekillerde görülen polinomlarda kullanılan her derece, veri noktlarını güçlü bir varsayım haline getirmektedir. Ancak eğitim kümesi içerisinde, gürültü olarak adlandırılan veriler de bulunduğundan varsayımlar yanlış olmaktadır. Aynı mantıkla, ÇKF algoritmasında elde edilen koniler de ilgili veriye dair varsayımlar olarak düşünüldüğünde, *A* kümesine ait bütün noktaları doğru bir şekilde tarifleyene kadar durmayan ÇKF algoritması, yüksek dereceli bir polinom olarak görülebilir.

Optimizasyon çalışmalarının diğer alanlarında, tariflenen amaç fonksiyonunun en iyi değerini bulmak yararlı bir durum iken, sınıflandırma problemlerinde durum farklı olabilmektedir. Sınıflandırmadaki amacımız bilinenen veri kümesini en iyi şekilde tariflemek yerine, bilinen ve daha önce karşılaşılmamış verilerin sınıf tahminlerini başarılı şekilde yapacak bir model geliştirmektir.




**Şekil 5.2** *Aşırı uyumun bir görselleştirilmesi*

Diğer bir olası ancak o kadar açık görülmeyen bir sebep ise $\xi$ parametresidir. Bulunan konilerin yüzey alanı $B$ kümesine ait karşılaşılan ilk nokta tarafından sınırlandırılmaktadır. $n$ boyut bir uzayda, $i = \{1, \dots, n\}$, bir koni, $B$ kümesine ait bir nokta ile $i$. boyutta karşılaştığında, sadece $i$. boyutta değil bütün yönlerde büyümesini durdurmaktadır.

## 5.1 Destek Vektör Makineleri Yaklaşımı

Marjin değeri en büyüklenmesinin, çok yüzlü konik fonksiyonlara bir uyarlaması olan bu yaklaşımın temeli, (3.1) ile ifade edilen çok yüzlü konik fonksiyon denkleminin, DVM yönteminde bulunması amaçlanan ayırıcı yüzey türünden ifade edilmesine dayanmaktadır.

Denklem (3.1) ile tarif edilen ayrıcı konik yüzey, denklem (5.1)'de görüldüğü gibi tariflenir. Bu tariflemede $w^* = \begin{bmatrix} \omega \\ \xi \end{bmatrix}$ matrisi, (4.2)'de gösterilen DVM yönteminde bulunmaya çalışılan ayırıcı yüzeyin katsayılarına, $\gamma$ değeri ise $b$ sabitine gelmektedir. $d$ olarak ifade edilen değer ise, $x$ noktasının $R^n$'den seçilen bir $c$ değerine göre merkezileştirilmiş mutlak değerlerini içerir. Denklem (5.1) ile yapılan bu dönüşüm,

klasik DVM formülasyonunda bir değişim getirmeyip, problem (4.10) ya da (4.12)'de tariflendiği gibi çözülebildiği açıktır.

$$g(x) = \begin{bmatrix} \omega \\ \xi \end{bmatrix}_{(n+1)\times 1} [x \quad d]_{1\times(n+1)} - \gamma \tag{5.1}$$

$$d = \sum_{i=1}^{n} |x_i - c_i| = \|x - c\|_1 \ , c \in R^n \tag{5.2}$$

Bu yaklaşımda, karar fonksiyonu için iki farklı varyasyon geliştirilmiştir. İlk varyasyonda, problem çözümünden elde edilen parametreler ile (3.1)'de tarif edilen koniler oluşturulmuştur. Dikkat edilmesi gereken bir durum, elde edilen parametrelerle oluşan konilerin konumlarıdır.

Ayrılmak istenen yüzey ile kesişmeyen ve farklı yönlere bakan koniler oluşabilmektedir. ÇKF algoritması $\xi, \gamma > 0$ kısıtları sayesinde sorunu aşarken, yapılan değişimle bu kısıtlar ortadan kalktığından, koniler elde edilen parametrelerin mutlak değerleri alınarak oluşturulmuştur.

$$g_{(\omega,|\xi|,|\gamma|,\alpha)}(x) \leq 0 \ ise -1, değil \ ise +1 \tag{5.3}$$

İkinci varyasyonda ise, karar fonksiyonu olarak (4.10)'da gösterilen klasik DVM karar kuralı kullanılmıştır. Bu karar kuralının kullanılabilmesi için, verilen $n$ boyutlu bir $x$ noktasının, ilgili $c$ değerine göre $n+1$. boyuta taşınması gerekmektedir.

$$w^*[x\ d] + \gamma \leq 0 \ ise -1, değil \ ise +1 \tag{5.4}$$

## 5.2 Genelleştirilmiş Özdeğer Problemi Temelli Konik Fonksiyonlar

Bu yaklaşımda, GÖPDVM yönteminde bulunan en yakın düzlem ile bir analoji geliştirilerek, veri kümlerini iyi ifade eden konilerin oluşturulması amaçlanmaktadır. Bu doğrulta bir önceki bölümde denklem (5.1) ile yapılan DVM dönüşümü denklem (5.5)'de görüldüğü gibi genişletilerek GÖPDVM formulasyonuna uygun hale getirilmişir.

$$g(x) = \begin{bmatrix} \omega \\ \xi \\ \gamma \end{bmatrix}_{(2n+1)\times 1} [x \quad d \quad -e]_{1\times(2n+1)} \tag{5.5}$$

$$d = [|x_0 - c_0| \ |x_1 - c_1| \ \dots \ |x_n - c_n|]_{1\times n}, c \in R^n$$

Burada yapılan önemli değişkiliklerden birisi, $\xi$ parametresinini $R^n$'den seçilerek $d$ matrisinin $n$ boyutlu olarak tanımlanmasıdır. Yapılan tanımlamalar göz önüne alındığında, A kümesine en uygun koni önerilen algoritma ile aşağıdaki gibi bulunmaktadır:

***Adım 1:*** $A$ kümesinin merkezini hesaplayarak $c_a$ değişkenine ata.

***Adım 2:*** $D_A$ ve $D_B$ matrislerini, $a \in A, b \in B$, aşağıdaki gibi oluştur.

$$D_A = \begin{bmatrix} |a_{11} - c_a| & \cdots & |a_{1n} - c_a| \\ \vdots & \ddots & \vdots \\ |a_{m1} - c_a| & \cdots & |a_{mn} - c_a| \end{bmatrix}_{m_1 \times n}$$

$$D_B = \begin{bmatrix} |b_{11} - c_a| & \cdots & |b_{1n} - c_a| \\ \vdots & \ddots & \vdots \\ |b_{m1} - c_a| & \cdots & |b_{mn} - c_a| \end{bmatrix}_{m_2 \times n}$$

***Adım 3:*** $G$ ve $H$ matrislerini oluştur.

$$G \coloneqq [A \quad D_A \quad -e]^T \times [A \quad D_A \quad -e] + \delta \times I$$

$$H \coloneqq [B \quad D_B \quad -e]^T \times [B \quad D_B \quad -e], z_1 \coloneqq [\omega_1, \xi_1, \gamma_1]^T$$

**Adım 4:** Problem (5.6)'yı çöz ve elde edilen $\omega_1, \xi_1, \gamma_1$, denklem (3.1)'e göre $g_1$çok yüzlü konisini oluştur.

$$Gz_1 = \lambda Hz_1, z_1 \neq 0 \tag{5.6}$$

$B$ kümesine en iyi ifade eden koniyi bulmak amacıyla, $Adım$ 1'de $B$ kümesinin merkezi hesaplanarak $c_b$ değişkenine atanır. $D_A$ ve $D_b$ matrisleri:

$$D_A = \begin{bmatrix} |a_{11} - c_b| & \cdots & |a_{1n} - c_b| \\ \vdots & \ddots & \vdots \\ |a_{m1} - c_b| & \cdots & |a_{mn} - c_b| \end{bmatrix}_{m_1 \times n}$$

$$D_B = \begin{bmatrix} |b_{11} - c_b| & \cdots & |b_{1n} - c_b| \\ \vdots & \ddots & \vdots \\ |b_{m1} - c_b| & \cdots & |b_{mn} - c_b| \end{bmatrix}_{m_2 \times n}$$

şeklinde hesaplanarak $L$ ve $M$ simetrik matrisleri oluşturulur. Problem (5.7) çözülerek bulunan $\omega_2, \xi_2, \gamma_2$ parametreleri ile $g_2$ çok yüzlü konisi oluşturulur.

$$L \coloneqq [B \quad D_B \quad -e]^T \times [B \quad D_B \quad -e] + \delta \times I$$

$$M \coloneqq [A \quad D_A \quad -e]^T \times [A \quad D_A \quad -e], z_2 \coloneqq [\omega_2, \xi_2, \gamma_2]^T$$

$$Lz_2 = \lambda Mz_2, z_2 \neq 0 \tag{5.7}$$

Bir noktanın hangi sınıfa ait olduğu bilgisine, oluşturulan konik fonksiyonlardaki mutlak değerinin en küçüğüne bakılarak karar verilir. Mutlak değerlerinin alınması, bir önceki yaklaşımda ifade edilen, konilerin yönlerinden kaynaklı yaşanan problemin üstesinden gelmektedir.

$$\underset{c=1,2}{\text{Enk}} |g_c(x)|$$

## 5.3 Ceza Parametreli Yaklaşım

Bir üst bölümde yapılan incelemede, ÇKF algoritmasının elde edilen koni sayısının fazla olmasından dolayı, yüksek dereceli bir polinom gibi görülebileceği ifade edilmiştir. Önerilen ceza parametreli yaklaşımda, ÇKF algoritmasından elde edilen koni sayısını sınırlamak amacıyla çeşitli düzenlemeler yapılmıştır.

İlk olarak (3.3)'de ifade edilen $B$ kümesine ait kısıt $\leq z_j$ olarak değiştirilmiş ve elde edilen konilerin içerisine $B$ kümesinden verilerin de olmasına izin verilmiştir. Diğer önemli bir düzenleme ise amaç fonksiyonundadır. Burada, $y, z$ değerleri kullanılarak hatalı tarfilenen veriler en küçüklenmeye çalışılırken, aynı zamanda koni tariflemede kullanılan $\omega, \xi, \gamma$ parametreleri de en küçüklenmiştir. A ve B kümleri $R^n$' de verilmiş iki küme olsun:

$$A = \{a^i \in R^n : i \in I\}\ I = \{1, \dots, m\}\ B = \{b^j \in R^n : i \in J\}\ J = \{1, \dots, p\}$$

***Başlangıç Adımı:*** $l = 1, l_l = I, A_l = A,\ j = 1, J_j = J, B_j = B, C \in R, c_1, c_2 = C$ atamalarını yap. Adım 1'e git.

***Adım 1:*** $a^l$ noktası $A_l$ kümesinin herhangi bir noktası olsun. $P_l$ problemini çöz.

$$\omega(a^i - a^l) + \xi \left\| a^i - a^l \right\|_1 + \gamma + 1 \leq y_i, \quad \forall_i \in I_l \tag{5.8}$$

$$-\omega(b^j - a^l) - \xi \left\| b^j - a^l \right\|_1 + \gamma + 1 \leq z_j, \quad \forall_j \in J_l \tag{5.9}$$

$$y = (y_1, \dots, y_m) \in R_+^m, z = (z_1, \dots, z_p) \in R_+^p, \omega \in R^n, \xi \in R, \gamma \geq 1$$

kısıtları altında;

$$P(l)\ Enk\left(c_1\left(\frac{ye_m}{m} + c_2\frac{ze_p}{p} + \frac{\|\omega\|_2^2 + \xi + \gamma}{(n+2)}\right)\right) \tag{5.10}$$

$P_l$ probleminin bir çözümünü bul $\omega^l, \xi^l, \gamma^l, y^l$. Bu çözüme karşılık gelen çok yüzlü konik fonksiyonu aşağıdaki gibi oluştur.

$$g_l(x) = g_{(\omega^l, \xi^l, \gamma^l, y^l, a^l)}(x) \tag{5.11}$$

***Adım 2:*** *A, B* kümelerini ve ceza parametrelerini güncelle. Eğer $A_l = \emptyset$ ya da $B_j = \emptyset$ ise Adım 3'e değilse Adım 1'e git.

$$I_{l+1} = \{i \in I_l : g_l(a^i) + 1 > 0\}, A_{l+1} = A_l : i \in I_{l+1}\}, l = l + 1$$

$$J_{j+1} = \{j \in J_j : g_l(b^j) + 1 > 0\}, B_{j+1} = B_j : j \in J_{j+1}\}, j = j + 1$$

$$\mathrm{c}_1 = \mathrm{C} \times \sqrt{|\,g| + 1}\,, \mathrm{c}_2 = \frac{\mathrm{C}}{\sqrt{|\,g| + 1}} \tag{5.12}$$

***Adım 3:*** $A$ ve $B$ kümelerini ayıran $g(x)$ fonksiyonunu aşağıdaki gibi tanımla ve dur.

$$g(x) = Enk_l \; g_l(x) \tag{5.13}$$

Güncelleme adımında ise, $l$. iterasyonda sadece $A$ kümesi değil $B$ kümesi de güncellenerek elde edilen koni içerisinde kalan veriler ilgili kümelerden çıkarılmıştır. Ayrıca ceza parametresi olarak kullanılan $c_1$ ve $c_2$ değerleri, $l$. iterasyonda bulunan koni sayıları gözetilerek güncellenir.

Yöntem tasarlanırken, ilk olarak sadece $c_1$ parametresi kullanılmış, ancak bu durumda $B$ kümesinden oluşan hataların yeterince kontrol edilemediği ve algoritmanın veriyi yeterince öğrenemeden durduğu görülmüştür.

Yapılan gözlem üzerine, modele $c_2$ parametresi eklenerek, güncelleme adımında $c_1$ parametresi artırılırken, $c_2$ parametresi azaltılmıştır. Bu seçimin nedeni, iterasyonlar ilerledikçe elde edilen konilerin maliyetini artarken $B$ kümesinden oluşan hata maliyetini azaltmaktır. Ayrıca deneylerde $C$ parametresinin artış ve azalış formulasyonu için doğrusal ve karesel yöntemler denenmiş, gerçekleşen artışın çok hızlı olduğu ve algoritmanın çözüm bulmada zorlandığı görülmüştür. Bu gözlem üzerine, daha yavaş artış ve azalışlara sahip olan (5.12)'de görülen kareköklü formulasyonlar seçilmiştir.

# 6. HESAPSAL SONUÇLAR

Çalışma kapsamında önerilen ve literatürde bulunan yöntemler, Python [18] programlama dili kullanılarak implemente edilmiş, optimizasyon problemlerinin çözümünde Gurobi [19] optimizasyon paketi kullanılmıştır. DVM yöntemi özelleşmiş algoritmalar gerektirdiğinden, bir Python kütüphanesi olan scikit-learn [6] kullanılmıştır. Scikit-learn ayrıca model seçimi ve tahmin değerlerinin hesaplanmasında da kullanılmıştır. GÖPDVM yönteminin genelleştirilmiş özdeğer probleminin çözümünde ise Scipy [20] kullanılmıştır. Yöntemlerin karşılaştırılması, "UCI repository of ML" [21] veri tabanından erişilebilen ve literatürde kullanılan veri kümeleri ile yapılan testler üzerinden yapılmıştır. Kullanılan veri kümelerine dair bilgiler aşağıdaki gibidir.

**BUPA-Liver:** Bu veri kümesinde her biri, bir bekar erkeğin karaciğer bozuklukları ile alakalı kan testlerinden ve günlük tüketilen alkollü içecek sayısından oluşan 345 adet veri bulunmaktadır.

**WBCD-Wisconsin göğüs kanseri tanısı veri kümesi:** Wisconsin Hastanesi'nden, kanserli hastalara dair elde edilen, 569 adet örnek içeren bu kümede, hastalıklar iyi huylu ve kötü huylu olarak ayrılmışlardır.

**WBCP-Wisconsin göğüs kanseri tedavi veri kümesi:** Wisconsin Hastanesi'nde, tedavi gören kanserli hastalara dair toplanan verilerde, tedaviye yanıt veren ve vermeyen hastalar ayrılmıştır.

**Ionosphere veri kümesi:** 351 örnek bulunan bu veri kümesinde, Goose Bay sisteminin topladığı radar sinyalleri iyi ve kötü olarak sınıflandırılmıştır. İyi olan durumlar iyonosferde bulunan bazı yapı tiplerinin kanıtıdır.

**Heart- Kalp hastalığı veri kümesi:** Hastaların 13 farklı özelliği kullanılarak oluşturulan bu veri kümesi, 270 örnekten oluşmakta ve her bir örnek hasta ya da sağlıklı olarak işaretlenmiş bulunmaktadır.

**Pima Diabetes:** En az 21 yaşında olan kadın hastaların 8 farklı özelliğinden oluşan bu veri kümesi, 768 örnek içermekte ve her bir örnek yapılan testler sonucunda hasta ya da değil olarak işaretlenmiş durumdadır.

Çalışmanın devamında kolaylık olması açısından, DVM yaklaşımında karar fonksiyonu olarak denklem (5.3)'ün kullanılması, konik sınıflandırıcı yüzey, ÇKFDVM(a), denklem (5.4)'ün kullanılması, sınıflandırıcı hiperdüzlem, ise

ÇKFDVM(b) olarak kısaltılmıştır. En iyi koni yaklaşımı GÖPÇKF olarak kısaltılırken, ceza parametreli yaklaşım ise *c*-ÇKF olarak kısaltılmıştır.

DVM ve ÇKFDVM yöntemleri için gerekli olan ceza parametresi, $[10^{-7}, 10^{7}]$ aralığındaki değerler, katmanlı 10-kere çapraz doğrulama yöntemi ile taranarak elde edilmiştir. Sunulan sonuçlar ise belirlenen en iyi parametre ile 10 kere tekrarlanan katmanlı-10-kere çapraz doğrulama kullanılarak elde edilen değerlerin ortalamalarıdır. Bir başka deyişle, 100 deneyden elde edilen ortalama sonuçlar, yüzde değerlerini ifade edecek şekilde sunulmuştur.

Tablo 6.1'de, DVM, ÇKFDVM(a) ve ÇKFDVM(b) yöntemlerinden elde edilen eğitim ve test başarıları görülmektedir. Sonuçlar incelendiğinde, ÇKFDVM yönteminin, DVM yöntemi ile aynı ya da daha iyi sonuçlar verdiği görülmektedir. Ayrıca, sınıflandırıcı olarak konik yüzeyin-ÇKFDVM(a), sınıflandırıcı hiperdüzlemden-ÇKFDVM(b) daha başarılı olduğu görülmektedir. Elde edilen bulgulara dayanarak, ÇKF'lerin DVM yöntemi ile beraber çalışabildiği söylenebilir. Ayrıca eğitim ve test başarılırını birbirlerine yakın olması aşırı uyum sorunu olmadığını göstermektedir.

**Tablo 6.1***: Yüzde olarak test ve eğitim başarıları*

| | **DVM** | | **ÇKFDVM(a)** | | **ÇKFDVM(b)** | |
|---|---|---|---|---|---|---|
| **Veri Kümeleri** | **Eğitim B.** | **Test B.** | **Eğitim B.** | **Test B.** | **Eğitim B.** | **Test B.** |
| **Liver** | 70 | $69 \pm 5$ | 71 | $\mathbf{71 \pm 5}$ | 70 | $68 \pm 6$ |
| **WBCD** | 97 | $\mathbf{97 \pm 2}$ | 97 | $\mathbf{97 \pm 2}$ | 97 | $97 \pm 3$ |
| **WBCP** | 80 | $79 \pm 11$ | 82 | $\mathbf{81 \pm 11}$ | 80 | $78 \pm 5$ |
| **Ionosphere** | 93 | $89 \pm 4$ | 95 | $\mathbf{92 \pm 4}$ | 97 | $92 \pm 5$ |
| **Heart** | 84 | $83 \pm 6$ | 84 | $82 \pm 6$ | 84 | $\mathbf{83 \pm 5}$ |
| **Pima Diabetes** | 71 | $\mathbf{71 \pm 4}$ | 71 | $\mathbf{71 \pm 4}$ | 71 | $71 \pm 6$ |

Deneyler yapılırken karşılaşılan bir durum, ÇKFDVM(a) ve ÇKFDVM(b) yöntemlerinin seçilen merkez $c$ değerlerine göre farklı sonuçlar vermesidir. Yapılan deneylerde üç farklı yöntem denenerek en iyi sonuç veren yöntemler seçilmiştir. Bu yöntemler sırasıyla, eğitim kümesinin merkezini kullanmak, eğitim kümesinden sadece $A$ kümesine ait verilerin merkezini kullanmak ve eğitim kümesinden sadece $B$ kümesine ait verilerin merkezini kullanmaktır.

Önerilen bir diğer yöntem GÖPDVM'de karar fonksiyonlarına göre ikiye ayrılmış ve sırasıyla GÖPDVM(a) ve GÖPDVM(b) olarak isimlendirilmiştir. Yöntemin başarılı

sonuçlar vermesi için belirlenmesi gerekli olan Tikhonov terimi $[10^{-7}, 10^{7}]$ aralığı taranarak belirlenmiştir. Bir önceki deneyde olduğu gibi burada da, en iyi parametre ile yapılan 100 adet deneyin ortalama değerleri sunulmuştur. Deney sonuçları değerlendirildiğinde, en iyi koni analojisinin anlamlı olduğu ve GÖPDVM yönteminin başarısının aşıldığı görülmektedir. Bu yaklaşımın ÇKFDVM yaklaşımına karşı bir avantajı da merkez seçme sorununun olmamasıdır.

**Tablo 6.2:** *Yüzde olarak test ve eğitim başarıları*

| Veri Kümeleri | GÖPDVM | GÖPÇKF(a) | GÖPÇKF(b) |
|---|---|---|---|
| **Liver** | 63,8 | 63±3,2 | **64±5,4** |
| **WBCD** | 96,1 | 97±1,4 | **97,8±1,7** |
| **WBCP** | 62,7 | **79±5,0** | 75,2±3,4 |
| **Ionosphere** | 75,19 | 73±1,0 | **88,4±4,7** |
| **Heart** | 81,8 | 83±4,4 | **83±4,7** |
| **Pima Diabetes** | 73,6 | **75±4,1** | 74±4,6 |

Tablo 6.4'de *c*-ÇKF algoritmasının, eğitim ve test başarılarını göstermektedir. Bulunan değerler incelendiğinde yöntemin beklenildiği gibi aşırı uyum sorununu azalttığı söylenebilir. Ayrıca elde edilen test başarıları ÇKF algoritmasından daha yüksektir. Yöntem için gerekli olan ceza parametresi $[10^{-3}, 10^{3}]$ aralığı taranarak, çizilen koni sayısının, test başarısını artırmadığı noktadaki değerler en iyi ceza parametresi olarak seçilmiştir. Diğer deneylerde olduğu gibi, en iyi parametre ile yapılan 100 deneyin ortalama değerleri sunulmuştur.

**Tablo 6.3:***Yüzde olarak test ve eğitim başarıları*

| | **ÇKF Algoritması** | | ***c*-ÇKF Algoritması** | |
|---|---|---|---|---|
| **Veri Kümeleri** | **Eğitim B.** | **Test B.** | **Eğitim B.** | **Test B.** |
| **Liver** | 100 | 54 ± 5,5 | 79,0 | **62 ± 7,0** |
| **WBCD** | 100 | **97 ± 2,3** | 98,7 | 96 ± 2,8 |
| **WBCP** | 100 | 63 ± 1,0 | 76,3 | **76 ± 1,7** |
| **Ionosphere** | 100 | 85 ± 7,1 | 96,0 | **89 ± 5,3** |
| **Heart** | 100 | 75 ± 7,7 | 90,0 | **80 ± 7,2** |
| **Pima Diabetes** | 100 | 69 ± 3,4 | 72,5 | 69 ± 3,4 |

Dikkatli bir araştırmacı, ÇKF algoritmasının ortaya konulduğu makelede sunulan değerlerin ile bu çalışamada sunulan değerlerden farklı olduğunu görecektir. Bunun

sebebi ilgili çalışmada bulunan en iyi değerler kullanılırken, bu çalışmada ifade edildiği gibi ortalama değerlerinin kullanılmasıdır.

Tablo 6.5'de ise, yöntemlerin çalışma sürelerini ve bu sürede çizilen koni sayısını göstermektedir. Sonuçlar incelendiğinde eğitim süresinin ve koni sayısının azaldığı görülmektedir. Bulgular, ÇKF algoritmasının yüksek dereceli bir polinom olarak görülmesi düşüncesini destekler niteliktedir. WBCD veri kümesi hariç, yapılan bütün deneylerde *c*-ÇKF algoritması daha az sayıda koni yardımı ile, ÇKF algoritmasına göre daha yüksek sonuçlar elde edilmiştir.

**Tablo 6.4** *Eğitim süreleri(sn) ve ortlama koni sayısı*

| | **ÇKF Algoritması** | | **c-ÇKF Algoritması** | |
|---|---|---|---|---|
| **Veri Kümeleri** | **Eğitim Süresi** | **ÇKF sayısı** | **Eğitim Süresi** | **ÇKF sayısı** |
| **Liver** | 9,4 | 146,6 | **5,94** | **85,68** |
| **WBCD** | 1,4 | 15,32 | 1,61 | 13,3 |
| **WBCP** | 7,1 | 62,24 | **0,23** | **1** |
| **Ionosphere** | 7,7 | 34,13 | **5,40** | **23,75** |
| **Heart** | 3,5 | 46,67 | **2,11** | **24,15** |
| **Pima Diabetes** | 33,3 | 226,11 | **5,23** | **36,67** |

Önerilen ceza parametleri yaklaşımın etkin bir yöntem olduğu açıkca görülmektedir. WPCP veri kümesi kullanılarak yapılan deneylerde, ÇKF algoritmasında yaklaşık olarak 62 koni çizilerek, yani 62 adet doğrusal programlamam problemi çözülerek, %63'lük test başarısı elde edilirken, *c*-ÇKF algoritmasında sadece 1 koni çizilerek %76'lık bir test başarısına ulaşılmıştır. Buna paralel olarak eğitim süresi de 7,1 saniyeden 0,23 saniyeye düşmüştür.

Bir diğer göze çarpan sonuç ise, Pima Diabetes veri kümesi üzerinde yapılan deneylerden elde edilmiştir. Burada, %69 değerindeki aynı test başarısı, ÇKF algorimatsı ile 33 saniye süren eğitim aşamasında 226 adet koni çizilerek elde edilirken, c-ÇKF algoritmasında yaklaşık olarak 5 saniye süren eğitim ile 37 adet koni çizilerek elde edilmiştir.

# 7. DEĞERLENDİRME VE ÖNERİLER

Yapılan çalışma kapsamında çok yüzlü konik fonksiyonlar, DVM yönteminden elde edilen en büyük marj değerine sahip ayırıcı hiperdüzlem olarak tariflenerek ÇKFDVM, GÖPÇKF yöntemleri geliştirilmiştir. Yapılan deneylerden, marj en büyüklenmesinin konik yüzeylere uygulanabileceğini ve bu doğrultuda yapılan araştırmaların devamının anlamlı olacağını gösterir sonuçlar elde edilmiştir.

İleriki çalışmalarda, ÇKFDVM yaklaşımında ortaya çıkan merkez seçme sorununu gidermeye yönelik araştırmalarım yapılması, yöntemin uygulanabilirliğinin sağlanması açısından anlamlı bir çalışma konusudur. Ayrıca, destek vektörleri kullanılarak her bir veri sınıfı için birer koni oluşturulması ve bu sayede uzaklık değerine dayanan konik sınıflandırıcı elde edilmesi, önerilen yöntemin geliştirilmesi amacıyla yapılabilecek bir diğer çalışmadır.

GÖPÇKF yönteminde yapılan en iyi koni analojisinin anlamlı olduğu test sonuçlarınca desteklenmiş olup, yöntemin GÖPDVM yönteminden daha iyi sonuçlar verdiği görülmektedir. Yöntemin merkez seçme sorunu bulunmaması, ÇKFDVM yöntemine karşı bir avantaj olarak ifade edilebilir. Elde edilen başarılı test sonuçların nedenlerine yönelik bir analizinin yapılması ve literatürdeki diğer özdeğer problemi temelli destek vektör makineleri yöntemlerinin incelenmesi, ileriki çalışmalarda ele alınabilecek olan potansiyel konulardır. Bunlara ek olarak, genelleştirilmiş öz değer problemi kullanılarak, konik fonksiyonlar doğrusal programlama problemi çözdürülmeden elde edilebilirmiştir. Sonraki çalışmalarda, ÇKF algoritmasında (3.4) denklemi ile ifade edilen problemin, genelleştirilmiş özdeğer problemi olarak tariflenerek

algoritmanın ticari optimizasyon paketlerine olan bağımlılığının ortadan kaldırılması amaçlanmaktadır.

Önerilen *c*-ÇKF algoritmasının beklenildiği gibi genelleştirme başarısı görece yüksek olduğu test sonuçları ile gösterilmiştir. Yöntemde manuel olarak belirlenen en iyi ceza parametresinin, literatürde bulunan yöntemler kullanılarak otomatik olarak bulunmasını amaçlayan ve farklı karar kuralları denenerek tahminleme başarısını amaçlayan çalışmalar yapılması planlanmaktadır.

## KAYNAKÇA


[1] Cortes, C. and Vapnik, V. (1995). Support vector networks, *Machine Learning*, 20,173-297.

[2] Gasimov, R. and Öztürk, G. (2006). Separation via Polyhedral Conic Functions. *Optimization Methods and Software*. 21(4).

[3] Öztürk, G. (2007). Sınıflandırma problemleri için yeni bir matematiksel programlama yaklaşımı. Doktora Tezi. Eskişehir: Osmangazi Üniversitesi.

[4] Ozturk, G. and Ciftci, M. T. (2015) Clustering Based Polyhedral Conic Functions Algorithm in Classification. *Journal Of Industrial And Management Optimization*, 11(3), 921-932.

[5] Mangasarian, O. L. and Wild, E. W. (2006) Multisurface proximal support vector machine classification via generalized eigenvalues, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1), 69-74.

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V.,Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011). Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*,12,2825-2830

[7] Bennet, K. P., Mangasarian, O. L. (1992) Robust linear programming discrimination of two linearly inseparable sets. *Optimization methods and software*, 1(1), 23-34.

[8] Astorino, A. and Gaudioso, M. (2002). Polyhedral separability through successive lp. *Journal of Optimization Theory and Applications*, 112(2), 265-293.

[9] Bagirov, A. M. (2005). Max-min separability. *Optimization Methods and Software*, 20(2-3), 271-290.

[10] Çimen, E. (2013). Çok Yüzlü Konik Fonksiyonlar Temelli Sınıflandırma Yaklaşımları ile Hareket Tanıma. Yüksek Lisans Tezi. Eskişehir: Anadolu Üniversitesi.

[11] Cevikalp, H. and Triggs, B. (2017). Visual Object Detection Using Cascades of Binary and One-Class Classifiers. *International Journal of Computer Vision*,1-16.

[12] MacQueen, J.B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 281-297.

[13] Burges, C. JC. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2), 121-167.

[14] Fung, G. and Mangasarian, O. L. (2001). Proximal support vector machine classifiers. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 76-78.

[15] He Yan, A., Qiaolin, B., Ying'an Liu C., Tian'an, Z. D. (2016). The GEPSVM Classifier Based on L1-Norm Distance Metric. *Chinese Conference on Pattern Recognition*,703-719.

[16] Guarracino, M. R., Cifarelli, C., Seref, O., Pardolas, P.M. (2005). A classification method based on generalized eigenvalue problems. *Optimisation Methods and Software*, 22(1),73-81.

[17] Guarracino, M. R., Irpino, A. Verde, R. (2010). Multiclass generalized eigenvalue proximal support vector machines. *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference*,25-32.

[18] Rossum, G. (1995). *Python tutorial, Technical Report CS-R9526.* Amsterdam: Centrum voor Wiskunde en Informatica (CWI).

[19] Gurobi Optimization Inc. (2010). *Gurobi Optimizer Reference Manual Version 3.0.* Houston, Texas: Gurobi Optimization.

[20] Jones, E., Oliphant, E., Peterson, P., et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/ [Online; accessed 2017-04-01].

[21] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[22] https://www.youtube.com/watch?v=_PwhiWxHK8o

(Erişim Tarihi: 01.01.2017)

[23] https://www.youtube.com/watch?v=XUj5JbQihlU

(Erişim Tarihi: 01.01.2017)

[24] https://www.youtube.com/watch?v=EQWr3GGCdzw

(Erişim Tarihi: 01.01.2017)