EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(DOKTORA TEZİ)

VERİ SINIFLANDIRMASI İÇİN MATEMATİKSEL OPTİMİZASYON TABANLI METOTLAR

Nur Uylaş SATI

Tez Danışmanı: Doç. Dr. Burak Ordin

Matematik Anabilim Dalı

Bilim Dalı Kodu: 619.03.03
Sunuş Tarihi: 12.07.2013

Bornova-İZMİR

2013

Nur Uylaş SATI tarafından **DOKTORA** tezi olarak sunulan "**Veri Sınıflandırması için Matematiksel Optimizasyon Tabanlı Metotlar**" başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve **12.07.2013** tarihinde yapılan tez savunma sınavında aday oybirliği/~~oyçokluğu~~ ile başarılı bulunmuştur.

Jüri Üyeleri:		**İmza**
Jüri Başkanı	: Doç.Dr.Ruşk ORDİN	
Raportör Üye	: Yard.Doç.Dr. Arif GÜRSOY	
Üye	: Prof.Dr. Urfat Nuriyev	
Üye	: Prof.Dr.Efendi Nasibov	
Üye	: Prof. Dr. Refail Kasımbeyli	

ÖZET

VERİ SINIFLANDIRMASI İÇİN MATEMATİKSEL OPTİMİZASYON TABANLI METOTLAR

SATI, Uylaş Nur

Doktora Tezi, Matematik Anabilim Dalı
Tez Danışmanı: Doç. Dr. Burak ORDİN
Temmuz 2013, 100 sayfa

Veri madenciliği yöntemlerinden biri olan sınıflandırma, herhangi bir veri kümesinden seçilen eğitim kümesini kullanarak belirli bir tanıma sisteminin yani sınıflandırıcının oluşturulmasıdır. Sınıflandırma mühendislikte, tıpta, ekonomide vb. birçok alanda önemli uygulamalara sahiptir ve bu sebeple birçok farklı disiplinden araştırmacı bu alanda çalışmakta ve daha etkin sınıflandırma modelleri bulmak için araştırmalar yapmaktadır.

Bu doktora tezinde, sınıflandırma problemi ve türleri ifade edilip, sınıflandırmada önemli bir yaklaşım olan çokyüzlü konik fonksiyonlar incelenmiştir. Daha anlaşılır olması açısından çalışma, örnekler ve grafiklerle desteklenmiştir. İkili ve çoklu sınıflandırmada, sonlu n-boyutlu kümelerin çokyüzlü konik fonksiyonlarla etkin bir şekilde ayrımı için, bir diğer veri madenciliği yöntemi olan kümelemeden yararlanılmıştır ve çokyüzlü konik fonksiyonlar temeline dayanan iteratif doğrusal problemlerin çözümüyle kümeleme yöntemini içeren algoritmalar önerilmiştir. Bunun yanı sıra ikili sınıflandırma yöntemlerinden biri olan destek vektör makinalarında çokyüzlü konik fonksiyonlar ve kümelemenin kullanıldığı yeni algoritmalar sunulmuştur. Önerilen yöntemler MATLAB programı kullanılarak programlanıp, gerçek hayat ve sentetik veri kümeleri üzerinde hesaplama denemeleri yapılmıştır. Önerilen yöntemler temel Çokyüzlü Konik Fonksiyonlar algoritması ile karşılaştırılmış ve sonuçlar tablolarla sunulmuştur.

Anahtar kelimeler: Sınıflandırma, Çokyüzlü Konik Fonksiyonlar, Kümeleme, Matematiksel Programlama, Veri Madenciliği.

ABSTRACT

METHODS BASED ON MATHEMATICAL OPTIMIZATION FOR DATA CLASSIFICATION

SATI, Uylaş Nur

Supervisor: Assoc. Prof. Burak ORDİN

July 2013, 100 pages

Classification, a method of data mining, is creation of a specific recognition system that uses a training set selected from a specific dataset. Classification has important applications in many areas as engineering, medicine, economy etc. that's why researchers from many different disciplines work on this area and search to find more effective classification methods.

In this Phd. thesis, classification problems and types are expressed and polyhedral conic functions that's an important approachment of classification is examined. To be more comprehensible the examination is supported by examples and graphics. In binary and multi classification for more effective separation of finite *n*-dimensional sets, clustering, another method of data mining, is used and algorithms based on polyhedral conic functions that includes clustering and iterative linear problem solutions, are suggested. Besides new algorithms that uses polyhedral conic functions and clustering in support vector machines method for binary classification are presented. The proposed methods programmed in MATLAB and calculation experiments have been made on real world and sentetic data sets. Proposed methods are compared with basis poyhedral conic functions algorithm and the results are presented in tables.

Key words: Classification, Polyhedral Conic Functions, Clustering, Mathematical Programming, Data Mining.

TEŞEKKÜR

Doktora çalışmam boyunca ve bu tezi hazırlama sürecinde benden bilgisini ve anlayışını hiçbir zaman esirgemeyen, her türlü sorunumda göstermiş olduğu anlayış ve sabır için değerli hocam Doç. Dr. Burak ORDİN'e sonsuz teşekkür ederim. Ayrıca, bu süreçte benden yardımlarını esirgemeyen Prof. Dr. Refail KASIMBEYLİ ve Prof. Dr. Urfat NURİYEV'e teşekkür etmeyi kendime bir borç bilirim.

Hayatım boyunca bana her türlü destek olan, beni her zaman anlayışla karşılayan aileme, eşim Erman SATI'ya ve tüm sevdiklerime çok teşekkür ederim.

Son olarak, beni maddi açıdan destekleyen Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na (TÜBİTAK) teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER (devam)

Sayfa

İÇİNDEKİLER (devam)

Sayfa

ŞEKİLLER DİZİNİ

ŞEKİLLER DİZİNİ (devam)

ÇİZELGELER DİZİNİ

1. GİRİŞ

Sınıflandırma (Classification) problemi veri madenciliğinin önemli yöntemlerinden biridir. Sınıflandırmada amaç, sınıfları bilinen verileri iki grupta ele alıp (eğitim ve test kümeleri), eğitim kümesini kullanarak yeni kurallar oluşturmak ve sonrasında bu kurallar yardımıyla test kümesi üzerinde kuralların yararlılığını test etmektir. Makine öğrenme alanında gözetimli öğrenme olarakta adlandırılan veri sınıflandırması; tıp, ekonomi, bankacılık, mühendislik vb. bir çok alanda önemli uygulamalara sahiptir. Örneğin tıp alanında hastalara kan vb. testler yardımıyla hastalık tahmini yapılmasında, mühendislik alanında optik karakter tanıma veya günümüzde kullanılan fotoğraf makinalarının birçoğunda bulunan yüz tanıma özelliğinde, bankacılıkta kredi uygulamaları ve sahtekarları belirlemede sınıflandırma yöntemlerinden faydalanılmaktadır. Günümüzde bunlara benzer çeşitli problemlerle karşılaşmak mümkündür bu nedenle farklı disiplinden birçok araştırmacı bu alanda çalışmalar yapmaktadır.

Veri sınıflandırması için veri madenciliği, makine öğrenme, istatistik, yapay sinir ağları, genetik algoritmalar, kaba ve bulanık küme, k en yakın komşu gibi farklı yaklaşımları temel alan çeşitli yöntemler mevcuttur. Son yıllarda bu yöntemlerin yanısıra optimizasyon ve matematiksel programlamayı temel alan çeşitli algoritmalar sunulmuştur (Astorino and Gaudioso, 2002; Bagirov, 2005; Bagirov et. al, 2002; Bennett and Mangasarian, 1992; Bradley and Mangasarian, 2000; Gasimov and Öztürk, 2006). Matematiksel programlama ilk olarak 1980 li yılların başlarında doğrusal ayırma analizinde kullanılarak tanımlanmıştır. O tarihten bu yana matematiksel programlama tabanlı birçok çalışma literatürde yer almaya başlamıştır (Erengüç and Koehler, 1990).

Veri sınıflandırma problemlerinin farklı türleri mevcuttur. Bu türlerden birisi olan ikili sınıflandırma problemlerinde amaç $\mathbb{R}^n$ de tanımlı iki farklı noktalar kümesini ayıran en uygun yüzeyi bulmaktır. Bir diğer türü olan çoklu sınıflandırmada ise amaç yine $\mathbb{R}^n$ de tanımlı ancak ikiden fazla "farklı" noktalar kümesini ayıran en uygun yüzeyi bulmaktır.

Rosen J. B. 1963 yılında, bir ikili sınıflandırma problemi olan örüntü ayırma problemini formüle etmiş ve matematiksel programlamayı temel alarak dışbükey programlama problemi, başka bir deyişle doğrusal kısıtlara göre dışbükey fonksiyonun minimizasyonu, şeklinde ele alarak çözümlemiştir (Rosen, 1963). Mangasarian'ın 1965 yılında yayınlanan makalesinde de benzer bir düşünceyle ayrım için bir düzlem veya doğrusal olmayan bir yüzey elde edilmiştir. Öyle ki iki kümeden birinin elemanları düzlemin veya yüzeyin bir tarafında kalırken diğer küme elemanları öteki tarafta kalmaktadır (Mangasarian, 1965). Böyle bir hiperdüzlemin bulunması için Bennett ve Mangasarian 1992 yılında bir teknik sunmuşlardır (Bennett and Mangasarian, 1992). Sonraki yıllarda benzer yaklaşımı temel alan algoritmalar Astorino, Gaudiso, Bagirov ve arkadaşları tarafından geliştirilmiştir (Astorino and Gaudioso, 2002; Bagirov, 2005; Bagirov et. al., 2011).

1978 yılında Liitschwager ve Wang karmaşık tamsayılı programlama problemi şeklinde formüle edilmiş bir sınıflandırma problemi sunmuşlardır (Liitschwager and Wang, 1978). Çözüm, beklenen toplam yanlış sınıflandırma değerini enküçükleyerek, parametrik olmayan sınıflandırma sağlamaktadır. Ayrıca iki sınıflı (yani iki kümeli) durumlar için özel bir sayım algoritması geliştirilmiştir. Benzer bir teori Vapnik tarafından Vapnik-Chervonenkis istatistiksel öğrenme teorisinde uygulanmıştır. Vapnik-Chervonenkis istatistiksel öğrenme teorisi marjin ayrım kavramıyla birleştirilip destek vektör makinaları tanımlanmıştır (Vapnik, 1995). Orjinal hali Vapnik ve arkadaşları tarafından geliştirilen destek vektör makinalarında amaç, çok boyutlu özellikler uzayında sayısal olarak en 'iyi' hiperdüzlemi bulmaktır, burada "iyi" ile anlatılmak istenen genelleştirmenin eniyi olması durumudur (Cristianini and Taylor, 2000; Shölkopf et al., 1999).

Yapay sinir ağları temelli bir başka yaklaşım Zhang tarafından 2000 yılında sunulmuş ve bu çalışma içerisinde sınıflandırma alanında yapay sinir ağlarıyla, literatürde yer alan temel çalışmalar ele alınmıştır (Zhang, 2000).

Bagirov, Rubinov ve Yearwood 2001 ve 2002 yıllarında hazırladıkları çalışmalarda, sınıflandırma problemleri küresel optimizasyon problemlerine

indirgenmiştir. Problemin çözümü için yerel arama ve kesen açılar yönteminin kombinasyonunu temel alan bir yöntem önerilmiştir. Ayrıca bu metot ikili sınıflandırma dışında çoklu sınıflandırma problemlerine de uyarlanmıştır (Bagirov et al., 2001, 2002).

2002 yılında Astorino ve Gaudioso *n* boyutlu uzayda iki sonlu noktalar kümesi *A* ve *B* nin ayrımı için dışbükey çokyüzlü oluşturan *h* adet hiperdüzlem kullanmıştır. *A* be *B* boştan farklı iki küme olmak üzere, eğer *A* ve *B* dışbükey kabuklarının kesişimleri boş küme ise iki kümenin kesin olarak ayrılabildiği gösterilmiştir ve çokyüzlü ayrılabilir olarak adlandırılmıştır. Ayrıca çokyüzlü ayrımın sağlanamadığı durumlar için ne dışbükey ne de içbükey olan parçalı doğrusal bir hata fonksiyonu ve doğrusal programlama altproblemlerinin iteratif çözümlerini temel alan bir yaklaşım tanımlanmıştır (Astorino and Gaudioso, 2002). Aynı yazarlar 2005 yılında sınıflandırma problemleri için elips şeklinde ayrımı tanımlamışlardır (Astorino and Gaudioso, 2005).

Bagirov 2005 yılında sınıflandırma problemini, sonlu sayıda hiperdüzlemin oluşturduğu parçalı doğrusal fonksiyon ile çözen enb-enk (max-min) ayrım metodunu tanımlamıştır. Burada bir hata fonksiyonu oluşturulup, minimizasyonu için bir algoritma önerilmiştir. İfade edilen metot Astorino ve Gaudioso'nun tanımladığı *h* çokyüzlü ayrımın bir genelleştirmesi olarak ele alınabilir. Eğer *A* ve *B* kümelerinin kesişimleri boş küme ise iki kümenin kesin olarak doğrusal fonksiyonların enb-enk metodu ile ayrılabilir olduğu gösterilmiştir (Bagirov, 2005).

Bagirov, Ugon, Webb ve Karasözen 2011 yılında genel olarak hiperdüzlemlerin eğitimi(training) için artımlı(incremental) bir yaklaşım kullanan algoritma sunmuşlardır. Bu yaklaşım genel sınıflandırma hata fonksiyonunun en yakın genel küçükleyenini bulmayı ve kümelerin ayrımı için ihtiyaç duyulan en az sayıdaki hiperdüzlemin hesaplanmasını sağlamıştır (Bagirov A. M. et al., 2011).

Gasimov ve Öztürk'ün 2006 yılında yayınlanan makalesinde *n*-boyutlu uzayda iki sonlu *A* ve *B* noktalar kümesinin ayrım problemi çokyüzlü konik fonksiyonların özel bir tipi kullanılarak çözümlenmiştir. Ayrım fonksiyonunu

bulmak için doğrusal programlama altproblemlerinin iteratif çözümlerini temel alan verimli, sonlu bir algoritma önerilmiştir. Herbir iterasyonda, grafiği çokyüzlü koni olan bir fonksiyon yapılandırılmış ve son ayırıcı fonksiyon ise bu yapılandırılan fonksiyonların noktasal en küçüğü olarak tanımlanmıştır (Gasimov and Öztürk, 2006).

Bagirov, Ugon, Webb, Öztürk ve Kasımbeyli 2011 yılında sınıflar arası parçalı doğrusal sınırların bulunması için yeni bir algoritma önermişlerdir. Bu algoritma iki ana bölümden oluşmuştur. İlk bölümde sınırın üzerinde veya sınıra yakın veri noktalarının belirlenmesi için Gasimov ve Öztürk tarafından tanımlanan çokyüzlü konik küme kullanılmış ve ikinci bölümde ise sadece belirlenen noktalar kullanılarak Bagirov tarafından tanımlanan parçalı doğrusal sınırlar hesaplanmış yani enb-enk ayrım uygulanmıştır. Parçalı doğrusal sınırlar bir hiperdüzlemden başlayarak artımlı şekilde hesaplanmıştır. Bu yaklaşım ile büyük veri kümeleri ve çoklu veri sınıflandırma problem çözümlerinde verimlilik sağlanmıştır.

Bu tezde, çokyüzlü konik fonksiyonların kullanıldığı bir matematiksel programlama yaklaşımı ele alınmış ve bu yaklaşım içerisinde iki veya ikiden fazla sonlu *n*-boyutlu kümenin etkin bir şekilde ayrımı için yine bir veri madenciliği yöntemi olan kümelemeden yararlanılmıştır. Kümelemede amaç, bir veri kümesini benzer veriler aynı kümede bulunacak şekilde bölümlere ayırmaktır. Önerilen algoritmada, kümeleme, çokyüzlü konik fonksiyonların tepe noktalarını bulma aşamasında kullanılacaktır. Aynı zamanda algoritmada yer alan doğrusal problemin kısıtlarında yapılan değişikliklerle eğitim ve test kümeleri başarım değerleri arasındaki farkı azaltmak amaçlanmıştır. Çalışmalar doğrultusunda çokyüzlü konik fonksiyonlar temeline dayanan iteratif doğrusal problemlerin çözümünü ve kümeleme yöntemini içeren algoritmalar önerilmişitr. Önerilen yöntem MATLAB programı kullanılarak programlanmış ve *(http://archieve.ics.uci.edu/ml/)*' den alınan gerçek hayat veri kümeleri ve *(http://www.datasetgenerator.com)'* da oluşturulan sentetik veri kümeleri üzerinde hesaplama denemeleri yapılmıştır. Önerilen algoritma Çokyüzlü Konik Fonksiyonlar algoritması ile kıyaslanıp elde edilen sonuçlar tablolarla ifade edilmiştir.

Bu tez, girişi izleyen 9 bölümden oluşmaktadır.

İkinci bölümde veri madenciliği konusu çalışılmış ve uygulama alanları incelenmiştir.

Üçüncü bölümde; veri sınıflandırma problemleri ele alınıp türleri belirtilmiş ve problem içerisinde yer alan kavramlar örneklerle tanıtılmıştır. Veri sınıflandırma problemleri için kullanılan yöntemler incelenmiştir.

Dördüncü bölümde; ikili sınıflandırma yöntemlerinden biri olan ve tezde kullanılacak çokyüzlü konik fonksiyonlarla ayrım konusu detaylı şekilde çalışılmıştır.

Beşinci bölümde; veri madenciliği yöntemlerinden biri olan veri kümeleme yöntemi tanıtılmış, tezde kullanılacak olan *k*-ortalamalar ve *k*-medyan algoritmaları verilmiştir.

Altıncı bölümde ise çokyüzlü konik fonksiyonlarla ikili sınıflandırma algoritmasına kümeleme yöntemi eklenerek algoritmanın verimliliği arttırılmaya çalışılmıştır ve daha anlaşılır olması açısından bir örnek üzerinde görseller kullanılarak açıklanmıştır. Bunun yanı sıra çokyüzlü konik fonksiyonlarla ayrım algoritmasında yer alan doğrusal programlama probleminde de değişiklikler yapılmıştır. Son olarak elde edilen algoritma veri kümeleri üzerinde denenmiş ve ÇKF (çokyüzlü konik fonksiyon) algoritması ile karşılaştırılarak sonuçlar tablolarla sunulmuştur.

Yedinci bölümde; çoklu sınıflandırma konusu incelenmiş ve bir önceki bölümde yapılan yenilikler çoklu sınıflandırma problemlerine de uygulanmıştır. Yine örnekler ve görseller kullanılarak açıklamalar yapılmış ve elde edilen çoklu sınıflandırma algoritması veri kümeleri üzerinde denenerek sonuçlar tablolarda sunulmuştur.

Sekizinci bölümde; sınıflandırma yöntemlerinden biri olan destek vektör makinalarında, çokyüzlü konik fonksiyonlar kullanılarak yeni bir yaklaşım önerilmiştir. Konu ile ilgili örnekler görsel grafiklerle desteklenerek sunulmuş ve

geliştirilen algoritma veri kümeleri üzerinde test edilmiştir. Ayrıca sınıflandırma verimini ve çalışma zamanını pozitif anlamda geliştirmek için kümeleme tabanlı bir algoritma önerilmiştir. Geliştirilen bu algoritma veri kümeleri üzerinde çalıştırılarak sonuçlar tablolarda sunulmuştur.

Dokuzuncu bölüm sonuç bölümüdür. Bu bölümde yapılan tüm çalışmalar kısaca özetlenmiş ve elde edilen genel sonuçlar sunulmuştur.

2. VERİ MADENCİLİĞİ VE UYGULAMA ALANLARI

Bu bölümde veri madenciliği kavramı incelenip, veri madenciliği içerisinde yer alan temel kavramlar ele alınacaktır. Bunun yanısıra veri madenciliğinin uygulama alanlarından bahsedilecektir.

2.1. Veri Madenciliğine genel bir bakış

Veri madenciliği büyük veri yığınlarını analiz ederek içerisinden önceden bilinmeyen ilişki ve eğilimlerin bulunmasını sağlayan, verileri kullanıcılar için yararlı hale dönüştüren yöntemler topluluğudur.

Farklı kaynaklarda veri madenciliği süreci, veritabanlarından bilgi çıkarımı (knowledge discovery in databases) olarakta adlandırılmakta ve veri madenciliği bu sürecin belli bir adımı olarak düşünülmektedir. Veri madenciliği veriden bir model çıkartmak için belli bir algoritmanın uygulanması adımı iken, veritabanında bilgi çıkarımı sürecindeki diğer adımlar: veri seçimi, veri temizleme, öncel uygun bilgileri özümseme ve sonuçların uygun şekilde yorumlanması adımlarıdır (Bradley et al.,1998).

Aşağıda veri madenciliği ile ilgili literatürde yer alan çeşitli tanımlar verilmiştir.

- Jacobs, veri madenciliğini, ham datanın tek başına sunamadığı bilgiyi çıkaran veri analizi süreci olarak tanımlamıştır (Jacobs, 1999).

- David, veri madenciliğinin büyük hacimli verilerdeki örüntüleri araştıran matematiksel algoritmaları kullandığını söylemiştir. David'e göre veri madenciliği hipotezleri keşfeder, sonuçları birleştirmek için insan yeteneğini kullanır. Veri madenciliğinin sadece bir bilim olmadığı, aynı zamanda bir sanat olduğu da söylenebilir (David, 1999).

- Hand, veri madenciliğini istatistik, veritabanı teknolojisi, örüntü tanıma, makine öğrenme ile etkileşimli yeni bir disiplin ve geniş veritabanlarında önceden tahmin edilemeyen ilişkilerin ikincil analizi olarak tanımlamıştır (Hand, 1998).

- Kitler ve Wang, veri madenciliğini, oldukça tahminci anahtar değişkenlerin binlerce potansiyel değişkenden izole edilmesini sağlama yeteneği olarak tanımlamışlardır (Kitler and Wang, 1998).

Veri madenciliği modelin kurulması aşamasında farklılıklar göstermekte ve gözetimli (supervised) ve gözetimsiz (unsupervised) öğrenme olarak ikiye ayrılmaktadır. Örnekten öğrenme olarak da isimlendirilen gözetimli öğrenimde, bir denetçi tarafından ilgili sınıflar önceden belirlenen bir kritere göre ayrılarak, her sınıf için çeşitli örnekler verilir. Sistemin amacı verilen örneklerden hareket ederek her bir sınıfa ilişkin özelliklerin bulunmasıdır. Gözetimsiz öğrenmede ise ilgili örneklerin gözlenmesi ve bu örneklerin özellikleri arasındaki benzerliklerden hareket ederek sınıfların tanımlanması amaçlanmaktadır (Frank and Witten, 2000).

Veri madenciliğinin uygulamada iki temel öncelikli amacı *tanımlama* ve *tahminleme'*dir. Tahminleme veritabanındaki bazı bilgi alanlarını veya değişkenleri kullanarak gelecekte karşılaşılabilecek diğer ilgili değişkenleri veya bilinmeyenleri tahmin etmektir. Tanımlama ise veriyi anlatan yorumlanabilir modelleri tanımlamaktır (Fayyad U. et al., 1996). Açıkça görülmektedir ki tahminleme ve tanımlama arasında keskin bir ayrım yapmak güçtür. Özetle tahminleme gelecekle ilgili tahminlerde bulunma, tanımlama ise veriden bilgi edinme işidir (Alpaydın, 2005). Her ikisi içinde amaçlara ulaşmak için çeşitli veri madenciliği yöntemleri kullanılmaktadır. Aşağıda bu yöntemler kısaca açıklanmıştır:

Birliktelik Analizi: Birliktelik analizi, bir veri kümesindeki kayıtlar arasındaki bağlantıları arayan gözetimsiz (unsupervised) veri madenciliği biçimidir. Birliktelik analizi çoğu zaman perakende sektöründe süpermarket müşterilerinin satın alma davranışlarını ortaya koymak için kullanıldığından "pazar sepeti analizi" olarak da adlandırılır (Bland, 2002).

Kümeleme: Kümeleme, verideki benzer kayıtların gruplandırılmasını sağlayan bir gözetimsiz (unsupervised) veri madenciliği biçimidir. Kümelemede amaç, bir veri kümesini, benzer veriler aynı kümede bulunacak şekilde bölümlere ayırmaktır. Kümelemede, genellikle k-ortalamalar algoritması ya da Kohonen şebekesi gibi istatistiksel yöntemler kullanılmaktadır. Hangi yöntem kullanılırsa kullanılsın süreç aynı şekilde işler. Her kayıt var olan kümelerle karşılaştırılır. Bir kayıt kendisine en yakın kümeye atanır ve bu kümeyi tanımlayan değeri değiştirir. Optimum çözüm bulununcaya kadar kayıtlar yeniden atanır ve küme merkezleri ayarlanır (Hui ve Jha, 2000).

Sınıflandırma: Veri madenciliğinde sınıflandırma, verilen örneklerden hareket ederek herbir sınıfa ilişkin özellikleri bulan ve bu özelliklerin kural cümleleri ile ifade edilmesini sağlayan bir yaklaşımdır. Sınıflandırmada veri kümeleri önceden tanımlanmış bir sınıfa göre etiketlenmiştir yani gözetimli (supervised) öğrenme mevcuttur. Bu veri kümeleri eğitim ve test kümeleri olmak üzere ikiye ayrılır. Eğitim veri kümesi kullanılarak test veri kümesindeki verilerin hangi sınıfa ait olduğu bulunur yani sınıflandırmada görev sınıflandırılmamış verileri sınıflandırmak üzere uygulanabilen bir model oluşturmaktır (Berry and Linoff, 1997).

2.2 Veri Madenciliği Uygulama Alanları

Veri madenciliği fen, mühendislik, tıp, bankacılık, borsa, eğitim, ticari, pazarlama ve telekominikasyon vb. alanlarda çeşitli uygulamalara sahiptir. Aşağıda bu alanlardan fen ve mühendislik, tıp ve pazarlama alanlarında çeşitli veri madenciliği örneklerine yer verilmiştir.

Pazarlama alanında en öncelikli ihtiyaçlardan biri farklı müşteri gruplarını belirleyebilen ve ileriki davranışlarını tahmin edebilen sistemlerdir. Business Week (Berry, 1994) tüm perakendecilerin yarıdan fazlasının bu sistemleri kullanıyor olduklarını veya satın almayı planladıklarını, ayrıca bu sistemleri kullananların geleceğe dönük iyi sonuçlarla karşılaştıklarını belirlemiştir. Başka bir pazarlama uygulaması da sepet analiz sistemleridir (Agrawal, 1996). Bu sistemlerde ise "Eğer bir müşteri X ürününü alıyorsa Y ve Z ürününü almaya

meyillidir" şeklinde analizler yapar ve bu analizler perakendeciler için çok yararlı bilgileri ortaya çıkarır (Fayyad et al., 1996). Bunların yanı sıra raf düzenlemesi ve promosyonlarda da veri madenciliği yöntemlerinden faydalanılır.

Tıp alanında hastane bilgi sistemlerinden veya diğer tıbbi veri toplayan sistemlerden alınan veriler üzerinde yapılan veri madenciliği çalışmaları, hem uzmanlar için hem hastane yönetimi için hem de hastaların daha kaliteli bir hizmet almalarında etkin rol alabilir. Tıp alanında veri madenciliği uygulamaları çeşitli konularda yapılmıştır. Örnek vermek gerekirse A. Kusiak ve arkadaşları tarafından akciğerdeki tümörün iyi huylu olup olmadığına dair, karar destek amaçlı bir çalışma yapılmıştır. İstatistiklere göre Amerika'da 160.000 den fazla akciğer kanseri vakasının olduğu ve bunların %90'ının öldüğü belirlenmiştir. Bu bağlamda bu tümörün erken ve doğru olarak teşhisi önem kazanmaktadır. Noninvaziv testler ile elde edilen bilgi sayesinde %40-60 oranında doğru teşhis konabilmektedir. İnsanlar kanser olup olmadıklarından emin olmak için biyopsi yaptırmayı tercih etmektedirler. Biyopsi gibi invaziv testler hem maliyeti yüksek hem çeşitli riskler taşımaktadır. Faklı yerlerde ve farklı zamanlarda kliniklerde toplanan invaziv test verileri arasında yapılan veri madenciliği çalışmaları teşhiste %100 oranında doğruluk sağlamıştır (Kusiak et al., 2000).

Fen ve Mühendislik alanında verilebilecek en belirgin uygulama optik karakter, yüz ve konuşma tanıma vb. uygulamalardan oluşan örüntü tanıma uygulamalarıdır. Optik karakter tanıması, tanımlamak istediğimiz karakterler kadar çok sınıfın olduğu bir uygulama örneğidir. Özellikle karakterlerin el yazısı olduğu durumda ise daha ilgi çekicidir. Yüz tanıma örnek olayında ise girdi bir görüntü, sınıflar da tanınacak insanlardır. Öğrenme programıda, yüz görüntülerini kimliklere benzetmeyi öğrenir. Konuşma tanıma uygulamalarında, girdiler sesle ilgilidir sınıflar da söylenebilecek kelimelerdir. Ele alınan konuşma tanıma probleminde akustik bir sinyal ile herhangi bir dildeki kelime arasındaki ilişki öğrenilmelidir (Öztürk, 2007).

3. VERİ SINIFLANDIRMA PROBLEMLERİ

Bu bölümde veri sınıflandırma probleminde yer alan matematiksel kavramlar ifade edilip, problemin daha iyi anlaşılması için örnekler verilmiştir. Ayrıca veri sınıflandırma problemlerinin çözümü için geliştirilmiş matematiksel programlama temelli yöntemler incelenmiştir.

3.1 Veri Sınıflandırma Problemlerinin Matematiksel Kavramları

Veri sınıflandırma problemlerinde amaç eğitim kümelerini kullanarak yeni bir model oluşturmak ve sonrasında bu model yardımıyla yeni verilerin belli bir doğruluk değeriyle sınıflandırılmasını sağlamaktır.

Veri sınıflandırma problemlerinde veri kümeleri gözlemler (örnekler) topluluğundan oluşur ve herbir gözlem $a = (a_1, ..., a_n) \in \mathbb{R}^n$ şeklinde bir vektör ile temsil edilir. $\mathbb{R}^n$ de tanımlı bu vektör koordinatları, gözlemin özellikleri (attributes) olarak adlandırılır.

Herbir $a = (a_1, ..., a_n) \in IR^n$ vektörü, *a*'nın normu olarak, negatif olmayan sayılarla aşağıdaki şekilde ifade edilir:

$$\|a\|_k = (\sum_{i=1}^{n} |a_i|^k)^{1/k}, k \geq 1.$$

Tezde genellikle $\|.\|_2$ normu kullanılacaktır. Veri kümesinden iki adet *a* ve *b* noktaları alındığında bu iki nokta (gözlem) arası uzaklık *d(a,b)*, $\|.\|_2$ normu kullanıldığında aşağıdaki şekilde hesaplanır:

$$d(a,b) = \|a - b\|_2$$

Bir veri kümesinin *m* adet gözlemden oluştuğunu düşünelim bu durumda veri kümesi $a^i = \left(a_1^i, ..., a_n^i\right), i = 1, ..., m$ noktalar topluluğu veya veri matrisi ile temsil edilir.

Bu veri matrisi *m* adet aynı cinsten gözlemler hakkında bilgileri içerir. Matrisin satırları herbir gözlemi temsil ederken, sütunları özellikleri temsil eder. Farklı veri matrislerinde farklı cinslerden gözlemler hakkında bilgiler yer alabilir. Örneğin bir veri matrisinde gözlemler insanlar iken bir diğer veri matrisinde ele alınan gözlemler hayvanlar, kitaplar, arabalar vb. cinslerden olabilir.

Herbir gözlem *n* farklı özellik (nitelik) ile tanımlanır. Örneğin bir insanın özellikleri cinsiyeti, yaşı, ismi, medeni durumu, doğum yeri vb. olabilir. Bu özellikleri numerik gösterimi aşağıdaki şekilde yapılabilir:

Cinsiyet özelliği iki farklı yanıta sahiptir. Bayanlar için numerik olarak "0", erkekler, için ise "1" kullanılır veya isteğe göre bu numerik değerler "1,2" olarak değiştirilebilir.

Medeni durumu dört farklı yanıta (bekar, evli, dul, boşanmış) sahiptir. Burada herbir durum için 4 farklı rakam kullanılır."0,1,2,3" veya "1,2,3,4" gibi.

Bu tür sayısal gösterimlerin kullanıldığı özelliklere, nominal özellikler denir. Eğer anlamlı bir düzen doğrultusunda sayısal değerler veriliyorsa ordinal özellik adını alır. Örneğin"iyi, çok iyi, kötü, çok kötü" gibi yanıtlar varsa.

Veri sınıflandırma problemleri kendi içinde ikiye ayrılır. Bunlardan biri sınıfları bilinen gözlemlerin kullanıldığı gözetimli (supervised) sınıflandırma problemleri diğeri ise sınıfları bilinmeyen gözlemlerin kullanıldığı gözetimsiz (unsupervised) sınıflandırma problemleridir.

Gözetimli sınıflandırma problemlerinde kullanılan veri matrislerinde özellikler sütununa ek olarak bir de sınıf sütunu yer almaktadır. Genelde veri matrislerinde gözlemlerin sınıfları ilk veya son sütunda belirtilmiştir ve sınıflarda özelliklere benzer olarak sayısal değerler yer almaktadır.

Gözetimli sınıflandırma problemleri de kendi içinde çoklu (multi) ve ikili (binary) sınıflandırma olarak ikiye ayrılır. İkili sınıflandırmada sınıf sayısı iki iken, çoklu sınıflandırmada sınıf sayısı ikiden fazladır. Örneğin kanser teşhislerinde, özellikler hastanın taşıdığı belirtiler iken, sınıflar kanserli hastalar

“0” ve kanser olmayan hastalar “1” şeklinde iki tanedir ve ikili sınıflandırma yapılır. Omurgalı hayvanları özelliklerine göre sınıflandırmak istediğimizde ise sınıf sayısı, balıklar “0”, kurbağalar “1”, sürüngenler “2”, kuşlar “3”, memeliler “4” olmak üzere beştir ve bu bir çoklu sınıflandırma problemidir.

Tezde ele alacağımız problemler gözetimli ikili ve çoklu sınıflandırma problemleridir.

İkili sınıflandırma problemlerinde sınıflar $C=\{C_1,C_2\}$ ile gösterilir ve genelde bu iki sınıf negatif ve pozitif sınıf olarak adlandırılır. Bir ikili sınıflandırma yöntemi eğitim verilerini kullanarak $f: X \to \{C_1,C_2\}$ formunda parametrize edilmiş bir fonksiyon oluşturur:

$$f(x)=\begin{cases} C_1 & \text{eğer } g(x)<0 \\ C_2 & \text{eğer } g(x)\geq 0 \end{cases}$$

Burada $g: x \to \mathbb{R}$, x in reel sayılar kümesine bir izdüşümüdür ve ayırıcı fonksiyon olarak tanımlanır.

Örnek 3.1: Çizelge 3.1’de konunun anlaşılması için basit bir gözetimli ikili sınıflandırma problem örneğinin eğitim veri matrisi sunulmuştur. Bu problemde verilen gözlemler; insanlar, özellikler; cinsiyet, yaş, medeni durum ve maaştır. Sınıflar ise evi olanlar ve evi olmayanlar şeklinde belirtilmiştir.

Cinsiyet: bayan “0”, erkek “1”

Yaş: yaşın numerik değeri

Medeni durum: evli “0”, bekar “1”

Maaş: almıyor “0”, alıyor “1”

Çizelge 3.1’de yer alan eğitim veri matrisi, bir sınıflandırma yönteminde kullanılarak elde edilen kurallarla, cinsiyeti, yaşı, medeni durumu ve maaş

durumu belli olan kişilerin, evinin olup olmadığı sorusuna belli bir doğruluk değeriyle yanıt verilebilir.

Çizelge 3.1: Eğitim veri matrisi

cinsiyet	yaş	medeni durum	maaş	evi var/yok
0	20	1	0	0
1	25	0	1	1
0	30	0	0	0
0	40	1	1	1
1	18	1	0	1
0	60	0	1	1
1	35	1	1	0
0	65	0	0	1
1	50	0	1	0
1	45	0	0	1

3.2 Veri Sınıflandırma Problemlerinin Çözüm Yöntemleri

Veri sınıflandırma problemlerinin çözümlerinde en sık kullanılan ve literatürde en çok karşılaşılan yöntemler bayes sınıflandırma, ayırma analizi, karar ağaçları, sinir ağları ve destek vektör makinalarıdır (Han and Kamber, 2001; Alpaydın, 2010). Bunların yanı sıra genetik algoritmalar, k-en yakın komşu, kaba ve bulanık küme yaklaşımları da kullanılmaktadır (Han and Kamber, 2001). Bu

yöntemler veri madenciliğinde sınıflandırma yöntemleri olduğu gibi aynı zamanda makine öğrenme alanında gözetimli öğrenme yöntemleri olarakta anılır.

Bu yöntemlerin yanısıra veri sınıflandırma problemlerinin çözümleri için literatürde matematiksel programlama tabanlı yöntemlere de çokça rastlanmaktadır. Bu yöntem detaylı olarak 3.3 bölümünde incelenecektir.

3.2.1 Bayes sınıflandırma

Sınıflandırmada kullanılan veriler tamamıyla bilinmeyen bir süreç sonrası ortaya çıkabilir. Bu bilgi eksikliği sürecin rastlantısal bir süreç olarak modellenmesiyle ifade edilebilir. Belki süreç gerçekten belirsiz değildir ancak sürecin tümü hakkında bilgiye ulaşılamadığı zamanlarda süreç rastlantısal olarak modellenir ve analiz edilmesi için olasılık teorisi kullanılır (Alpaydın, 2010).

Bayes sınıflandırma, olasılıkta bayes teorisinin kullanıldığı bir istatistiksel sınıflandırma yöntemidir ve amaç verilen örneklerden yararlanarak belirli bir sınıfa ait olma olasılığının kestirimini yapmaktır. Bayes teoremi aşağıdaki şekilde açıklanabilir.

"x" ler gözlem yani veri, $j=1,2,\ldots,n$ için "C_j" sınıfları temsil etsin.

- $p(x|C_j)$: Sınıf j den bir örneğin x olma olasılığı
- $p(C_j)$: Sınıf j'nin ilk olasılığı
- $p(x)$: Herhangi bir örneğin x olma olasılığı
- $p(C_j|x)$: x olan bir örneğin sınıf j den olma olasılığı (aranan cevap)

Bu verilenler doğrultusunda bayes teoremi aşağıdaki şekilde uygulanır:

$$P(C_j|x) = \frac{p(x|C_j)P(C_j)}{p(x)} = \frac{p(x|C_j)P(C_j)}{\sum_k p(x|C_k)P(C_k)}$$

Ve sonuç olarak "*x*" verisi,

C_j 'ye atanır öyle ki $P(C_j|x) = \max_k P(C_k|x)$.

3.2.2 Ayırma analizi

Ayırma analizi ilk kez Fisher tarafından 1930'lu yıllarda çalışılmıştır (Han and Kamber, 2001). Ayırma analizinin iki temel görevi vardır (Öztürk, 2010):

1) Grupları birbirinden ayırmayı sağlayan fonksiyonları bulmak,

2) Hesaplanan fonksiyonlar aracılığı ile yeni bir veriyi sınıflama hatası en küçük olacak şekilde sınıflardan herhangi birine atamaktır.

Ayırıcı fonksiyon için Φ_i parametreler kümesi ile ifade edilen $g_i(x|\Phi_i)$ modeli tanımlanır. Burada öğrenme, ayırımın kalitesini arttıran, yani verilen etiketli (sınıfları bilinen) eğitim kümesi üzerinde sınıflandırma doğruluk değerini enbüyükleyen, model parametrelerinin optimizasyonudur.

Ayırıcı tabanlı yaklaşımlarda bayes sınıflandırmada olduğu gibi sınıflarda doğru yoğunluk tahminlerine ihtiyaç duyulmaz bunun yerine sınıflar arası doğru sınırların tahmini yapılmaya çalışılır. Ayırıcı tabanlı yaklaşımı savunanlar, örn. Vapnik, sınıf yoğunluklarını tahminleme probleminin, ayırıcı sınırları tahminleme probleminden daha zor olduğunu belirtirler. Ancak bu ayırıcının basit bir fonksiyonla tanımlanabildiği durumlar için geçerlidir. En basit durum, ayırıcı fonksiyonun doğrusal olarak tanımlanabildiği durumlardır (Alpaydın, 2010).

Aşağıda ayırma analizinde doğrusal ve lojistik ayırma konuları incelenmiştir.

3.2.2.1 Doğrusal ayırma analizi

Doğrusal ayrım, modelinin basitliği nedeniyle çok tercih edilir. Burada basitlikten kastedilen zaman ve yer karmaşıklığının *O(d)* olmasıdır. Doğrusal ayırma analizi ile bulunan ayırıcı fonksiyonlar aşağıdaki şekilde ifade edilir:

$$g_i\left(x|w_i, w_{i0}\right) = w_i^T x + w_{i0} = \sum_{j=1}^{d} w_{ij} x_j + w_{i0}$$

Doğrusal bir modeli anlamak ve analiz etmek kolaydır. Fonksiyon sonucunda elde edilen değer, x_j, $j=1,...,n$, gözlem özelliklerinin ağırlıklı toplamıdır. w_j değeri *j.* özelliğin ağırlık değeridir ve büyüklüğü, x_j özelliğinin önemini belirtir, işareti ise probleme etkisinin pozitif veya negatif yönlü olduğunu gösterir. Örneğin bir müşteri kredi başvurusunda bulunduğunda banka başvuru yapan kişinin kredi skorunu belirlerken çeşitli niteliklerinin etkileri toplamını ele alır; bu özelliklerden yıllık geliri sonuca pozitif etki ederken, giderleri negatif olarak etki eder (Alpaydın, 2010).

Ayırıcı *g* fonksiyonu, *w* ağırlık vektörü, w_0 başlangıç (eşik) değeri ile bir hiperdüzlemi tanımlar. Bu hiperdüzlem girdi uzayını iki yarı-uzaya böler: B_1 karar bölgesi C_1 sınıfını, B_2 karar bölgesi C_2 sınıfını temsil eder. B_1 bölgesindeki *x* vektörü hiperdüzlemin pozitif tarafında, B_2 bölgesindeki negatif tarafında yer alır.

Doğrusal ayırıcı fonksiyonu geometrik olarak açıklamak gerekirse, *w* vektörü hiperdüzlem üzerinde yer alan herhangi bir vektörün normalidir:

$$g\left(x_1\right), g\left(x_2\right) = 0 \text{ ise } w^T\left(x_1 - x_2\right) = 0$$

ve *x* vektörü aşağıdaki şekilde ifade edilebilir (Duda et al., 2001):

$$x = x_p + r\frac{w}{\|w\|}$$

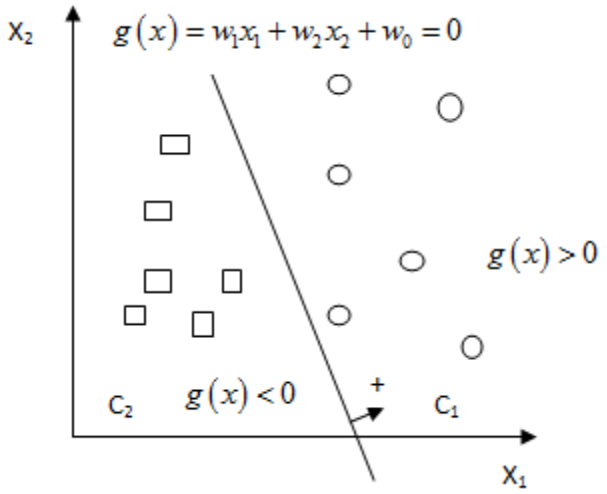
Burada x_p , x noktasının hiperdüzlem üzerine normal izdüşümünü, r ise x ile hiperdüzlem arasındaki uzaklığı belirtir.

$$r = \frac{g(x)}{\|w\|}$$

Tüm bu hesaplamalar sonrasında orijine olan uzaklığın $r_0 = \frac{w_0}{\|w\|}$ olduğu görülür.

Dolayısıyla w_0 orjine göre hiperdüzlemin konumunu, w ise oryantasyonunu (yönünü) belirler (Alpaydın, 2010).

$x \in \mathbb{R}^2$ olduğu durumda doğrusal ayırıcı iki sınıfı ayıran bir düzlemdir ve Şekil 3.1'de gösterilmiştir:



Şekil 3.1. İki boyutlu uzayda doğrusal ayırıcı

Doğrusal ayırma analizi birçok araştırmacı tarafından çalışılmıştır (Mangasarian, 1965; Glover, 1990; Smith, 1968; Bennett and Mangasarian, 1992). Bu çalışmaların genelindeki temel amaç A ve B kümeleri dışbükey kabuk kesişimleri boşküme olan durumda kesin ayırıcı bir düzlem elde etmektir. Ancak gerçek hayat veri kümelerinde kesişimin boşküme olmadığı durumlarda mevcuttur

bu durumlar içinde hatalı noktaların minimizasyonunu içeren doğrusal ayırım yöntemleri sunulmuştur (Bennett and Mangasarian, 1992).

Doğrusal programlama tekniklerinden yararlanılan matematiksel programlama temelli doğrusal ayırım konusu 3.3.1 bölümünde detaylı şekilde açıklanacaktır.

3.2.2.2 Lojistik ayırma analizi

Lojistik regresyon analizinin kullanım amacı istatistikte kullanılan diğer model yapılandırma teknikleri ile aynıdır. En az değişkeni kullanarak en iyi uyuma sahip olacak şekilde bağımlı-bağımsız değişkenler arasındaki ilişkiyi tanımlayabilen bir model kurmaktır (Coşkun ve Kartal, 2004).

Lojistik ayrımda, sınıf koşullu yoğunluklar, $p(x|C_i)$ değil, oranlar modellenir. Ayırma analizlerinde, çok değişkenli normal dağılım varsayımı ön koşul olarak kabul edilmesine rağmen regresyon bakış açısına dayanan lojistik ayırma analizinde ise herhangi bir varsayım ileri sürmeksizin sınıflandırma yapabilmek mümkündür.

İki sınıflı sınıflandırma problemleri için lojistik ayırma analizi en çok kullanılan yöntemlerden birisidir.

İki sınıflı bir problemde verilen bir x noktasının C_1 ve C_2 sınıfında olma olasılıkları aşağıdaki şekilde tanımlanır:

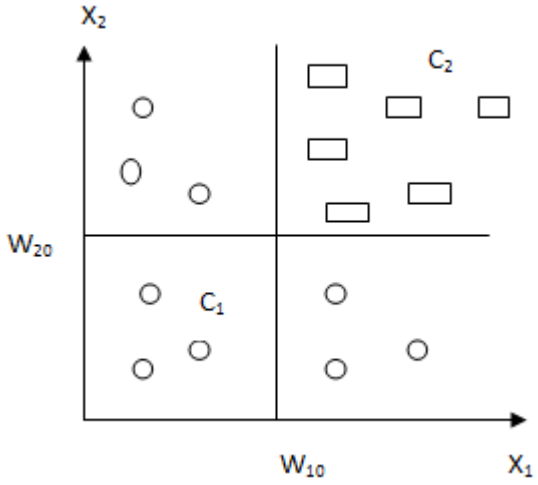
$$p(C_1|x) = \frac{1}{1+\exp(\beta' x)}, \; p(C_2|x) = \frac{\exp(\beta' x)}{1+\exp(\beta' x)}$$

Ve son olarak x noktası, en büyük olasılık değerini aldığı sınıfa atanır (Öztürk, 2010).

3.2.3 Karar Ağaçları

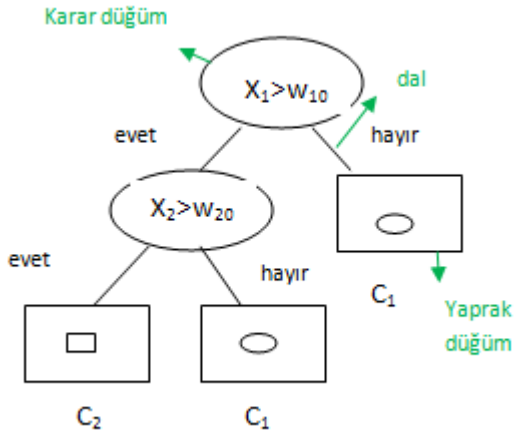
Karar ağacı, yerel bölgelerin daha küçük adımlarla tekrarlı ayrımlar dizisi ile tanımlandığı gözetimli öğrenmede kullanılan hiyerarşik bir modeldir.

Bir karar ağacı iç karar düğümleri, dallar ve uçta bulunan yapraklardan oluşur (Bkz. Şekil 3.3). Herbir *m* karar düğümünde, dalları ayrık çıktılarla etiketleyen $f_m(x)$ fonksiyonu uygulanır. Verilen bir veriye yönelik herbir düğümde bir test uygulanır ve çıktıya göre yeni bir iç düğüm ile devam edilir. Bu süreç kökten başlar ve yaprak içinde yazan değer istenen çıktıyı verene kadar tekrarlı olarak devam eder. Karar ağacı yapılandırması sınıflandırmada yer alan eğitim kümesindeki tüm verileri doğru sınıflandıracak şekilde oluşturulur ve böylece gelecekte verilen verilerin detaylı şekilde sorgulanarak doğru sınıflandırılması amaçlanır.



Şekil 3.2. Eğitim kümesi

Şekil 3.2.'de verilen veri uzayına (eğitim kümesine) yönelik Şekil 3.3'de bir karar ağacı oluşturulmuştur (Alpaydın, 2010).



Şekil 3.3. Eğitim kümesine yönelik karar ağacı

Herbir $f_m(x)$ fonksiyonu *d*-boyutlu veri uzayında bir ayırıcı fonksiyondur öyle ki, bu fonksiyon veri uzayını küçük bölümlere ayırır, sonrasında bir alt kökte yer alan diğer bir $f_m(x)$ ile küçük bölümler kendi içinde yine alt bölümlere parçalanır. $f_m(.)$ fonksiyonu basit bir fonksiyondur ve ağaç üzerinde tüm fonksiyonlar ele alındığında kompleks bir fonksiyonun basit kararlar dizisine parçalandığı görülür. Farklı karar ağaç yöntemleri, $f_m(.)$ için farklı modeller tanımlayabilir ve bu model sınıfları veri uzayının ve ayırıcının şeklini belirler. Herbir yaprak düğümü bir çıktı etiketine sahiptir ve bu etiketler sınıflandırma problemlerinde sınıf kodlarını tanımlar. Bir yaprak düğümü veri uzayında belli bir bölgeyi tanımlar öyle ki bu bölgede yer alan tüm örnekler aynı etikete sahiptir. Bölge sınırları, kökten yaprak düğüme kadar uzanan yolda iç düğümlerde kodlanan ayırıcı fonksiyonlar ile tanımlanır (Alpaydın, 2010).

En iyi karar ağacını yapılandırma problemi bir NP-tam problemdir ve bu nedenle birçok kuramcı tarafından en iyiye en yakın ağacı yapılandırmak için verimli açgözlü yaklaşımlar araştırılmıştır. Murthy 1998' de yaptığı çalışma ile karar ağaçları ile ilgili yapılan çalışmaları özetlemiştir (Kotsiantis, 2007).

Eğitim verilerini en iyi şekilde bölümleyen özelliği sorgulayan fonksiyon, karar ağacında kök düğümde yer almalıdır. Bu önemli özelliği seçmek için de

(Hunt et. al.,1966) bilgi kazanımı, (Breimann et. al.,1984) gini indeksi gibi çeşitli çalışmalar sunulmuştur.

Sınıflandırmada karar ağacı kullanımının avantaj ve dezavantajlarını aşağıdaki şekilde sıralayabiliriz.

Avantajlar:

- Karar ağacı oluşturmak zahmetsizdir.
- Küçük ağaçları yorumlamak kolaydır.
- Anlaşılabilir kurallar oluşturulabilinir.
- Sürekli ve ayrık nitelik değerleri için kullanılabilinir.

Dezavantajlar:

- Sürekli nitelik değerlerini tahmin etmekte çok başarılı değildir.
- Sınıf sayısı fazla ve öğrenme kümesi örnekleri sayısı az olduğunda model oluşturma çok başarılı değildir.
- Zaman ve yer karmaşıklığı öğrenme kümesi örnekleri sayısına, nitelik sayısına ve oluşan ağacın yapısına bağlıdır.
- Hem ağaç oluşturma karmaşıklığı hem de ağaç budama karmaşıklığı fazladır.

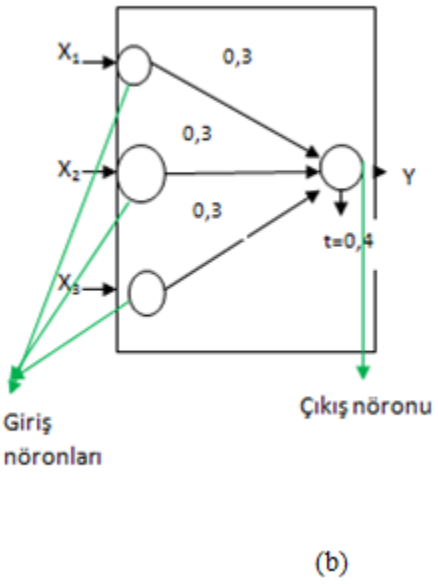
3.2.4 Yapay sinir ağları

Yapay sinir ağları (YSA) insan beynindeki sinir hücrelerinin işlevini modelleyen bir yapıdır ve birbiri ile bağlantılı katmanlardan oluşur. Katmanlar arası iletim vardır ve bu iletim katmanlar arasındaki bağın ağırlığına ve her hücrenin değerine bağlı olarak değişebilir.

Eğitim veri kümesi verilen bir problem ile oluşturulan basit bir yapay sinir ağı örneği aşağıdaki şekilde gösterilmiş ve fonksiyonel olarak ifade edilmiştir.

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

(a)



(b)

Şekil 3.4. a) Eğitim veri matrisi b)Eğitim kümesine yönelik yapay sinir ağı

$$Y = I(0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4) \text{ ise } I(Z) = \begin{cases} 1 \text{ eğer } z > 0 \\ 0 \text{ diğer.} \end{cases}.$$

Yapay sinir ağı ile öğrenme toplamda 4 adımdan oluşur:

- Yapay sinir ağı oluşturma: giriş verisi modellenir, gizli katman sayısı ve gizli katmanlardaki nöron sayısı belirlenir.
- Yapay sinir ağını eğitme
- Sinir ağını küçültme
- Sonucu yorumlama

YSA'da giriş nöron sayısı, öğrenme kümesi verilerinin nitelik sayısı, çıkış nöron sayısı, sınıf sayısı ile belirlenir. Gizli nöron sayısı ise öğrenme sırasında ayarlanır.

YSA avantajları aşağıdaki teorik bakış açılarına dayanır (Zhang, 2000):

1. YSA, veri güdümlü, öz uyarlanan yöntemlerdir öyle ki temeldeki model için dağılımsal veya fonksiyonel bir formun

belirgin bir tanımlaması olmaksızın kendilerini veriye göre ayarlayabilirler.

2. Evrensel fonksiyonel yaklaşımcılardır ve keyfi bir doğrulukla herhangi bir fonksiyona yaklaşabilirler (Cybenko,1989; Hornik, 1991; Hornik et. al.,1989); öyle ki herhangibir sınıflandırma süreci grup üyeleri arası fonksiyonel ilişkileri ve objelerin özelliklerini araştırdığı için bu temel fonksiyonun kesin tanımlanması kuşkusuz çok önemlidir.

3. YSA, doğrusal olmayan modellerdir ve bu özellik onları gerçek hayattaki karmaşık ilişkileri modellemede daha esnek kılar.

4. YSA, ardıl olasılık değerlerini tahminleyebilir ve bu istatistiksel analiz uygulamalarında ve sınıflandırmada kural belirleme çalışmalarında temel sağlar (Richard and Lippmann,1991).

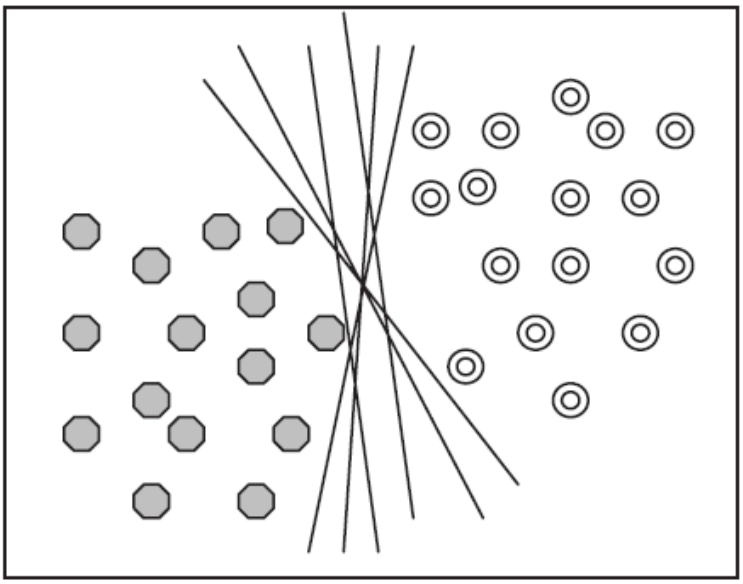
3.2.5 Destek Vektör Makinaları

Sınıflandırma problemlerinde kullanılan yöntemlerden biride Destek vektör makinaları (Support vector machines) yöntemidir. 1990 yılında *Vapnik* ve arkadaşları tarafından tanıtılan bu yöntem, verilen etiketli örnekler yardımıyla, doğrusal ayırıcının parametrelerini bulduran ayırıcı tabanlı (discriminant-based) optimizasyona dayalı bir yaklaşımdır ve birçok araştırmacı tarafından çeşitli alanlarda kullanılmıştır (Mammone et. al., 2009).

3.2.5.1 Verilerin doğrusal olarak ayrılabilme durumu

D veri kümesi $(x_1, y_1)(x_2, y_2)...(x_n, y_n)$ elemanlarından oluşsun. Burada *n* veri kümesinin eleman sayısıdır ve $y \in \{+1, -1\}$ sınıfları temsil eder, *x* ise veri özelliklerini içeren vektördür (Cristianini and Shawe, 2000).

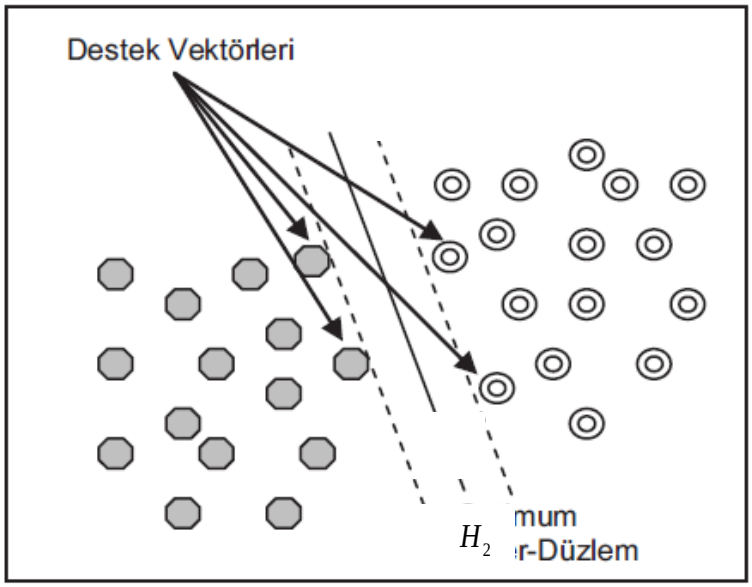
Şekil 3.5 üzerinde görüldüğü gibi veri farklı ve çok sayıda doğru ile ayrılabilmektedir. Çok boyutlu uzayda bu doğruların yerini hiper düzlemler alacaktır.



Şekil 3.5. İki boyutlu uzayda sınıfları ayıran doğrular

Tüm hiperdüzlemler örneklerin hepsini doğru şekilde sınıflandırmaktadır (Bkz. Şekil 3.5) ancak daha iyi genelleştirme için örneklerin hiperdüzlemden belli bir mesafe uzakta olmasıda istenir. Bu uzaklık, marjin olarak adlandırılır ve en büyüklenmeye çalışılır. En iyi ayırıcı düzlem marjini en büyükleyen hiperdüzlemdir. Basit bir tanımla ayrım için veriyi birbirinden ayıran bu doğrulardan ya da hiper düzlemlerden aralarında en geniş mesafeye sahip olanları seçmek en uygun yoldur.

Şekil 3.6 üzerinde yer alan H_2 ve H_1 hiper düzlemleri gözönüne alınsın. Bu iki hiper düzlem arasında bulunan ve en geniş marjine sahip H_0 hiper düzlemi ise iki sınıf veriyi birbirinden ayıran doğrusal hiper düzlemdir. Bu H_0 düzlemine "**optimal ayırma hiper düzlemi"** adı verilir.



Şekil 3.6. Optimum hiper-düzlem

H_0 düzlemi, üzerindeki noktalar cinsinden aşağıdaki şekilde ifade edilir :

$$H_0 : W^T X + b = 0$$

Bu ifade şu şekilde de yazılabilir:

$$\sum_{i=1}^{n} w_i x_i + b = 0$$

Burada W ağırlık vektörünü $W = \{w_1, w_2, ..., w_n\}$, n ise niteliklerin sayısını göstermektedir. İfade içinde yer alan b ise sabit bir sayıyı temsil eder.

H_1 hiper düzlemi şu şekilde ifade edilir:

$$H_1 : W^T X + b = 1$$

Benzer şekilde H_2 hiper düzlemi,

$$H_2 : W^T X + b = -1$$

biçiminde ifade edilir.

Bu eşitsizlikler birleştirilerek tek bir eşitsizlik biçimine dönüştürüldüğünde

$$y_i(W^T X + b) - 1 \geq 0 \quad \forall i$$

ifadesi elde edilir. Burada H_1 ve H_2 hiper düzlemleri göz önüne alındığında, bu düzlemler üzerindeki gözlemler "**destek vektörü**" adını alır.

Burada *d* ve *sınır* değerlerinin hesapları aşağıda verilmiştir ve $\langle .,. \rangle$ vektörel çarpım, $\|.\|$ ise iki-normu temsil eder.

$$d = \frac{|b|}{\|w\|} - \frac{|b+1|}{\|w\|} = \frac{1}{\|w\|}, \qquad sınır = 2d = \frac{2}{\|w\|}.$$

Sınırın yani marjinin enbüyüklenmesi, paydanın enküçüklenmesi anlamına geleceği için bu problem enküçükleme problemi olarak ele alınmıştır ve elde edilen minimizasyon problem modeli aşağıda belirtilmiştir:

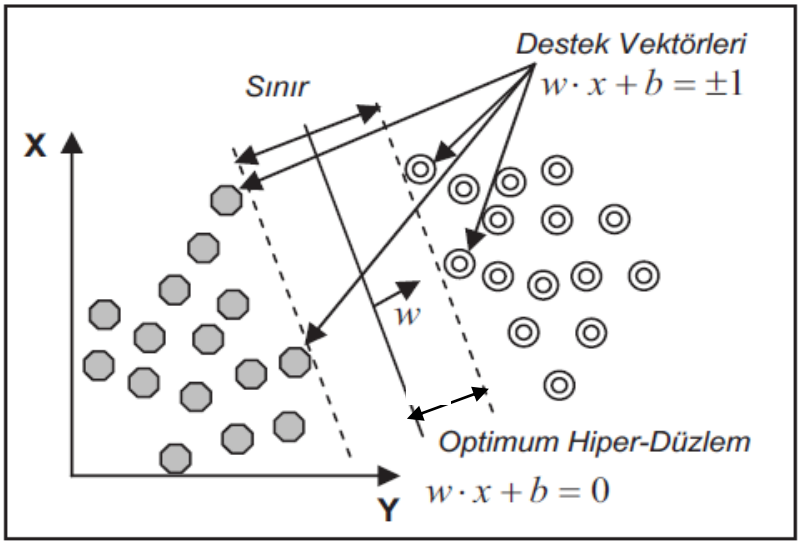
$$\min \frac{\langle w, w \rangle}{2}$$

$$y_i \left(\langle x_i, w \rangle + b \right) - 1 \geq 0. \tag{3.1}$$

Çözüm için problemin Langrange formulasyonu kullanılır:

$$L(w, b, \alpha) = \frac{1}{2} \langle w, w \rangle - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\langle w, x_i \rangle + b \right) - 1 \right]$$

Burada $\alpha_i \geq 0, i = 1, ..., l$, eşitsizlik kısıtlarının (3.1) herbiri için langrange çarpanlarıdır (Mammone et. al., 2009).



Şekil 3.7. Sınır (marjin) gösterimi

3.2.5.2 Verilerin doğrusal olarak ayrılamama durumu

Verilerin doğrusal olarak ayrılamama durumunda negatif olmayan ve hataları ifade eden ξ_i gevşek değişkenlerinin optimizasyon modeline aşağıdaki şekilde eklenmesi sağlanarak soruna çözüm aranmıştır.

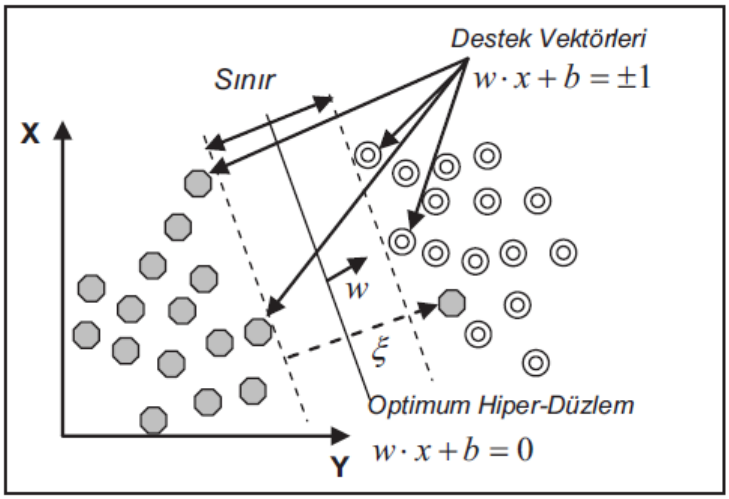
$$w^T x_i + b \geq 1 - \xi_i, \;\; y_i = +1 \; için$$

$$w^T x_i + b \leq -1 + \xi_i, \;\; y_i = -1 \; için$$

veya,

$$y_i (w^T x_i + b) \geq 1 - \xi_i \;\; (i = 1, 2, .., n), \; \xi_i \geq 0$$

Burada $\xi_i > 1$ olan veriler, hiper düzlemin yanlış tarafında kalan, yani doğrusal olarak ayrılmayı önleyen bölgedeki gözlem değerleridir. $0 < \xi_i < 1$ ise hiper düzlemin doğru yanında yer alan , ancak en geniş marjin bölgesi içinde kalan gözlem değerlerini ifade eder (Bennett and Demiriz, 1998).



Şekil 3.8. ξ hata değerinin geometriksel gösterimi

Sonuç olarak elde edilen minimizasyon problem modeli aşağıda belirtilmiştir:

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

$$y_i (w^T x_i + b) \geq 1 - \xi_i \quad (i = 1, 2, .., n), \; \xi_i \geq 0$$

3.2.6 Diğer sınıflandırma yöntemleri

Yukarıda anlatılan sınıflandırma yöntemlerinin yanısıra literatürde genetik algoritmalar, *k*-en yakın komşu, kaba ve bulanık küme yaklaşımlarına rastlanmaktadır (Han and Kamber, 2001).

Genetik Algoritmalar: Genetik algoritmalarda, Darwin tarafından geliştirilen "evrim teorisi" temel alınır. Genelde bulgusal bilginin seyrek ve tamamlanmamış olduğu çok sınıflı veya yüksek boyutlu sınıflandırma problemlerinde tercih edilir. Genetik algoritmalar genel arama yapısına sahip olduğu için daha genel kurallar ortaya çıkartır. Algoritmaların yerel arama yaptığı diğer sınıflandırma yöntemleri ise kural çıkarım paradigması temellidir (Gündoğan et. al., 2004). Algoritma ilk olarak popülasyon adı verilen bir eğitim kümesi ile başlatılır. Bir popülasyondan alınan sonuçlar bir öncekinden daha iyi olacağı beklenen yeni bir popülasyon

oluşturmak için kullanılır. Evrim süreci yani yeni popülasyonlar yaratma iterasyonu tamamlandığında bağımlılık kuralları veya sınıf modelleri ortaya çıkartılmış olur (Shah and Kursak, 2004).

k-en yakın komşu: En yakın komşu yöntemi, bir örneğin sınıfını tahmin eden sınıflandırma algoritmaları arasında belkide en basit olanıdır ancak zor olan kısmı bellek ihtiyacının fazla olmasıdır. Eğitim süreci çok geleneksel ve sıradandır bu süreçte tüm eğitim örnekleri sınıflarıyla birlikte kaydedilir. Yeni bir örneğin tahminini yapmak için öncelikle bu örneğin diğer tüm eğitim örneklerine uzaklığı hesaplanır. Uzaklık hesaplamasında öklid uzaklık metodu kullanılır ve $x, y \in \mathbb{R}^n$ örnekleri arası mesafe aşağıdaki şekilde hesaplanılır:

$$D(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Sonrasında en yakın *k* adet, $k \geq 1$ sabit tamsayı, eğitim örneği kaydedilir ve bu *k* adet örnek arasında en çok rastlanan sınıf aranır. Bu sınıf yeni örneğin sınıf etiketidir (Bhatia, 2010).

Kaba Küme: Kaba küme teorisi 1970 li yıllarda Pawlak tarafından geliştirilmiştir. Nesneler aynı bilgi ile nitelendiriliyorlarsa nesneler aynıdır veya ayırt edilemezdir. Ortaya konulan ayırt edilememe ilişkisi, Kaba Küme kuramının temelini oluşturur. Kaba küme teorisinde bir yaklaştırma uzayı ve bir kümenin alt ve üst yaklaştırmaları vardır. Alt yaklaşım (lower approximation) kesin olarak kavrama ait olan bütün nesnelerden oluşur. Üst yaklaşım (upper approximation) ise kavrama ait olması muhtemel bütün nesneleri içerir. Alt ve üst yaklaşımlar arasındaki fark sınır bölgesini oluşturur. Alt ve üst sınırlar arasında tanımlanan herhangibir nesne ise "kaba küme" olarak adlandırılır (Pawlak and Skowron, 1994; Pawlak, 2002).

Bulanık küme: Bulanık küme teorisi ilk olarak 1965 yılında Zadeh tarafından önerilmiştir (Zadeh, 1965). Bulanık sınıflandırmanın görevi problem için sınıflandırma yapabilecek kurallar üretmektir. Bulanık sınıflandırmada dilsel terimlerle ifade edilen üyelik fonksiyonları kullanılır. Üyelik fonksiyonları çeşitli

şekillerde olabilir ancak en çok tercih edileni üçgensel üyelik fonksiyonlarıdır. Üyelik fonksiyonlarının taban değerleri uzmanların bilgisiyle ya da otomatik olarak genetik algoritma uygulamalarıyla oluşturulur. Bulanık mantıkta özellik değerleri bulanık değerlere dönüştürülür. Verilen yeni bir örnek için birden fazla bulanık kural uygulanabilir. Uygulanabilir olan herbir kural kategorilerdeki üyelik için bir adaylığa neden olur. Tipik olarak tahmin edilen herbir kategorinin gerçeklik değerleri toplanır ve elde edilen toplamlar sistem tarafından döndürülen bir değerle birleştirilir (Nabiyev, 2005).

3.3 Matematiksel Programlama Tabanlı Sınıflandırma Yöntemleri

Bir önceki bölümde açıklanan yöntemler dışında 1950 li yıllardan bu yana araştırma konusu olan ve verimli sonuçlar doğuran matematiksel programlama temelli sınıflandırma yöntemleride mevcuttur. Bu yöntemlerden doğrusal programlama temelli doğrusal ayrım konusu ilk olarak 1965 yılında Mangasarian tarafından incelenmiştir (Mangasarian, 1965). Sonraki yıllarda doğrusal programlama tabanlı olan gürbüz doğrusal ayrım ve ikili doğrusal ayrım yaklaşımları sunulmuştur (Bennett and Mangasarian, 1992, 1993). Bunun yanı sıra yine doğrusal ayrımın kullanıldığı *h*-çokyüzlü ayrım 2002 yılında Astorino ve Gaudioso tarafından çalışılmıştır (Astorino and Gaudioso, 2002). Bagirov ise *h*-çokyüzlü ayrımın bir genellemesi olan enb-enk ayrım yaklaşımını sunmuştur (Bagirov, 2005).

Bunların yanı sıra tezde de kullanılacak olan bir diğer önemli matematiksel programlama temelli yaklaşım Gasimov ve Öztürk tarafından 2006 yılında tanıtılan çokyüzlü konik fonksiyonların kullanıldığı ikili sınıflandırma yaklaşımıdır (Gasimov and Öztürk, 2006). Bölüm 3.4'te bu yöntem detaylı olarak incelenecektir.

Aşağıdaki bölümlerde matematiksel programlama temelli yöntemlerden doğrusal ve ikili ayrım, *h*-çokyüzlü ayrım ve enb-enk ayrım yaklaşımları incelenecektir.

3.3.1 Doğrusal Ayrım

Sırasıyla *A* ve *B*, *n*-boyutlu *m* ve *p* adet vektör içeren kümeler olsun:

$$A = \{a^1, ..., a^m\}, a^i \in IR^n, i = 1, ..., m,$$
$$B = \{b^1, ..., b^p\}, b^j \in IR^n, j = 1, ..., p.$$

A ve *B* kümeleri doğrusal ayrılabilirdir eğer $x \in \mathbb{R}^n, y \in \mathbb{R}^1$ olacak şekilde $\{x, y\}$ hiperdüzlemi mevcutsa öyle ki (Bagirov, 2005);

herhangi $j = 1, ..., m$ için

$$\langle x, a^j \rangle - y < 0,$$

herhangi $k = 1, ..., p$ için

$$\langle x, b^k \rangle - y > 0.$$

Doğrusal ayrımın bir özelliği olarak iki kümenin dışbükey kabukları kesişmez ancak eğer kesişimleri boş küme değil ise yanlış sınıflandırma hata değerlerini minimize eden hiperdüzlem ya da doğrusal olmayan ayrım yüzeyleri araştırılabilir. Bu hiperdüzlemi bulma probleminin matematiksel modeli aşağıdaki gibi verilebilir (Bennett and Mangasarian, 1992):

$$enk\ f(x, y),\ (x, y) \in \mathbb{R}^{n+1}. \qquad (3.2)$$

Burada

$$f(x, y) = \frac{1}{m}\sum_{i=1}^{m} enb\ (0, \langle x, a^i \rangle - y + 1) + \frac{1}{p}\sum_{j=1}^{p} enb\ (0, -\langle x, b^j \rangle + y + 1)$$

bir hata fonksiyonudur ve $\langle .,. \rangle$, $\mathbb{R}^n$ de skaler çarpımı simgeler. (3.2) problemi aşağıdaki doğrusal programa eşdeğerdir:

$$enk \qquad \frac{1}{m}\sum_{i=1}^{m} t_i + \frac{1}{p}\sum_{j=1}^{p} z_j$$

$$\langle x, a^i \rangle - y + 1 < t_i, \quad i = 1,...,m$$

$$-\langle x, b^j \rangle + y + 1 < z_j, \quad j = 1,...,p$$

$$t \geq 0, \;\; z \geq 0 .$$

Burada t_i , negatif olmayan ve $a^i \in A$ noktası için hatayı temsil eden, z_j negatif olmayan ve $b^j \in B$ noktası için hatayı temsil eden sayılardır.

A ve *B* kümeleri ancak ve ancak (3.2) probleminin çözümü olan (x^*, y_*) için $f^* = f(x^*, y_*) = 0$ ise doğrusal ayrılabilirdir. $x = 0$ aşikar çözümünün ortaya çıkmadığı Bennett ve Mangasarian tarafından 1992'de ispatlanmıştır.

3.3.2 İkili doğrusal ayrım

Bennett ve Mangasarian (1993) yayınladıkları "Bilineer separation of two sets in *n*-space" makalesinde ikili doğrusal ayrım konusunu incelemişlerdir. Bu yaklaşımda iki küme iki hiperdüzlem kullanılarak ayrılmıştır. Yine *A* ve *B* kümelerinin sırasıyla *m* ve *p* adet *n*-boyutlu vektör içerdiği varsayılmıştır.

Tanım 3.1. *A* ve *B* kümeleri ikili doğrusal ayrılabilirdir ancak ve ancak $\left(x^1, y_1\right) ve \left(x^2, y_2\right)$ gibi iki hiperdüzlem vardır öyle ki aşağıdaki koşullardan herhangi birisi sağlanır (Bennett and Mangasarian,1993).

1. herhangi $j = 1,...,m$ için

$$\left\langle x^l, a^j \right\rangle - y_l < 0, l = 1,2$$

ve herhangi $k = 1,..., p$ için $l \in \{1,2\}$ vardır öyle ki;

$$\left\langle x^l, b^k \right\rangle - y_l > 0.$$

2. herhangi $k = 1,..., p$ için

$$\left\langle x^l, b^k \right\rangle - y_l < 0, l = 1,2$$

ve herhangi $j = 1,..., m$ için $l \in \{1,2\}$ vardır öyle ki;

$$\left\langle x^l, a^j \right\rangle - y_l > 0.$$

3. herhangi $j = 1,..., m$ için ya

$$\left\langle x^l, a^j \right\rangle - y_l < 0, l = 1,2$$

yada

$$\left\langle -x^l, a^j \right\rangle + y_l < 0, l = 1,2$$

ve herhangi $k = 1,..., p$ için ya

$$\left\langle x^1, b^k \right\rangle - y_1 < 0, \left\langle -x^2, b^k \right\rangle + y_2 < 0$$

yada

$$\left\langle -x^1, b^k \right\rangle + y_1 < 0, \left\langle x^2, b^k \right\rangle - y_2 < 0.$$

Tanım 3.1 enb ve enk ifadeleri ile aşağıdaki gibi yazılabilir.

Tanım 3.2 *A* ve *B* kümeleri ikili doğrusal ayrılabilirdir ancak ve ancak $\left(x^1, y_1\right) ve \left(x^2, y_2\right)$ gibi iki hiperdüzlem vardır öyle ki aşağıdaki koşullardan herhangi birisi sağlanır (Bennett and Mangasarian,1993).

1. herhangi $j = 1,...,m$ için

$$\underset{l=1,2}{enb}\left\{\left\langle x^l, a^j \right\rangle - y_l\right\} < 0,$$

ve herhangi $k = 1,..., p$ için

$$\underset{l=1,2}{enb}\left\{\left\langle x^l, b^k \right\rangle - y_l\right\} > 0.$$

2. herhangi $k = 1,..., p$ için

$$\underset{l=1,2}{enb}\left\{\left\langle x^l, b^k \right\rangle - y_l\right\} < 0,$$

ve herhangi $j = 1,...,m$ için

$$\underset{l=1,2}{enb}\left\{\left\langle x^l, a^j \right\rangle - y_l\right\} > 0.$$

3. herhangi $j = 1,...,m$ için

$$enb\left[enk\left\{\left\langle x^1, a^j \right\rangle - y_1, -\left\langle x^2, a^j \right\rangle + y_2\right\}, enk\left\{-\left\langle x^1, a^j \right\rangle + y_1, \left\langle x^2, a^j \right\rangle - y_2\right\}\right] < 0,$$

ve herhangi $k = 1,..., p$ için

$$enb\left[enk\left\{\left\langle x^1, b^k \right\rangle - y_1, -\left\langle x^2, b^k \right\rangle + y_2\right\}, enk\left\{-\left\langle x^1, b^k \right\rangle + y_1, \left\langle x^2, b^k \right\rangle - y_2\right\}\right] > 0.$$

İkili doğrusal ayrım problemi belli bir ikili doğrusal programlama problemine indirgenmiştir ve "Bennett K.P., Mangasarian O.L. : Bilineer

separation of two sets in n-space" (1993) makalesinde bu problemin çözümü için bir algoritma önermiştir.

3.3.3 Çokyüzlü ayrım

Astorino ve Gaudiso 2002 yılında *h*-çokyüzlü ayrım kavramını tanıtmış ve incelemişlerdir. Bu metotta bir sınıf, çokyüzlü bir fonksiyon ile yakınsanır. Uzayın geri kalan kısmı ise diğer sınıfı yakınsamada kullanılır. Sınıflandırma hata fonksiyonu dışbükey ve düzgün olmayan fonksiyonların toplamı ile temsil edilir.

A ve *B* kümeleri *h*-çokyüzlü ayrılabilirdir ancak ve ancak $x^i \in \mathbb{R}^n,\ y_i \in \mathbb{R}^1,\ i = 1,...,h$ olacak şekilde $\{x^i, y_i\}$ *h*- hiperdüzlem kümesi varsa öyle ki;

1) herhangi $j = 1,...,m$ ve $\mathrm{i} = 1,...,h$ için

$$\langle x^i, a^j \rangle - y_i < 0,$$

2) herhangi $\mathrm{k} = 1,...,p$ için en az bir tane $i \in \{1,...,h\}$ vardır öyle ki

$$\langle x^i, b^k \rangle - y_i > 0.$$

Ayrıca *A* ve *B* kümelerinin, $h \le p$ için ancak ve ancak $coA \cap B = \varnothing$ iken *h*-çokyüzlü ayrılabilir olduğu da Astorino ve Gaudiso tarafından ispatlanmıştır (Astorino and Gaudiso, 2002).

A ve *B* kümelerinin çokyüzlü ayrımı aşağıdaki probleme indirgenmiştir:

$$enk\ \ f(x,y)\ ,\ (x,y) \in \mathbb{R}^{(n+1)\times h} \tag{3.3}$$

Öyle ki;

$$f(x,y)=\frac{1}{m}\sum_{j=1}^{m} enb\left[0, \underset{1\le i\le h}{enb}\left\{\left\langle x^i, a^j\right\rangle - y_i + 1\right\}\right]$$
$$+\frac{1}{p}\sum_{k=1}^{p} enb\left[0, \underset{1\le i\le h}{enk}\left\{-\left\langle x^i, b^k\right\rangle + y_i + 1\right\}\right]$$

Bu fonksiyon dışbükey olmayan parçalı bir doğrusal hata fonksiyonudur. $x^i=0,\ i=1,...,h$, sonucunun eniyi çözüm olamayacağı ispatlanmıştır. $\left\{\bar{x}^i, \bar{y}_i\right\}, i=1,...,h$, (3.3) probleminin genel çözümü olsun. *A* ve *B* kümeleri *h*-çokyüzlü ayrılabilirdir ancak ve ancak $f(\bar{x},\bar{y})=0$ ise. Eğer $\bar{I}\subset\{1,...,h\}$ boş olmayan küme içerisinde herhangi $x^i=0, i\in\bar{I}$ varsa o zaman *A* ve *B* kümeleri $\left(h-\left|\bar{I}\right|\right)$-çokyüzlü ayrılabilirdir.

Astorino ve Gaudiso tarafından 2002 yılında yayınlanan makalede (3.3) probleminin çözümü için bir algoritma geliştirilmiştir. Bu algoritmanın her bir adımındaki iniş yön hesaplaması belli bir doğrusal programlama problemine indirgenmiştir. Bu tekniğin avantajı araştırmayı sadece dışbükey çokyüzlü alana sıkıştırmamasıdır. Böylece *A* ve *B* her iki kümede dışbükey olmak zorunda değildir.

Orsenigo ve Vercellis 2007 yılında yayınladıkları makalede daha az veri ile öğrenme sağlayan, çokyüzlü ayrım algoritmasının bir modifikasyonunu tanıtmış ve tamsayılı destek vektör makinaları kavramına kadar uzanan karmaşık tamsayılı programlama modelini kullanmışlardır (Orsenigo and Vercellis, 2007).

3.3.4 Enb-Enk Ayrım

Birçok uygulamada iki küme ne doğrusal ne ikili doğrusal ne de çokyüzlü ayrılabilirdir. Bu gibi durumlarda iki küme daha karmaşık parçalı doğrusal fonksiyon ile ayrılabilir. Enb-enk ayrım adıyla anılan ve parçalı doğrusal fonksiyonun kullanıldığı bir yöntem 2005 yılında Bagirov tarafından önerilmiştir.

$H=\{h_1,...,h_l\}, h_j=\{x^j, y_j\}, j=1,...,l, x^j\in\mathbb{R}^n, y_j\in\mathbb{R}^1,$ olacak şekilde sonlu hiperdüzlemler kümesi ve $J=\{1,...,l\}$ hiperdüzlem sayısı olsun. Bu kümenin herhangi bir parçası $J^r=\{J_1,...,J_r\}$ dir öyle ki

$$J_k\neq\varnothing, k=1,...,r, J_k\cap J_j=\varnothing, \bigcup_{k=1}^{r} J_k=J.$$

J kümesinin özel bir parçası $J^r=\{J_1,...,J_r\}$, $I=\{1,...,r\}$ aşağıdaki enb-enk tipinde fonksiyonu tanımlar (Bagirov, 2005):

$$\varphi(z)=\max_{i\in I}\min_{j\in J_i}\{\langle x^j, z\rangle - y_j\}, z\in\mathbb{R}^n.$$

Enb-enk ayrılabilirlik için aşağıdaki tanımlar verilmiştir (Bagirov, 2005):

1) $A,B\in\mathbb{R}^n$, verilen iki ayrık küme olsun, yani $A\cap B=\varnothing$. A ve B kümeleri enb-enk ayrılabilirdir. Eğer

$$\{x^j, y_j\}, j\in J=\{1,...,l\}, x^j\in\mathbb{R}^n, y_j\in\mathbb{R}^1$$

şeklinde sonlu hiperdüzlemler kümesi var ise ve J kümesinin $J^r=\{J_1,...,J_r\}$ parçalanması aşağıdaki özellikleri taşıyorsa;

a) her $i\in I$ ve $a\in A$ için

$$\min_{j\in J_i}\{\langle x^j, a\rangle - y_j\}<0;$$

b) herhangibir $b\in B$ için en az birtane $i\in I$ vardır öyle ki

$$\min_{j\in J_i}\{\langle x^j, b\rangle - y_j\}>0.$$

2) A kümesi $A_i,\ i=1,...,q$ kümelerinin bileşimi olsun:

$$A=\bigcup_{i=1}^{q}A_i$$

Ve herhangi $i=1,...,q$ için,

$$B\bigcap coA_i=\varnothing.$$

Bu durumda A ve B kümeleri enb-enk ayrılabilirdir.

3) A ve B kümeleri enb-enk ayrılabilirdir ancak ve ancak ayrık kümeler iseler:

$$A\cap B=\varnothing.$$

Doğrusal ve çokyüzlü ayrılabilirlik enb-enk ayrılabilirliğinin özel durumları olarak ele alınabilir. Eğer $I=\{1\}\ ve\ J_1=\{1\}$ ise doğrusal ayrım söz konusudur ve eğer $I=\{1,...,h\}\ ve\ J_i=\{i\}, i\in I$ ise h-çokyüzlü ayrım vardır.

İkili doğrusal ayrım da yine enb-enk ayrımın özel durumu olarak ele alınabilir. A ve B kümelerinin ikili doğrusal ayrımı aşağıdaki durumlardan biriyle çakışabilir:

1. $coA\cap B=\varnothing$ ise A ve B iki çokyüzlü ayrılabilirdir;

2. $coB\cap A=\varnothing$ ise A ve B iki çokyüzlü ayrılabilirdir;

3. A ve B aşağıdaki hiperdüzlemlerle enb-enk ayrılabilirdir:

$$\left\{\left(x^1,y_1\right),\left(-x^1,-y_1\right),\left(x^2,y_2\right)\left(-x^2,-y_2\right)\right\}.$$

Bu durumda $I=\{1,2\}\ ve\ J_1=\{1,4\}, J_2=\{2,3\}$ dur.

A kümesi , A_i, $i=1,...,q$ lerin, B kümesi ise B_j, $j=1,...,d$ lerin bileşimi olsun öyle ki;

$$A=\bigcup_{i=1}^{q}A_i\,,\ B=\bigcup_{j=1}^{d}B_j$$

ve

$$\text{herbir } i=1,...,q,\ j=1,...,d \text{ için } coB_j\bigcap coA_i=\varnothing.$$

Bu durumda A ve B kümelerinin ayrımı için gerekli hiperdüzlem sayısı en fazla $q.d$ kadardır (Bagirov, 2005).

Enb-enk ayrılabilirliğin sağlanmadığı durumlar için bir hata fonksiyonu tanımlanmıştır (Bagirov, 2005).

$\left\{x^j,y_j\right\}$, $j=1,...,l, x^j\in\mathbb{R}^n, y_j\in\mathbb{R}^1$ hiperdüzlemler kümesi ve J kümesinin herhangi bir parçası $J^r=\left\{J_1,...,J_r\right\}$ olarak verilsin. Aşağıdaki koşul sağlanıyorsa herhangi bir $a\in A$ noktası B kümesinden iyi bir şekilde ayrılır denir:

$$\underset{i\in I}{enb}\ \underset{j\in J}{enk}\left\{\left\langle x^j,a\right\rangle-y_j\right\}+1\le 0.$$

Öyleyse herhangi bir $a\in A$ noktası için hata fonksiyonu aşağıdaki şekilde tanımlanabilir:

$$enb\left[0,\underset{i\in I}{enb}\ \underset{j\in J_i}{enk}\left\{\left\langle x^j,a\right\rangle-y_j+1\right\}\right].$$

Benzer olarak, herhangibir $b\in B$ noktası eğer aşağıdaki koşulu sağlıyorsa A kümesinden iyi bir şekilde ayrılır denir:

$$\underset{i\in I}{enk}\ \underset{j\in J}{enb}\left\{-\left\langle x^j,b\right\rangle+y_j\right\}+1\le 0.$$

Öyleyse herhangi bir $b \in B$ noktası için hata fonksiyonu aşağıdaki şekilde tanımlanabilir:

$$enb\left[0, \underset{i\in I}{enk}\ \underset{j\in J_i}{enb}\left\{-\left\langle x^j, b\right\rangle + y_j + 1\right\}\right].$$

Böylece tüm noktalar için hata fonksiyonu aşağıdaki şekilde tanımlanabilir;

$$f(x,y) = (1/m)\sum_{k=1}^{m} enb\left[0, \underset{i\in I}{enb}\ \underset{j\in J_i}{enk}\left\{\left\langle x^j, a^k\right\rangle - y_j + 1\right\}\right]$$
$$+(1/p)\sum_{t=1}^{p} enb\left[0, \underset{i\in I}{enk}\ \underset{j\in J_i}{enb}\left\{-\left\langle x^j, b^t\right\rangle + y_j + 1\right\}\right].$$

Burada $x = (x^1, ..., x^l) \in \mathbb{R}^{l\times n}, y = (y_1, ..., y_l) \in \mathbb{R}^l$ dir ve her $x \in \mathbb{R}^{l\times n}, y \in \mathbb{R}^l$ için $f(x,y) \geq 0$ olduğu açıktır.

A ve *B* kümeleri enb-enk ayrılabilirdir eğer;

$$\left\{x^j, y_j\right\}, j \in J = \left\{1, ..., l\right\}, x^j \in \mathbb{R}^n, y_j \in \mathbb{R}^1$$

hiperdüzlemler kümesi ve J kümesinin $J^r = \left\{J_1, ..., J_r\right\}$ parçası olacak şekilde $f(x,y) = 0$ ise.

Birçok durumda özel bir parça hiperdüzlemler kümesi *A* ve *B* kümelerini ayırıyorsa aynı parça için başka bir hiperdüzlemler kümeside *A* ve *B* kümelerini ayırabilir. Tanımlanan hata fonksiyonu dışbükey değildir ve eğer *A* ve *B* kümeleri enb-enk ayrılabilir ise bu fonksiyonun genel enküçüğü $f(x^*, y_*) = 0$ dır ve genel en küçükleyeni tek değildir.

Enb-enk problemi aşağıdaki matematiksel programlama problemine indirgenebilir (Bagirov, 2005):

$$enk\ f(x,y)\ ,\ (x,y) \in \mathbb{R}^{(n+1)\times l}$$

Burada f fonksiyonu aşağıdaki formda tanımlanır:

$$f(x,y) = f_1(x,y) + f_2(x,y)$$

ve

$$f_1(x,y) = (1/m)\sum_{k=1}^{m} enb\left[0, \underset{i\in I}{enb}\ \underset{j\in J_i}{enk}\left\{\left\langle x^j, a^k\right\rangle - y_j + 1\right\}\right],$$
$$f_2(x,y) = (1/p)\sum_{t=1}^{p} enb\left[0, \underset{i\in I}{enk}\ \underset{j\in J_i}{enb}\left\{-\left\langle x^j, b^t\right\rangle + y_j + 1\right\}\right].$$

Burada tanımlanan enküçükleme problemi bir küresel optimizasyon problemidir. Ancak problemdeki değişken sayısı çok fazla olduğu için küresel optimizasyon için geliştirilmiş yöntemler direkt olarak bu probleme uygulanamamaktadır. Bu nedenle ifade edilen problemin f fonksiyonunun enküçüğünü bulmada yerel yöntemler ön plana çıkmıştır.

Bagirov, Ugon, Webb, Öztürk ve Kasımbeyli tarafından 2011 yılında yayınlanan makalede enb-enk ayrım algoritması geliştirilerek hesaplama güçlüğünü azaltmak ve ayrım kalitesini arttırmak amaçlanmıştır. Bu doğrultuda sınıflar arası parçalı doğrusal sınırları bulmak için birden fazla aşamalı bir algoritma tasarlanmıştır ve ikinci aşamada enb-enk ayrım kullanılırken, ilk aşamada, bir sonraki bölümde anlatılacak olan çokyüzlü konik fonksiyonlardan faydalanılmıştır (Bagirov et. al., 2011).

4. ÇOKYÜZLÜ KONİK FONKSİYONLAR İLE İKİLİ SINIFLANDIRMA

Çokyüzlü konik fonksiyonlar, Gasimov'un 2001 yılında tanıttığı çokyüzlü fonksiyonların özel bir sınıfına, doğrusal bir parça eklenmesi sayesinde l_1 normun genişletilmesiyle oluşturulur. Bu gibi fonksiyonların grafının, en fazla 2^n yarı uzayın kesişimini içeren altseviye kümesine sahip ve tepe noktası $(a,-\gamma)\in\mathbb{R}^n\times\mathbb{R}$ olan bir çokyüzlü koni olduğu Gasimov ve Öztürk tarafından ispatlanmıştır (Gasimov and Öztürk, 2006; Gasimov, 2001).

$g_{(w,\xi,\gamma,a)}:\mathbb{R}^n\to\mathbb{R}$ çokyüzlü konik fonksiyonu aşağıdaki şekilde tanımlanır:

$$g_{(w,\xi,\gamma,a)}:\mathbb{R}^n\to\mathbb{R}=w'(x-a)+\xi\left\|x-a\right\|_1-\gamma$$

Burada $w,a\in\mathbb{R}^n,\xi,\gamma\in\mathbb{R},\quad w'x=w_1x_1+...+w_nx_n$ ifadesi w ve x vektörlerinin skaler çarpımı ve $\left\|x\right\|_1=\left|x_1\right|+...+\left|x_n\right|$ ise x vektörünün l_1 normudur.

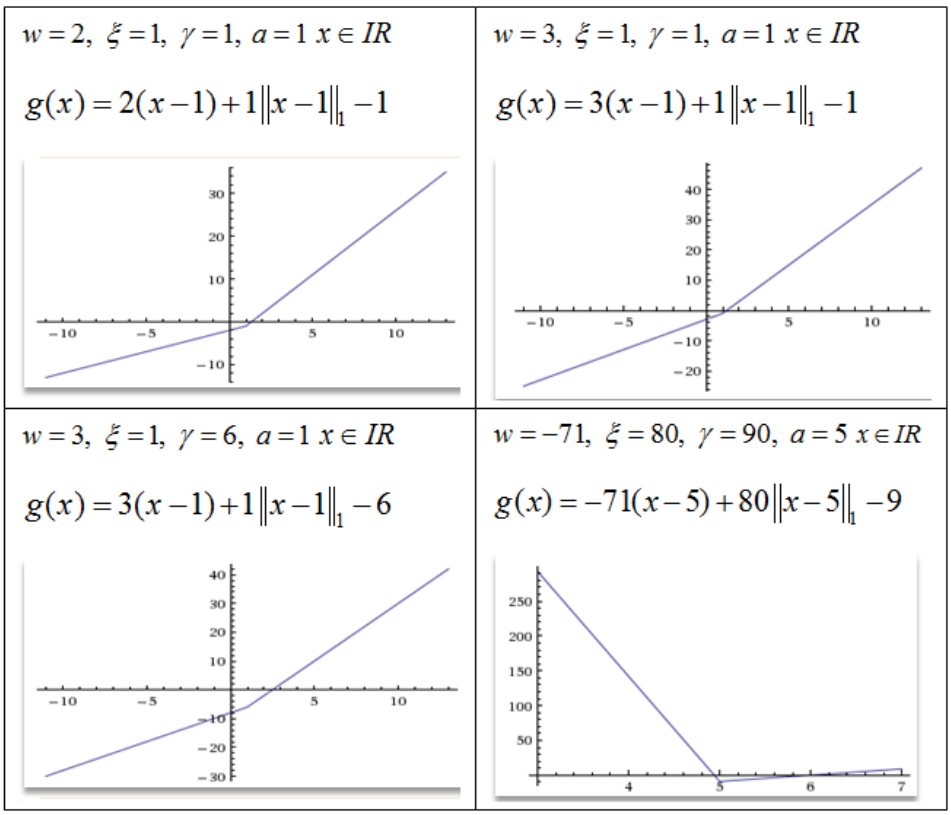
Çokyüzlü konik fonksiyonların altseviye kümeleri

$$S_\alpha=\left\{x\in\mathbb{R}^n:g(x)\le\alpha\right\},$$

$\alpha\in\mathbb{R}$, için çokyüzlü şeklindedirler.

Bu tanımlar doğrultusunda eğer $g_{(w,\xi,\gamma,a)}:\mathbb{R}^n\to\mathbb{R}$ fonksiyonunun grafı bir koni ve $\alpha\in\mathbb{R}$ için tüm altseviye kümeleri birer çokyüzlü ise bu g fonksiyonu çokyüzlü bir konik olarak tanımlanır (Gasimov and Öztürk, 2006).

Şekil 4.1'de çokyüzlü konik fonksiyonların anlaşılması için 2 boyutlu uzayda w,ξ,γ ve a fonksiyon parametrelerine farklı değerler atanarak oluşturulan fonksiyon grafikleri MATLAB kullanılarak görsel olarak sunulmuştur.



Şekil 4.1. Çokyüzlü konik fonksiyonların 2-boyutlu uzayda gösterimi

4.1 Çokyüzlü Konik Fonksiyon Algoritması

Gasimov R. ve Öztürk G. tarafından oluşturulan çokyüzlü konik fonksiyonlar (ÇKF) temelli açgözlü ikili sınıflandırma algoritması aşağıda şekilde verilmiştir (Gasimov and Öztürk, 2006).

Algoritma 1: (ÇKF algoritması)

A ve B, $\mathbb{R}^n$ de tanımlı iki küme olsun.

$$A=\left\{a^i\in\mathbb{R}^n : i\in I\right\}, B=\left\{b^j\in\mathbb{R}^n : j\in J\right\}$$

burada $i=\{1,..,m\}$, $j=\{1,..,p\}$ dir.

Başlangıç Adımı: l=1, $I_l=I$, $A_l=A$ olsun. Adım 1 e git.

Adım 1: a^l , A^l nin keyfi bir noktası olsun. P_l altproblemini çöz.

$$(P_l) \quad \min\left(\frac{y^{'} e_{|I_l|}}{|I_l|}\right) \tag{4.1}$$

$$w^{'}(a^i - a^l) + \xi \left\| a^i - a^l \right\|_1 - \gamma + 1 \le y_i, \quad \forall i \in I_l, \tag{4.2}$$

$$-w^{'}(b^j - a^l) - \xi \left\| b^j - a^l \right\|_1 + \gamma + 1 \le 0, \quad \forall j \in J, \tag{4.3}$$

$$y = (y_1, ..., y_m) \in R_+^m, w \in R^n, \xi \in R, \gamma \ge 1 \tag{4.4}$$

$w^l, \xi^l, \gamma^l, y^l$, (P_l)'nin bir çözümü olsun.

$$g_l(x) = g_{(w^l, \xi^l, \gamma^l, a^l)}(x) \tag{4.5}$$

Adım 2 ye git.

Adım2: $I_{l+1} = \left\{ i \in I_l : g_l(a^i) + 1 > 0 \right\}, A_{l+1} = \left\{ a^i \in A_l : i \in I_{l+1} \right\}, l = l + 1$ yap.

Eğer $A_l \neq \varnothing$ ise Adım 1'e git.

Adım 3: *A* ve *B* kümelerini birbirinden ayıran *g(x)* fonksiyonunu aşağıdaki şekilde tanımla:

$$g(x) = \min_l g_l(x) \tag{4.6}$$

P_l minimizasyon probleminin (4.4) kısıtında bulunan $\gamma \ge 1$ eşitsizliği koninin tepe noktasının *z=0* hiperdüzlemi altında yeralmasını sağlar; yani $\mathbb{R}^n \times (0, -\infty)$ yarı uzayında çalışmayı garanti eder. (4.2) kısıtı a^l ve a^l ye yakın tüm $a \in A^l$ noktalarının, altseviye kümesine, $\left\{ x : g_l(x) \le -1 \right\}$, karşılık gelen çokyüzlü içerisine düşmesini garanti eder. Bu A^l noktalarının, a^l tepe noktasına

olan yakınlıkları (4.1) de yer alan (P_l) amaç fonksiyonunun optimal değeri ile tanımlanır. Öyle ki, bu amaç fonksiyonunun değeri "0" değerine yaklaştıkça $\{x : g_l(x) \le -1\}$ altseviye kümesi, A^l içerisinden daha fazla elemanı içine alır. (4.3) kısıtı ise herbir iterasyonda B kümesinin tüm elemanlarının altseviye kümesi dışında kalmasını garanti eder.

Aşağıda algoritmanın daha iyi anlaşılması için 2 ve 3 boyutlu uzayda iki örnek üzerinde algoritmanın işleyişi gösterilmiş ve MATLAB programı yardımıyla sonuçlar görsel olarak sunulmuştur.

Örnek 4.1: A ve B kümeleri aşağıdaki şekilde tanımlansın. Şekil 4.2'da sunulan grafiksel gösterimde mor noktalar B küme elemanlarını, beyaz noktalar ise A küme elemanlarını temsil etmektedir.

$A, B \in \mathbb{R}$ olmak üzere, $A = \{1,2,5,6\}$, $B = \{3,4,7\}$

ÇKF algoritması ile çözüm:

Algoritma bu problem için çalıştırıldığında ilk keyfi noktanın a=1 seçildiğini varsayalım.

Adım1: P_1 minimizasyon probleminin çözümüyle $g_1(x) = 3(x-1) + 1\|x-1\|_1 - 6$ elde edilsin.

Adım 2: $I_2 = \{i \in I_2 : g_1(a^i) > 0\}$, $A_2 = \{a^i \in A_1 : i \in I_2\} = \{5,6\}$ $A_2 \neq \varnothing$ olduğundan Adım 1 e git.

Adım 1: A_2 kümesi içinden keyfi nokta a=5 seçildiğini varsayalım.

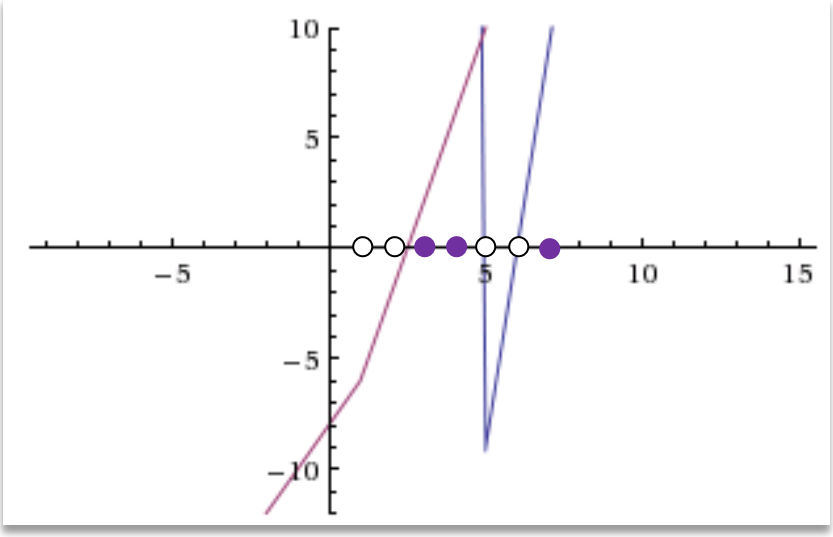
P_2 çözümü sonrası $g_2(x) = -71(x-5) + 80\|x-5\|_1 - 9$ elde edilsin.

Adım 2: $I_3 = \{i \in I_3 : g_2(a^i) > 0\}$, $A_3 = \{a^i \in A_1 : i \in I_3\} = \varnothing$

$A_3 = \varnothing$ olduğundan Adım 3 e git.

Adım 3: A yı B den ayıran $g(x)$ fonksiyonu aşağıdaki şekilde tanımlanır.

$$g(x) = \min_l g_l(x)$$



Şekil 4.2. Elde edilen *g(x)* ayırıcı fonksiyonunun 2-boyutlu uzayda grafiksel sunumu

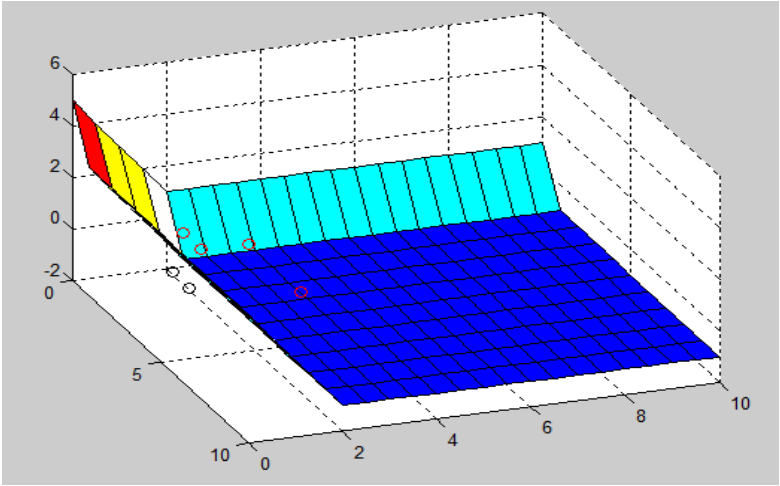
Örnek 4.2: A ve B kümeleri aşağıdaki şekilde tanımlansın. Grafiksel gösterimde siyah noktalar B küme elemanlarını, kırmızı noktalar ise A küme elemanlarını temsil etmektedir.

$A, B \in \mathbb{R}^2$ olmak üzere , A={{1,2},{2,2},{2,3},{5,3}}, B={{3,1},{4,1}}

ÇKF algoritmasının uygulanması sonucunda elde edilen ayırıcı fonksiyon aşağıdaki şekilde tanımlanmıştır ve şekil 4.3'de görsel olarak sunulmuştur:

$$g(x) = \begin{pmatrix} -1 \\ -1 \end{pmatrix} (x - (1,2)) + \left\| x - (1,2) \right\|_1 - 1$$

Şekil 4.3'de görüldüğü üzere tek bir çokyüzlü konik fonksiyon kullanımı ile kesin ayrım sağlanmıştır.



Şekil 4.3. 3-boyutlu uzayda elde edilen *g(x)* ayırıcı fonksiyonu

Yukarıda belirtilen çokyüzlü konik fonksiyon algoritması sonlu adımda durmakta ve elde edilen ayırıcı fonksiyon ile herhangi iki kümeyi tam olarak ayırabilmektedir. Bu çözüm yolunda, iki kümenin ayrımı büyük ölçüde herbir iterasyonda oluşturulan çokyüzlü konik fonksiyonu temsil eden çokyüzlü konilerin tepe noktalarının $(a^l, -\gamma^l) \in \mathbb{R}^n \times \mathbb{R}$ yerine bağlıdır. Bu noktanın γ^l koordinatı DP altprobleminin çözümü ile bulunurken a^l koordinatı her bir iterasyonda A^l kümesinden keyfi olarak seçilmektedir. Yapılan bu keyfi nokta seçimleri kimi zaman ayrım kalitesini düşürmekte ve iterasyon sayısını arttırmaktadır. Gasimov ve Öztürk'ün 2006 yılında yayınlanan makalelerinde en iyi noktayı seçmek için ÇKF algoritmasının ilk adımı aşağıdaki şekilde modifiye edilmiştir.

Herbir *l.* iterasyonunda $a_i^l \in A^l$ noktası için (P_l) altproblemi çözülüp A^l kümesinde yer alıp *B*'de yer almayan A^l noktalarının sayısı, l_i, bulunur sonrasında tepe noktasının a^l koordinatı, $a^l = a^{l_0}$, $l_0 = \max\{l_i : i \in I_l\}$ olacak şekilde tanımlanır (Gasimov and Öztürk, 2006).

Gerçektende geliştirilen bu algoritma ile verimlilik arttırılmıştır; ancak verimliliğin sağlandığı bu uygulamalar sadece *A* kümesi eleman sayısının az olduğu uygulamalarla kısıtlı kalmıştır (Gasimov and Öztürk, 2006).

Tezde bu zorluğu ortadan kaldırmak için ÇKF algoritmasına yine bir veri madenciliği yaklaşımı olan kümeleme yöntemi eklenmiştir. Bir sonraki bölümde kümeleme yönteminin temel kavramları ve çözüm algoritmaları incelenecektir.

5. VERİ KÜMELEME YÖNTEMİ

Veri madenciliği yöntemlerinden biri olan kümeleme analizi, verilerin özelliklerini gözönüne alarak, birbirleri ile benzer olan verileri alt kümelere ayırmayı sağlayan çok boyutlu veri analiz yöntemidir. Kümelemede amaç, bir veri kümesini benzer veriler aynı kümede bulunacak şekilde bölümlere ayırmaktır. Kümeleme, aynı zamanda gözetimsiz (unsupervised) veri sınıflandırması olarakta bilinir ve sınıflandırma gibi tıp, ekonomi, bankacılık, mühendislik vb. birçok alanda uygulamalara sahiptir.

Kümeleme problemlerinin kesin (hard) ve bulanık (fuzzy) kümeleme olmak üzere iki farklı türü vardır. Kesin kümelemede herbir veri noktasının tek bir kümeye ait olduğu varsayılırken, bulanık kümelemede veri elemanları belli bir üyelik derecesi ile birden fazla kümeye ait olabilirler. Tezde kesin kümeleme problem çözümünden faydalanılacağı için bulanık kümeleme konusu incelenmemiştir.

5.1 Kümeleme Probleminin Matematiksel Tanımı

n-boyutlu *m* noktadan oluşan $\mathbb{R}^n$ uzayında sonlu sayıda elemana sahip *A* kümesini ele alalım:

$$A = \left\{a^1, ..., a^m\right\}, \text{ ve } a^i \in R^n,\ i = 1, ..., m.$$

Buna göre kesin kümeleme probleminin amacı, *A* kümesindeki noktaları, verilen *k* adet ayrık A^j, $j=1,...,k$ altkümeye, önceden tanımlanmış aşağıdaki kurallara göre ayırmaktır:

1) $A^j \neq \varnothing,\ j = 1, ..., k;$

2) $A^j \bigcap A^l = \varnothing,\ \ j, l = 1, ..., k,\ j \neq l;$

3) $A = \bigcup_{j=1}^{k} A^j;$

4) A^j , $j=1,...,k$ kümeleri için bir kısıt bulunmamaktadır.

Burada A^j, $j=1,...,k$ altkümelerine küme(cluster) adı verilir. Yukarıdaki maddelerdende görüldüğü üzere, oluşacak her bir kümenin boş küme olmaması, hiçbir küme çiftinin ortak bir elemana sahip olmaması, kümelerin birleşiminin veri kümesine eşit olması ve kümeler için bir kısıt bulunmaması kuralları esas alınmaktadır.

Varsayalım ki herbir A^j kümesi kendi merkezi olan x^j noktası ile tanımlansın. O zaman kümeleme problemi aşağıdaki optimizasyon problemine indirgenebilir (Bagirov et al., 2003):

$$enk\psi_k(x,w) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k} w_{ij}\left\|x^j - a^i\right\|^2$$

$$\sum_{j=1}^{k} w_{ij} = 1, i = 1,...,m$$

$$x = (x^1,...,x^k) \in R^{n\times k},$$

$$w_{ij} = 0 \text{ veya } 1,\ i = 1,...,m,\ j = 1,...,k \text{ kısıtları altında}$$

Burada w_{ij}: a^i örneğinin j kümesine aşağıdaki koşullara göre ait olma durumudur:

$$w_{ij} = \begin{cases} 1, \text{eğer } a^i \text{elemanı } j \text{ kümesine atandıysa,} \\ 0, \text{aksi halde} \end{cases}$$

ve $$x^j = \frac{\sum_{i=1}^{m} w_{ij}a^i}{\sum_{i=1}^{m} w_{ij}},\ j = 1,...,k \quad .$$

Burada $w, m\times k$ boyutlu bir matristir ve $\|.\|$ öklid normunu ifade eder.

Kümeleme problemi yukarıda da görüldüğü gibi bir küresel optimizasyon problemidir ve çözümü için çok çeşitli yöntemler önerilmiştir. Bu yöntemler dinamik programlama, dal sınır, düzlem kesme, iç nokta arama metodları, tabu arama, genetik algoritmalar gibi kesin ve yaklaşık çözüm yöntemleridir ancak bu

yöntemler büyük veri setlerinde çözüm için etkili sonuçlar vermemektedirler bu nedenle kesin kümeleme problemleri için *k*-ortalamalar (MacQueen, 1967), genel *k*-ortalamalar (Likas et al., 2001) ve modifiye edilmiş genel *k*-ortalamalar (Bagirov et al., 2003; Bock, 1998) vb. farklı çözüm yöntemleri de sunulmuştur.

Tezde, 1967 yılında James MacQueen in önerdiği *k*-ortalamalar ve bir yönüyle *k*-ortalamalardan farklı olan *k*-medyan metodları kullanılacaktır. Bu nedenle bir sonraki bölümde *k*-ortalamalar ve *k*-medyan algoritmaları incelenmiştir.

5.2 *k*-Ortalamalar Algoritması

Kesin kümeleme için en çok kullanılan algoritmalardan biri *k*-ortalamalar algoritmasıdır. Bu algoritma aşağıdaki şekilde ifade edilir (MacQueen, 1967):

***k*-Ortalamalar algoritması:**

Adım 1: k adet merkez içeren (merkezlerin A kümesine ait olması gerekmez) bir başlangıç çözümü seçilir.

Adım 2: A kümesindeki verilerin herbiri, seçilen k adet merkez gözönüne alınarak en yakın merkezli kümeye atanır.

Adım 3: Adım 2 deki kümeleme gözönüne alınarak, her bir kümenin yeni merkezleri hesaplanır ve hiçbir veri, kümesini değiştirmeyinceye kadar adım 2 ye dönülür.

Bu algoritma, sözü edilen başlangıç noktalarının seçimine oldukça duyarlıdır ve büyük veri kümeleri için, bu başlangıç noktalarına bağlı olarak, genel çözümden oldukça farklı yerel çözümlerde bulabilmektedir.

5.3 *k*- Medyan Algoritması

k-ortalamalar algoritmasının ilk adımında seçilen k adet keyfi merkezin A kümesine ait olması gerekmiyordu ancak bazı uygulamalarda bu noktaların A

kümesine ait olması istenebilir, bu gibi durumlar için Kaufmann ve Rousseeuw'un 1990'da tanıttıkları *k*-medyan kümeleme yöntemi kullanılır (Kaufmann and Rousseeuw, 1990).

***k*-medyan Algoritması:**

Adım 1: *k* adet merkez içeren (merkezlerin *A* kümesine ait olması gerekir) bir başlangıç çözümü seçilir.

Adım 2: *A* kümesindeki verilerin herbiri, seçilen *k* adet merkez gözönüne alınarak en yakın merkezli kümeye atanır.

Adım 3: Adım 2 deki kümeleme gözönüne alınarak, her bir kümenin yeni merkezleri hesaplanır ve hiçbir veri, kümesini değiştirmeyinceye kadar adım 2 ye dönülür.

6. ÇOKYÜZLÜ KONİK FONKSİYONLAR TABANLI İKİLİ SINIFLANDIRMADA KÜMELEME YÖNTEMİNİN KULLANIMI

4. bölümde incelenen ÇKF algoritmasında iki kümenin ayrımı büyük ölçüde herbir iterasyonda oluşturulan çokyüzlü konik fonksiyonu temsil eden çokyüzlü konilerin tepe noktalarının $(a^l, -\gamma^l) \in \mathbb{R}^n \times \mathbb{R}$ yerine bağımlı idi. Bu noktanın γ^l koordinatı DP altprobleminin çözümü ile bulunurken, a^l koordinatı her bir iterasyonda A^l kümesinden keyfi olarak seçilmekteydi. Yapılan bu keyfi nokta seçimleri kimi zaman ayrım kalitesini düşürdüğü ve iterasyon sayısını arttırdığı için Gasimov ve Öztürk'ün 2006 yılında yayınlanan ve ÇKF ile ikili sınıflandırmayı tanımladıkları makalelerinde en iyi noktayı seçmek için ÇKF algoritmasının ilk adımına değişiklik uygulanmıştı ve algoritmanın herbir *l.* iterasyonunun ilk adımında $a_i^l \in A^l$ noktası için (P_l) altproblemi çözülüp, A^l kümesinde yer alan *B*'de yer almayan A^l noktalarının sayısı, l_i, bulunup sonrasında tepe noktasının a^l koordinatı, $a^l = a^{l_0}$, $l_0 = \max\{l_i : i \in I_l\}$ olacak şekilde tanımlanmıştı (Gasimov and Öztürk, 2006).

Geliştirilen bu algoritma tepe noktasının keyfi olarak seçildiği algoritma uygulamalarından daha verimli sonuçlar vermiştir ancak bu algoritmanın verimli olduğu uygulamalar sadece *A* kümesinde yer alan eleman sayısının az olduğu durumlardır.

Bu tezin amaçlarından bir tanesi ÇKF algoritmasını geliştirerek *A* kümesi eleman sayısı büyük olduğu durumlarda dahi verimli sonuçlar bulacak şekilde tasarlamak ve dolayısıyla algoritmayı yoğun işlem yükünden kurtarmaktır.

Algoritmanın verimliliği daha önceden de bahsedildiği üzere büyük ölçüde tepe noktası $(a^l, -\gamma^l) \in \mathbb{R}^n \times \mathbb{R}$ seçimlerine bağlı idi. Bu nedenle yeni tasarlanacak algoritmada yine bir veri madenciliği uygulaması olan ve 5. bölümde açıklanan veri kümeleme yöntemi kullanılarak en iyi a^l noktalarını bulmak amaçlanmıştır (Bkz. Bölüm 5). Kümeleme aigoritmasının uygulanması sonucunda bulunan herbir *k* adet küme merkezi, ÇKF tepe noktalarının a^l koordinatlarını

temsil edecektir. Bu $(a^l, -\gamma^l) \in \mathbb{R}^n \times \mathbb{R}$ tepe noktaları kullanılarak oluşturulan g_l fonksiyonlarının, ençok sayıda A^l noktasını B den ayırabilir duruma gelmesi, böylece iterasyon sayısını azaltarak algoritma verimliliğini arttırmak amaçlanmaktadır.

6.1 Kümeleme Tabanlı ÇKF Algoritmaları

Aşağıda kümeleme yöntemi tabanlı geliştirilmiş iki ÇKF algoritması tanıtılmıştır. İlk olarak tasarlanan algoritmada kümeleme yöntemi için *k*-ortalamalar ve *k-medyan* yöntemleri kullanılmıştır. Bir diğer algoritmada ise ÇKF algoritmasına sadece kümeleme yönteminin eklenmesinin yanısıra algoritma daha da geliştirilerek eğitim ve test başarım değerleri arası farkı azaltarak aşırı uyumdan kaçınmak için, algoritmada yer alan doğrusal programlama problemi yeniden tasarlanmıştır.

6.1.1 Kümeleme tabanlı ÇKF algoritması 1

ALGORİTMA 2: **KMLM-ÇKF 1**

A ve B, $\mathbb{R}^n$ de tanımlı kümeler olsun.

$$A = \left\{ a^i \in IR^n : i \in I \right\}, B = \left\{ b^j \in IR^n : j \in J \right\}.$$

burada $i = \{1,..,m\}$, $j = \{1,..,p\}$ dir.

Başlangıç Adımı: k=1, $A_k = A, I^k = I$.

Adım 1: A kümesinde kümeleme algoritmasını uygula. Küme(cluster) sayısı (s) olsun. $\bigcup_k^s I^k = I$.

Adım 2: $a_k \in A$, A nin k. orta noktası olsun. P_k altproblemini çöz.

$$(P_k) \quad \min\left(\frac{y' e_{|I^k|}}{|I^k|}\right) \tag{6.1}$$

$$w'(a^i - a_k) + \xi \left\| a^i - a_k \right\|_1 - \gamma + 1 \le y_i, \quad \forall i \in I^k \tag{6.2}$$

$$-w'(b^j - a_k) + \xi \left\| b^j - a_k \right\|_1 + \gamma + 1 \le 0, \quad \forall j \in J \tag{6.3}$$

$$y = (y_1, ..., y_{I^k}) \in \mathbb{R}_+^{I^k}, w \in \mathbb{R}^n, \xi \in \mathbb{R}, \gamma \ge 1 \tag{6.4}$$

$w_k, \xi_k, \gamma_k, y_k$, (P_k) ' nın bir çözümü olsun.

$$g_k(x) = g_{(w_k, \xi_k, \gamma_k, a_k)}(x) \tag{6.5}$$

Adım 3: $k < s$ ise $k=k+1$ yap. Değilse Adım 5'e git.

Adım 4: $A_k = \left\{ a^i \in A_{k-1} : g_l(a^i) > 0, l = 1, .., k \right\}$, $I^k = \left\{ i \in I^k : a^i \in A_k \right\}$ yap ve Adım 2 'ye git.

Adım 5: *A* ve *B* kümelerini birbirinden ayıran *g(x)* fonksiyonunu aşağıdaki şekilde tanımla,

$$g(x) = \min_k g_k(x)$$

ve Dur.

Algoritmanın ilk adımında *A* kümesine bir kümeleme yöntemi uygulanarak "s" adet tepe noktası belirlenmiş ve bir sonraki adımda belirlenen ilk tepe noktasına göre DP problemi (P_k), k=1, çözülmüştür elde edilen $g_k(x)$ ayırıcı fonksiyon ile ayrımı sağlanamayan *A* noktaları belirlenmiş ve $k=k+1$ yapılarak bu noktalarla yeni bir A_k kümesi, $A_k = \left\{ a^i \in A_{k-1} : g_l(a^i) > 0, l = 1, .., k \right\}$,

tanımlanmıştır ve bu küme ikinci merkez nokta tepe noktası tanımlanarak yeni bir DP problemi (P_k), k=2, çözülmüştür. İteratif olarak bu çözümler $k>s$ olana dek devam etmiştir. Görüldüğü gibi algoritmanın sonlandırma kriteri, ÇKF de olduğu gibi $A_k = \varnothing$ olma durumu değildir. Bu sebeple eğitim kümesinde %100 doğruluk sağlanamaz ancak süre ve genelleştirmede verimlilik elde edilebilir.

KMLM-ÇKF 1, ÇKF algoritmasıyla karşılaştırma amaçlı Gasimov ve Öztürk'ün 2006 yılında yayınladıkları makalede yer alan açıklayıcı örneğe uygulanmıştır .

Açıklayıcı Örnek 6.1: $\mathbb{R}^2$ de tanımlanan ve Çizelge 6.1'de belirtilen *A* ve *B* sonlu kümelerini ele alalım.

Çizelge 6.1: *A* ve *B* kümesinde yer alan veri noktaları

A	1	2	3	4	5	6	7	8	9	10	11	12
x	-2	-2	-2	-2	-0,5	-0,5	0,5	0,5	2	2	2	2
y	0,5	-0,5	2	-2	2	-2	2	-2	0,5	-2	-0,5	2

B	1	2	3	4	5	6	7	8	9	10	11	12
x	8	6	20	6	15	20	1	-1	-6	20	-6	-6
y	-6	-1	6	-6	6	1	6	6	1	-6	-1	-6

A	13	14	15	16	17	18	19	20	21	22	23	24
x	12	12	12	12	13,5	13,5	14,5	14,5	16	16	16	16
y	2	-2	0,5	-0,5	2	-2	2	-2	-0,5	0,5	2	-2

B	13	14	15	16	17	18	19	20	21	22	23	24
x	8	13	20	6	15	13	8	-1	-6	6	8	1
y	-1	6	-1	1	-6	-6	1	-6	6	6	6	-6

Bu kümelerin ayrım problemi çözümünde kümeleme için *k-medyan* ve *k-ortalamalar* yöntemleri kullanılmıştır. İlk olarak *k-medyan* yöntemi tercih edilmiştir çünkü bu yöntemle elde edilen merkez noktalar veri noktalarından biri

olma zorunluluğunu taşımaktadır ve bu noktayla yapılandırılan bir çokyüzlü konik fonksiyonun en az bir elemanı *B* kümesinden ayıracağı ispatlanmıştır (Gasimov and Öztürk, 2006). *k-ortalamalar* yönteminde ise böyle bir kısıt yoktur ve merkez noktası herhangibir nokta olarak belirlenebilmektedir. Her ne kadar bu yöntemde en az bir noktanın ayrımı garanti edilemesede bazı veri kümelerinde bu eksilen kısıt sayesinde daha iyi sonuçlar elde edildiği gözlenmiştir. İki yönteminde kullanıldığı MATLAB ortamında uygulanan algoritma sonuçları matematiksel ifadeler ve görsel sunumlarla aşağıda belirtilmiştir.

***k-medyan* kümeleme yöntemi ile ÇKF çözümü:**

Bu iki kümeyi birbirinden ayıran fonksiyonu yapılandırmak için *k*-medyan kümeleme yöntemli KMLM-ÇKF 1 uygulanmıştır. *k*-medyan kümeleme yöntemiyle (*k*=2) elde edilen merkezler ile bulunan çokyüzlü konik fonksiyonlar g_k ,(k=1,2) ve *A* kümesinin parçalanmaları olan ve g_k konilerinin, S_0^k altseviye kümeleri belirtilmiş ve Şekil 6.1'de gösterilmiştir.

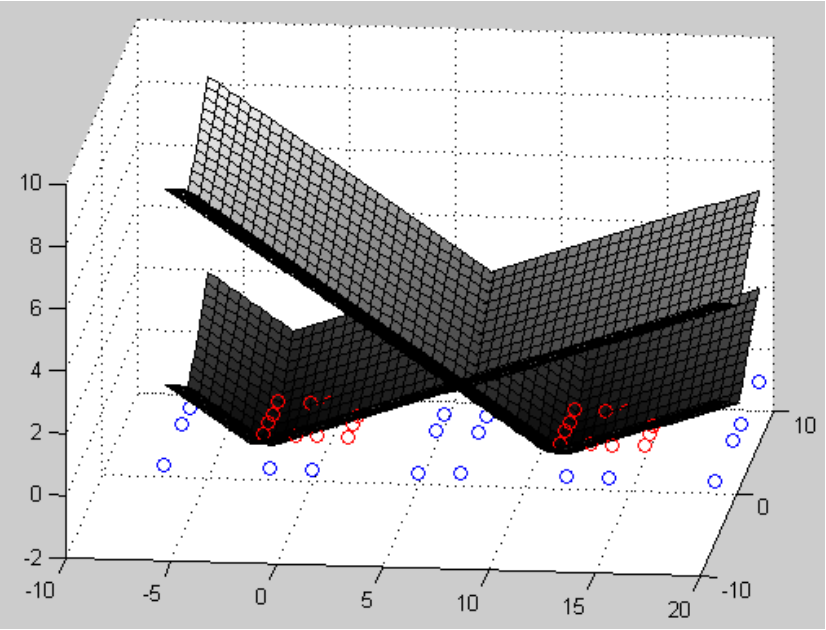
$$g_1(x) = -0.1124(x_1 - 12) + 0.0643(x_2 - 0.5) + 0.3373(|x_1 - 12| + |x_2 - 0.5|) - 1$$

$$S_0^1 = \{\{12, 2\}, \{12, -2\}, \{12, 0.5\}, \{12, -0.5\}, \{13.5, 2\}, \{16, 0.5\}\}$$

$$g_2(x) = -0.1124(x_1 + 2) + 0.0643(x_2 - 0.5) + 0.3373(|x_1 + 2| + |x_2 - 0.5|) - 1$$

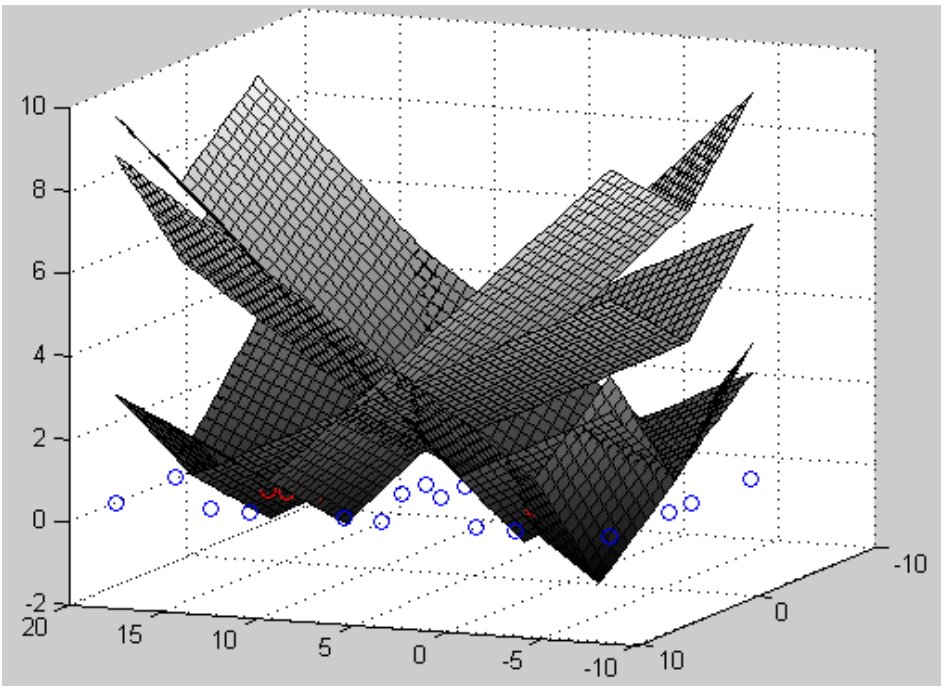
$$S_0^2 = \{\{-2, 0.5\}, \{-2, -0.5\}, \{-2, 2\}, \{-2, -2\}, \{-0.5, 2\}, \{2, -0.5\}\}$$

Görüldüğü gibi *k*-medyan ile bulunan merkezler doğrultusunda yapılandırılan 2 adet çokyüzlü konik fonksiyonun S_0 altseviye kümeleri toplamda 12 adet *A* noktasını içerip *B* noktalarını dışarıda bırakmıştır. Dolayısıyla verilerin yarısı ayrılmış yarısı ise ayrılamamıştır.



Şekil 6.1. *k*-medyan, *k*=2, ile elde edilen ayırıcı çokyüzlü konik fonksiyonlar

k-medyan yönteminde *k=4* alınırsa elde edilen sonuçta toplamda 22 noktada ayrım sağlanmış sadece (13.5,2) ve (14.5,2) noktalarında ayrım sağlanamamıştır. Elde edilen çokyüzlü konik fonksiyonlar Şekil 6.2'de gösterilmiştir.



Şekil 6.2. *k*-medyan, *k*=4, ile elde edilen ayırıcı çokyüzlü konik fonksiyonlar

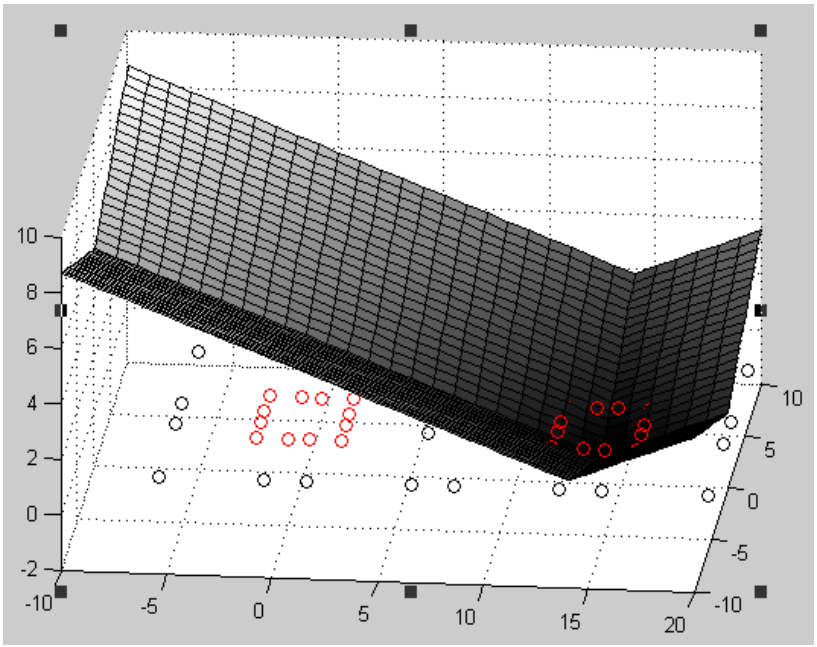
***k-ortalamalar* kümeleme yöntemi ile ÇKF çözümü:**

Bu çözümde KMLM-ÇKF 1'de yer alan Adım1 içinde kümeleme yöntemi için *k-medyan* yerine *k-ortalamalar* kullanılmıştır.

k-ortalamalar kümeleme yöntemiyle (*k*=2) elde edilen merkezler kullanılarak bulunan çokyüzlü konik fonksiyonlar g_k,(k=1,2) ve A kümesinin parçalanmaları olan ve g_k konilerinin, S_0^k altseviye kümesinde yer alan elemanlar belirtilmiş ve Şekil 6.3 ve 6.4'de gösterilmiştir.

$$g_1(x) = -0.0(x_1 - 14) + 0.0(x_2 - 0) + 0.2857(|x_1 - 14| + |x_2 - 0|) - 1$$

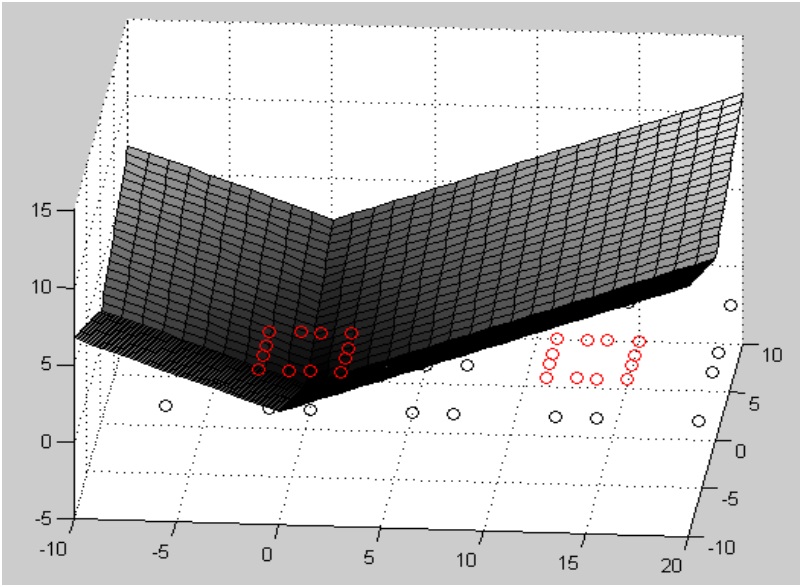
$$S_0^k = \begin{Bmatrix} \{12, 0.5\}, \{12, -0.5\}, \{13.5, 2\}, \{13.5, -2\}, \{14.5, 2\}, \\ \{14.5, -2\}, \{16, -0.5\}, \{16, 0.5\} \end{Bmatrix}$$



Şekil 6.3. Ayrımda kullanılan $g_1(x)$ çokyüzlü konik fonksiyonu

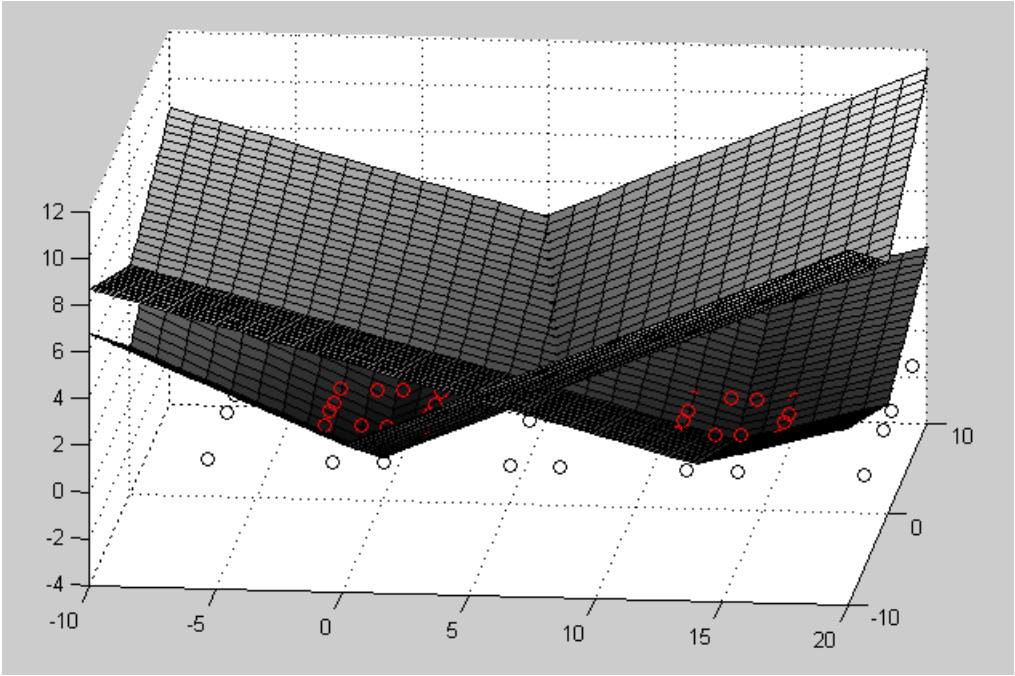
$$g_2(x) = 0.0(x_1 - 0) + 0.0(x_2 - 0) + 0.444(|x_1 - 0| + |x_2 - 0|) - 2.1111$$

$$S_0^2 = \begin{Bmatrix} \{-2,0.5\},\{-2,-0.5\},\{-2,2\},\{-2,-2\},\{-0.5,2\},\{-0.5,-2\}, \\ \{0.5,2\},\{0.5,-2\},\{2,0.5\},\{2,-2\},\{2,-0.5\},\{2,2\} \end{Bmatrix}$$



Şekil 6.4 Ayrımda kullanılan $g_2(x)$ çokyüzlü konik fonksiyonu

Her iki ayırıcı fonksiyonunda aynı grafikte gösterimi Şekil 6.5'de verilmiştir.



Şekil 6.5. Ayırmda kullanılan her iki çokyüzlü konik fonksiyon

Görüldüğü gibi *k-ortalamalar* ile bulunan merkezler doğrultusunda yapılandırılan 2 çokyüzlü konik fonksiyonun S_0 altseviye kümeleri toplamda 20 adet *A* noktasını *B* noktasından ayırmıştır. 4 adet, (12,2), (12,-2), (16,2), (16,-2), noktalarının ise ayrımı sağlanamamıştır.

Kümeleme yönteminde önemli bir yere sahip olan "*k"* değeri daha büyük değerler aldığında doğrulukta artış sağlanır ancak kullanılan çkf sayısı ve süre doğru orantılı olarak artış gösterir bu da verimliliği düşürebilir. Bu nedenle *k* değerinin seçimi büyük önem taşımaktadır.

ÇKF algoritmasıyla karşılaştırıldığında elde edilen sonuçlar Çizelge 6.2'de gösterilmiştir. Burada "*k*", kümeleme yöntemleri için belirlenen küme sayısını, süre, algoritma sonlanana kadar geçen süreyi, doğruluk, eğitim kümesinde yer alan verilerin % kaçının doğru sınıflandırıldığını ve kullanılan çkf sayısı ise ayırıcı fonksiyonda toplamda kaç çkf kullanıldığını belirtir.

Sonuç olarak açıklayıcı örneğe uygulanan *k*-ortalamalar ın kullanıldığı algoritmada, aynı *k* değeri için, *k*-medyan yönteminin kullanıldığı algoritmadan daha iyi sonuçlar elde edildiği gözlenmiştir. Ancak sadece bir örneğe dayanarak *k*-ortalamalar' in *k*-medyan ten daha iyi sonuçlar verdiği kesin yargısına varılamaz. Farklı veri kümelerinde sonuçlar değişiklik gösterebilmektedir. Aynı zamanda ÇKF algoritması ile karşılaştırıldığında *k=2* de doğrulukta eşitlik gözlenmese de sürede düşüş vardır. *k* değeri 4 yapıldığında ise doğruluk eşitlenmiş aynı zamanda süre de kısalmıştır.

Çizelge 6.2: Açıklayıcı örnek çözümünde elde edilen sonuçlar

	k	süre(sn)	doğruluk(%)	kullanılan çkf sayısı
k-means	2	0.32	84	2
	4	0.36	100	4
k-medoid	2	8.77	50	2
	4	9.32	84	4
ÇKF	_	4.5	100	4

Sonuç olarak bu açıklayıcı örneğe *k-ortalamalar* kümeleme yönteminin kullanıldığı algoritma uygulandığında daha verimli sonuçlar elde edilmiştir.

6.1.2 Kümeleme tabanlı ÇKF algoritması 2

Tasarlanan bu algoritmada ÇKF algoritmasına kümeleme yönteminin eklenmesinin yanısıra algoritma daha da geliştirilerek eğitim ve test başarım değerleri arası farkı azaltarak aşırı uyumdan (overfitting) kaçınmak için, algoritmada yer alan doğrusal programlama probleminde de değişiklikler yapılmıştır.

Tasarlanan algoritma ve yapılan değişikliklerin açıklaması aşağıda verilmiştir:

ALGORİTMA 3: **KMLM-ÇKF 2**

A ve B, $\mathbb{R}^n$ de tanımlı kümeler olsun.

$$A=\left\{a^i \in \mathbb{R}^n : i \in I\right\}, B=\left\{b^j \in \mathbb{R}^n : j \in J\right\}.$$

burada $i=\{1,..,m\}$, $j=\{1,..,p\}$ dir.

Başlangıç Adımı: l=1, $A_l = A$, $I_l = I$.

Adım 1: A kümesinde kümeleme algoritmasını uygula. Küme(cluster) sayısı (s) olsun. k=1 yap. $\bigcup_{k=1}^{s} I_k = I$

Adım 2: a_k , A nin k. orta noktası olsun. P_k altproblemini çöz.

$$(P_k) = \min\left\{\frac{\sum_{i=1}^{I_k} y_i}{\left|I_k\right|} + \frac{C\sum_{j=1}^{J} z_j}{\left|J\right|}\right\} \tag{6.6}$$

$$w'(a^i - a_k) + \xi \left\| a^i - a_k \right\|_1 - \gamma + 1 \le y_i, \quad \forall i \in I_k \tag{6.7}$$

$$-w'(b^j - a_k) + \xi \left\| b^j - a_k \right\|_1 + \gamma + 1 \le z_j, \quad \forall j \in J \tag{6.8}$$

$w_k, \xi_k, \gamma_k, y_k$, (P_k) ' nın bir çözümü olsun.

$$g_k(x) = g_{(w_k, \xi_k, \gamma_k, a_k)}(x) \tag{6.9}$$

Adım 3: $k < s$ ise $k=k+1$ yap ve Adım 5'e git.

Adım 4: $A_{l+1} = \left\{ a^i \in A_l : g_{k-1}(a^i) > 0 \right\}$, $l=l+1$, $I_k = \left\{ i \in I_k : a^i \in A_l \right\}$ yap.

Eğer $A_l \neq \varnothing$ ise Adım 2 'ye git.

Adım 5: *A* ve *B* kümelerini birbirinden ayıran *g(x)* fonksiyonunu aşağıdaki şekilde tanımla,

$$g(x) = \min_k g_k(x) \tag{6.10}$$

ve Dur.

Bu algoritma, ÇKF algoritması ile karşılaştırıldığında yapılan değişiklikler sırasıyla belirtilmiştir.

İlk olarak Adım 1 'de tepe noktalarının a^l koordinat seçimleri için ÇKF de olduğu gibi herbir nokta için DP problemi çözülmemiş bunun yerine *A* kümesine bir kümeleme algoritması (*k-ortalamalar*) uygulanarak bulunan herbir küme merkezi, a^l ($l=1,..,k$) koordinatı olarak seçilmiştir. Bu değişiklik bir önceki bölümde tanıtılan KMLM-ÇKF 1'de de mevcuttur.

İkinci değişiklik ise Adım 2 de yeralan (P_k) altprobleminde mevcuttur. Bu altproblemin ikinci kısıtında yer alan (6.8) eşitsizliğinde, *B* noktaları için

yapılan katı kısıtlamayı gevşetecek şekilde (z_j) karar değişkeni eklenmiş ve bu değişiklikle test doğruluk değerini arttırıp eğitim doğruluk değerini düşürerek aşırı uyumdan kurtulmak ve iyi bir genelleştirme sağlamak amaçlanmıştır.

Bu değişiklikle beraber (6.6) amaç fonksiyonunda da değişiklikler yapılmıştır. ÇKF'de yeralan DP probleminde sadece (y_i) hata değerleri ortalamasının enküçüklemesi yapılırken bu DP probleminde (y_i) ve (z_j) hata değerleri ortalaması toplamlarına enküçükleme işlemi yapılmaktadır. Ayrıca amaç fonksiyonunda yeralan $C>0$ değeri sabit bir ceza parametresidir. Bu değer küçük alındığında problemin çözümüyle oluşan çokyüzlü konik fonksiyonun altseviye kümelerinin çok sayıda *B* noktası içerme durumuna, yani hatalı konilere izin verilir, büyük alındığında ise tam tersi gerçekleşir yani oluşan koni içerisinde mümkün olduğunca az sayıda *B* noktası düşmesi istenmektedir ve hata toleransı azdır. Aşikardır ki, *C* değeri 0 alındığında hatalı durumlar amaç fonksiyonunu etkilemez 0 dan küçük negatif değerler alındığında ise problem amacından sapma gösterir ve yanlış çözümler elde edilir.

Bu algoritmada da KMLM-ÇKF 1 de olduğu gibi ayırıcı fonksiyon, "*k*" adet çokyüzlü konik fonksiyonun noktasal en küçüğü olacak şekilde (6.10) ile tanımlanmıştır.

6.2 Hesaplama Denemeleri

Sunulan KMLM-ÇKF 2'nin verimliliğini doğrulamak için gerçek hayattan iki sınıflı veri kümeleri üzerinde hesaplama denemeleri yapılmıştır. Uygulamada AMD Phenom(tm) II P820 üç-çekirdekli 1.80 GHz işlemcili bir dizüstü bilgisayar ve algoritma yazılımı için de MATLAB programı kullanılmıştır. Kullanılan 8 adet gerçek hayat veri kümesi "UCI machine learning repository" veri havuzundan alınmıştır. Veri setleri ile ilgili bilgiler aşağıda ifade edilmiş olup, veri setlerinin örnek ve özellik sayıları Çizelge 6.2'de belirtilmiştir.

Diabetes-Pima Hintli şeker hastaları veri kümesi: Sınıflar şeker hastası olanlar ve şeker hastası olmayanlar şeklinde tanımlanır. Pima şeker hastalığı veri kümesi, en az 21 yaşında olan ve Pima Hindistan kalıtımına sahip olan 768 bayan

hastanın verisini içermektedir. Belirlenen 8 özellik, her bir hastanın fiziksel özelliklerini belirtmektedir.

Ionosphere veri kümesi: Veritabanında bulunan veriler Labrador yarımada sında bulunan bir sistem tarafından toplanmıştır. Bu sistem, toplam 6.4 kilovatlık iletim gücü olan 16 tane yüksek frekanslı antenin bir dizisinden oluşmaktadır. Iyo nosferdeki serbest elektronlar bu çalışmanın hedefidir. Radarın “iyi" olarak verdiği dönütler iyonosferde bazı yapı tiplerinin kanıtıdır. “Kötü" dönütler ise sinyalleri iyonosferi geçip gittiği için herhangi bir yapının kanıtını göstermezler.

Kanıtları, titreşim süresi ve titreşim sayısı olan sinyaller bir otokorelasyon fonksiyonu kullanılarak işlenir. Bu sistem için 17 tane titreşim sayısı bulunmaktadır. Bu veritabanındaki her bir titreşim sayısı için karmaşık elektromanyetik sinyallerden oluşan fonksiyonlar ile elde edilen karmaşık değerlere karşı gelen 2 nitelik tanımlanmıştır.

Liver- BUPA karaciğer bozuklukları veri kümesi: BUPA veri kümesi altı tane sayısal niteliği olan 345 tane bekar erkeğe ilişkin verileri içermektedir. Niteliklerden beşi, karaciğer bozukluğu ile ilgili olduğu düşünülen kan testleri; diğeri de hergün içilen alkollü içecek sayısıdır.

WBCD- Wisconsin göğüs kanseri tanısı veri kümesi: Göğüs kanseri veri tabanında, Wisconsin Üniversitesi Hastanesi'nden elde edilen veriler yer almaktadır. Veri kümesinde iyi huylu ve kötü huylu olarak adlandırılan iki sınıf bulunmaktadır.

Heart- Kalp hastalığı veri kümesi: Cleveland klinikten Robert Detrano tarafından oluşturulan bu veri kümesi; her hasta için 13 farklı özellik olmak üzere 297 hastadan alınan veriler ile oluşturulmuştur. Hastalara ait özellikler; yaş, cinsiyet, göğüs ağrısı tipi, tansiyon gibi bilgilerin yanısıra kan ve efor testlerinden elde edilmiştir. Sınıflar “hasta" veya “sağlıklı" olmak üzere ikiye ayrılmaktadır.

WBCP- Wisconsin göğüs kanseri tedavi süreci veri kümesi: Wisconsin Üniversitesi Hastanesi'nden elde edilen göğüs kanseri hastalarını iki yıl boyunca

tedavi süreçlerinin izlenmesi ile ilgili veriler yer almaktadır. Veri kümesindeki iki sınıf tedaviye olumlu yanıt verenler ile vermeyenleri göstermektedir.

Connectionist Bench- Sonar, mayınlar ve kayalar veri kümesi: Bu veri kümesinde 111 adet "sonar mayınları" örnekleri mevcuttur. Çeşitli koşullar altında ve çeşitli açılarda metal silindirin sıçraması ile elde edilmiştir. "sonar kayaları" ise 97 adettir. Bu örnekler benzer koşullar altındaki kayalardan elde edlimiştir.

Herbir örnek 0.0-1.0 aralığında 60 adet sayıdan oluşur. Herbir sayı belli bir sürede entegre olmuş özel frekanslı bir şeridin enerjisini belirtir. Daha yüksek frekanslar için entegrasyon açıklığı sonraki zamanlarda ortaya çıkar çünkü bu frekanslar cırılıtı süresince iletilemez.

Sınıf etiketleri sırasıyla mayın ve kaya için "M", "R" ile temsil edilir.

Haberman's Survival veri kümesi: Bu veri kümesi göğüs kanser tedavisi için ameliyat olan hastaların hayatta kalıp kalmama durumlarını içerir.

Veriler 1958-1970 yılları arasında Chicago'nun Billings Hastanesinde göğüs kanser tedavisi için ameliyat olan hastalardan elde edilmiştir.

Çizelge 6.3. Veri kümelerine ait örnek ve özellik sayıları

Veri Kümeleri	Örnek Sayısı	Özellik sayısı
Diabetes	768	8
Ionosphere	351	34
Liver	345	6
WBCD	683	9
Heart	297	13
WBCP	194	32
Connectionist Bench	208	60
Haberman	306	3

Çizelge 6.4'de KMLM-ÇKF 2 ve ÇKF algoritması arasında süre ve doğruluk açısından karşılaştırma yapmak amaçlı sonuçlar sunulmuştur. Burada doğruluk, eğitim küme eleman sayısı ile doğru sınıflandırılan nokta sayısı arası oranı vermektedir. Bu oran aşağıdaki şekilde hesaplanır.

ds= doğru sınıflandırılan eleman sayısı,

es= eğitim kümesi eleman sayısı,

olarak tanımlandığında, doğruluk= $\frac{100 \times ds}{es}$.

Kullanılan KMLM-ÇKF 2'de, başlangıçta kümeleme yöntemi için *k-ortalamalar* yöntemi kullanılmıştır. *k* (1-20) en yüksek doğruluk değerini alacak şekilde tanımlanmıştır ve *C>0* penaltı sabiti ise $b^j \in B$ noktalarına $a^i \in A$ noktalarından daha az hata toleransı vermek amaçlı *C=10* şeklinde tanımlanmıştır.

Görüldüğü üzere doğruluk oran değeri ÇKF algoritmasında olduğu gibi 100 değildir. Bunun sebebi $b^j \in B$ noktalarında hataya izin verme ve *k.* (kümelemede belirlenen küme sayısı) iterasyonda algoritmayı sonlandırmadır.

Süre açısından bakıldığında KMLM-ÇKF 2 algoritmasının verimliliği göze çarpmaktadır. Bunun sebebi ÇKF'de her iterasyonda tepe noktası koordiantının bulunması için tüm noktalarda DP probleminin çözümlenmesi gerekirken KMLM-ÇKF 2'de sadece başlangıçta kümelemeyle bulunan merkezlerin seçilerek DP probleminin bu noktalar için uygulanmasının yeterli olmasıdır.

Çizelge 6.6'da aynı karşılaştırmalar 8 adet sentetik veri kümesi üzerinde yapılmıştır. Bu sentetik veri kümelerinin örnek ve özellik sayıları Çizelge 6.5'de belirtilmiştir. Elde edilen sonuçlar doğrultusunda sentetik veri kümelerinde, gerçek hayat veri kümelerinden daha iyi sonuçlar elde edilerek herbirinde de hem %100 lük doğruluk sağlanmış hem de süre kısalmıştır.

Çizelge 6.4. KMLM-ÇKF 2 ve ÇKF algoritması uygulama sonuçları

	KMLM-ÇKF 2		ÇKF	
Veri Kümesi	Doğruluk %	Süre sn	Doğruluk %	Süre sn
Diabetes	74.73	69 sn.	100	2400 sn.
Ionosphere	98.86	42 sn.	100	756 sn.
Liver	70.14	10 sn.	100	2320 sn.
WBCD	97.28	35 sn.	100	1763 sn.
Heart	85.90	22 sn.	100	613 sn.
WBCP	72.90	30 sn.	100	299 sn.
Connectionist Bench	83.80	84 sn.	100	287 sn.
Haberman	68.00	126 sn.	100	2655 sn.

Çizelge 6.5. Sentetik veri kümeleri örnek ve özellik sayıları

	örnek sayısı	özellik sayısı
veri kümesi1	20	6
veri kümesi2	20	6
veri kümesi3	50	6
veri kümesi4	50	6
veri kümesi5	100	6
veri kümesi6	100	6
veri kümesi7	300	6
veri kümesi8	300	6

Çizelge 6.6. Sentetik veri kümeleri için KMLM-ÇKF 2 ve ÇKF sonuçları

	ÇKF		KMLM-ÇKF 2	
	Doğruluk	Süre(sn)	Doğruluk	Süre(sn)
Veri kümesi1	100	0.38	100	0.16
Veri kümesi2	100	0.54	100	0.06
Veri kümesi3	100	1.70	100	0.11
Veri kümesi4	100	1.41	100	0.11
Veri kümesi5	100	6.72	100	2.76
Veri kümesi6	100	8.21	100	0.2
Veri kümesi7	100	211.09	100	0.92
Veri kümesi8	100	231.11	100	1.077

Aynı zamanda veri kümelerine, sınıflandırma fonksiyonlarının geçerliliğini test etme amaçlı *10* kez çapraz doğrulama testleri uygulanmış ve ÇKF algoritması ile karşılaştırma yapılmıştır.

k-kez çapraz doğrulamada veri kümesi *D*, eşit sayıda veri içeren *k* altkümeye $D_1,D_2,..,D_k$ ayrılır. *k* kez eğitim ve test etme işlemi uygulanır öyle ki her defasında, $t \in \{1,2,...,10\}$, $D \backslash D_t$ üzerinde eğitim yapılır ve D_t üzerinde test edilir. Çapraz doğrulama doğruluk değeri ise doğru sınıflandırılan veri sayısının veri kümesi eleman sayısına bölümüyle elde edilir. Çoğu uygulamada *k=10* tercih edilir (Kohavi, 1995).

İstenilen, aşırı uyum (over fitting) dan kurtulmak yani eğitim ve test doğruluk değerleri arası farkı azaltmaktır. Çünkü sınıflandırmada asıl amaç sınıfları bilinen ve eğitimde kullanılan verileri doğru sınıflandırmak değil gelecekte karşılaşılacak benzer verileri en iyi şekilde sınıflandırmaktır.

Çizelge 6.7. Gerçek hayat veri kümelerinde 10-kez çapraz doğrulama sonuçları

	ÇKF		KMLM-ÇKF 2 *k-means*			KMLM-ÇKF 2- *k-medoid*		
Veri kümeleri	Eğitim başarısı	Test başarısı	Eğitim başarısı	Test başarısı	k	Eğitim başarısı	Test başarısı	k
Ionosphere	100	95.73	98.10	94.28	10	96.83	97.14	10
			98.10	94.28	20	98.10	94.44	20
Liver	100	77.97	73.63	73.52	10	72.95	73.52	10
			76.12	68.57	20	72.25	68.57	20
WBCD	100	100	97.72	100	10	98.37	100	10
			98.21	98.55	20	98.21	100	20
Diabetes	100	80.47	65.55	70.12	10	67.58	68.83	10
			70.07	77.63	20	73.22	79.22	20
Heart	100	86.95	86.51	88.00	10	86.57	80.44	10
			88.88	91.66	20	87.09	86.95	20
WBCP	100	84.21	73.71	63.15	10	62.85	57.89	10
			84.00	89.47	20	71.42	78.94	20
Connectionist Bench	100	80.38	90.37	85.71	10	87.70	90.47	10
			100	96.79	20	95.21	95.00	20
Haberman	100	65.50	53.98	46.66	10	57.60	56.66	10
			69.66	70.00	20	66.30	63.33	20

Çizelge 6.7'de gerçek hayat veri kümeleri üzerinde uygulanan 10 kez çapraz doğrulama sonuçları sunulmuştur. KMLM-ÇKF 2 uygulamalarında *k-ortalamalar* ve *k-medyan, k=10* ve *k=20* değerleri için ayrı ayrı uygulanmıştır.

Çizelge 6.7'den de görüldüğü üzere Ionosphere, Heart, WBCP ve Connectionist Bench veri kümelerinde 10 kez çaprazlama başarısında daha verimli sonuçlar elde edilmiştir.

Gerçek hayat veri kümelerinin yanısıra sentetik veri kümelerine de 10 kez çaprazlama testi uygulanmıştır sonuçlar Çizelge 6.8 de belirtilmiştir. Sentetik veri kümelerinde, gerçek hayat veri kümelerine istinaden verimlilik daha net görülmektedir.

Çizelge 6.8. Sentetik veri kümelerinde 10-kez çapraz doğrulama sonuçları

	ÇKF		KMLM-ÇKF 2	
	Eğitim başarısı	Test başarısı	Eğitim başarısı	Test başarısı
Verikümesi1	100	90	100	95
Verikümesi2	100	90	100	100
Verikümesi3	100	96	100	100
Verikümesi4	100	100	100	100
Verikümesi5	100	100	100	100
Verikümesi6	100	100	100	100
Verikümesi7	100	100	100	100
Verikümesi8	100	100	100	100

7. ÇOKLU SINIFLANDIRMADA ÇOKYÜZLÜ KONİK FONKSİYONLAR VE KÜMELEME TABANLI SINIFLANDIRMA YÖNTEMİNİN KULLANILMASI

Önceki bölümlerde ikili sınflandırma problemleri ele alınmıştı. Bu bölümde ise çoklu sınıflandırma problemleri ele alınmış ve ikili sınıflandırma için Bölüm 6'da geliştirilen yöntem genişletilerek çoklu sınıflandırma problem çözümlerinde kullanılmıştır.

7.1 Çoklu Sınıflandırma Problemleri

Çoklu sınıflandırmada $A = A_1 \cup A_2 ... \cup A_n, n > 2, n \in R$ veri kümesi "*n*" farklı sınıftan oluşmaktadır. Çoklu sınıflandırma problemlerine örnek olarak optik karakter tanıma, hastalık teşhisi, ses tanıma vb. problemler verilebilir. Çoklu sınıflandırıcıların oluşturulması için sinir ağları (Ou and Murphey, 2007), matematiksel programlama (Bennett and Mangasarian, 1994) ve destek vektör makinaları (Bredensteiner and Bennett, 1999) kullanılmaktadır. İkili sınıflandırma ile çoklu sınıflandırma arasındaki tek fark sınıf sayısının ikiden büyük olmasıdır ve bu nedenle ikili sınıflandırma yöntemlerini çoklu sınıflandırma problemlerine uyarlamak mümkündür.

İkili sınıflandırma yöntemlerinin çoklu sınıflandırmaya genişletilmesi için bire karşı bir, bire karşı hepsi, yönlü çevrimsiz serim veya ardışık sınıflandırma yaklaşımları kullanılır (Öztürk, 2010). Bu yaklaşımlardan en çok kullanılanlar bire karşı bir ve bire karşı hepsi yaklaşımlarıdır. Bire karşı bir yaklaşımında her bir sınıf ikilisi için ikili sınıflandırma uygulanır. Örneğin sınıf sayısı *n* olduğunda sınıflandırıcı sayısı, *n(n-1)/2* ile belirlenir. Bire karşı hepsi yaklaşımında ise bir sınıf ile geri kalan tüm *n-1* sınıfın birleşimi arasında ikili sınıflandırma uygulanır ve sınıf sayısı kadar sınıflandırıcı oluşturulur (Tax and Duin, 2002).

Tezde Bölüm 6'da geliştirilen ve ikili sınıflandırmada kullanılan çokyüzlü konik fonksiyonlar ve kümeleme yönteminin kullanıldığı algoritma bire karşı hepsi yaklaşımı ile genişletilerek çoklu sınıflandırma problemlerine uygulanmıştır.

7.2 Çokyüzlü Konik Fonksiyonlar ve Kümeleme Tabanlı Yöntem ile Çoklu Sınıflandırma

Çoklu sınıflandırma problem çözümlerinde kullanılacak algoritma için Bölüm 6'da geliştirilen KMLM-ÇKF 1 temel alınmıştır.

Bire karşı hepsi yaklaşımına benzer şekilde uyarlanan algoritmada herbir sınıf ile geri kalan tüm sınıflar arası ikili sınıflandırma uygulanmaktadır ancak kümeleme yönteminin kullanımı sebebiyle sınıflandırıcı sayısı, sınıf sayısı "*n*" ile değil "*k.n*" ile belirlenir. Burada "*k"* algoritma başlangıcında kümeleme için belirlenen küme sayısını temsil eder.

Herbir iterasyonda ikili sınıflandırma A_j, $j=1,2,...,n$ ile $A \backslash A_j$ arasında yapılır ve "k" adet $g_j^{1,...,k}(.)$ ayırıcı fonksiyon elde edilir. Test aşamasında bir "a" noktasının hangi sınıfa ait olduğunun belirlenmesinde üç farklı durumla karşılaşılmaktadır. Test noktası, sınıflandırıcılardan sadece biri ile sınıflandırılabilir, birden fazlası ile sınıflandırılabilir veya hiçbiri ile sınıflandırılamaz. Bu durumlarla karşılaşıldığında Ou ve Murphey (Ou and Murphey, 2007) farklı yaklaşımların kullanılabileceğini belirtmişlerdir.

Bu tezde algoritma sonucu elde edilen sınıflandırıcılar kullanılarak verilen bir "a" noktasının hangi sınıfa atanacağı aşağıdaki şekilde belirlenir:

$$j = \arg\min g_j^{1,...,k}(a).$$

Dolayısıyla ayırıcı fonksiyon ikili sınıflandırmalar sonrası elde edilen tüm fonksiyonların noktasal en küçüğü olarak tanımlanır:

$$g(x) = \min_j g_j^{1,...,k}(x),\ j = 1,...,n.$$

Çoklu sınıflandırma için KMLM-ÇKF 1 genişletilerek oluşturulan sözde algoritma aşağıda belirtilmiştir.

ALGORİTMA 4: **Çoklu-KMLM-ÇKF**

A, $\mathbb{R}^n$ de tanımlı ve "*c*" sınıflı bir küme olsun.

Başlangıç Adımı: $A = A_1 \cup A_2 \cup ... \cup A_c$, $A = \left\{ a_l^i \in \mathbb{R}^n : i \in I_l, l = 1,...,c \right\}$ l=1 yap.

Adım 1: $B = A / A_l$, $B = \left\{ b_l^j \in \mathbb{R}^n : j \in I / I_l \right\}$

Adım 2: A_l kümesinde kümeleme algoritmasını uygula. Küme(cluster) sayısı (k) olsun. s=1 , $I_l^1 = I_l$, $A_l^1 = A_l$ yap.

Adım 3: $a_s \in A_l$, A_l in s. orta noktası olsun. P_l^s altproblemini çöz.

$$(P_l^s) \quad \min\left(\frac{y^{'} e_{|I_l^s|}}{\left|I_l^s\right|}\right) \tag{7.1}$$

$$w'(a^i - a_s) + \xi \left\| a^i - a_s \right\|_1 - \gamma + 1 \le y_i, \quad \forall i \in I_l^s \tag{7.2}$$

$$-w'(b_l^j - a_s) + \xi \left\| b_l^j - a_s \right\|_1 + \gamma + 1 \le 0, \quad \forall j \in I / I_l \tag{7.3}$$

$$y = (y_1,..., y_{I_l^s}) \in \mathbb{R}_+^{I_l^s}, w \in \mathbb{R}^n, \xi \in \mathbb{R}, \gamma \ge 1$$

$w_s, \xi_s, \gamma_s, y_s$, (P_l^s) ' in bir çözümü olsun.

$$g_l^s(x) = g_{(w_s, \xi_s, \gamma_s, a_s)}(x) \tag{7.4}$$

Adım 4: $s < k$ ise s=s+1 , $A_l^s = \left\{ a^i \in A_l^{s-1} : g_l^{s-1}(a^i) > 0 \right\}$, $I_l^s = \left\{ i \in I_l^s : a^i \in A_l^s \right\}$ yap ve Adım 3'e git.

Adım 5: l<c ise l=l+1 yap ve Adım 2' ye git.

Adım 6: A_l, l=1,...,c kümelerini birbirinden ayıran $g(x)$ fonksiyonunu aşağıdaki şekilde tanımla,

$$g(x) = \min_{l} g_l^{1,\ldots,k}(x)$$

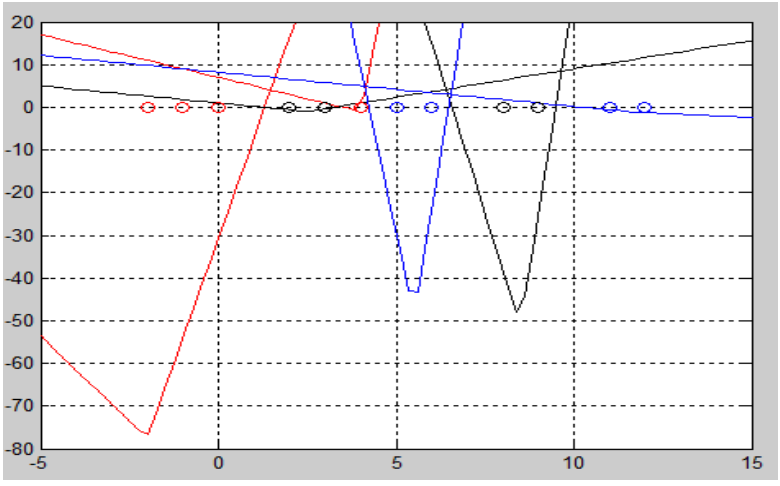
ve Dur.

Çizelge 7.1: Veri kümesi değerleri

kırmızı	siyah	mavi
-2	2	5
-1	3	6
0	8	11
4	9	12

Çizelge 7.2: Hesaplanan ÇKF parametre değerleri

g(x)	*a*	*w*	ξ	γ
1.	4	18.8324	20.8324	1
2.	-2	7.7376	-18,5875	77.4006
3.	2,5	0.2667	15.6637	1
4.	8,5	12.5785	38.9016	51.0646
5.	11,5	0.6007	0.1993	1
6.	5,5	5.8987	44.3293	48.6761



Şekil 7.1. *2*-boyutlu uzayda çoklu sınıflandırma örneği

Şekil 7.1'de konunun anlaşılması için *2*-boyutlu uzayda basit bir çoklu sınıflandırma probleminin çözümünün grafiksel gösterimi yapılmıştır. Kırmızı, siyah ve mavi olmak üzere 3 farklı sınıf vardır. Bu sınıf verileri, Çizelge 7.1 de belirtilmiştir. Herbir sınıf için algoritma sonrası oluşturulan ayırıcı fonksiyonlar yine sınıfıyla ilişkili kırmızı siyah ve mavi renkler kullanılarak çizilmiştir. Çizelge 7.2 de Çoklu-KMLM-ÇKF sonrası hesaplanan bu 6 çokyüzlü konik fonksiyonun parametre değerleri belirtilmiştir. Görüldüğü üzere herbir sınıf için iki ayırıcı fonksiyon kullanılmıştır bunun sebebi algoritma başlangıcında kümeleme yöntemi için küme sayısı, *k*'nın 2 olarak belirlenmesidir.

7.3 Hesaplama Denemeleri

Sunulan Çoklu KMLM-ÇKF, gerçek hayattan beş farklı çok-sınıflı veri kümesine uygulanmıştır. Uygulamada AMD Phenom(tm) II P820 üç-çekirdekli 1.80 GHz işlemcili bir dizüstü bilgisayar ve algoritma yazılımı için de MATLAB programı kullanılmıştır. Kullanılan 6 adet gerçek hayat veri kümesi "UCI machine learning repository" veri havuzundan alınmıştır. İlgili bilgiler aşağıda verilmiş, örnek ve özellik sayıları Çizelge 7.3'de belirtilmiştir.

Algoritmada kümeleme yöntemi kullanımının sağladığı verimliliği görmek amaçlı, Bölüm 4'te incelenen ÇKF algoritmasına bire karşı hepsi yaklaşımı uygulanmış ve elde edilen algoritma Çoklu-ÇKF, Çoklu-KMLM-ÇKF ile karşılaştırılmıştır.

İris Plant veri kümesi: Veri kümesi her birinde 50 örnek olan 3 sınıftan oluşmaktadır. Her bir sınıf, bir iris bitkisi tipini belirtir. Bir sınıf, kalan sınıflardan ikisiyle doğrusal ayrılabilirdir ancak geri kalan sınıflar kendi aralarında doğrusal ayrılabilir değildir.

Glass tanımlama veri kümesi: Veri kümesi 6 farklı tipten oluşmaktadır. Camın 10 özelliği belirlenmiştir. Cam türlerinin sınıflandırıldığı bu çalışma kriminolojik araştırmalarda kullanılmaktadır.

Wine tanıma veri kümesi: Bu veritabanında bulunan veriler, İtalya'nın aynı bölgesinde yetişen ancak üç farklı biçimde ekilen üzümlerden elde edilen şarapların kimyasal bir analizinin sonuçlarıdır.

Vowel tanıma veri kümesi: Bu veritabanında tüm sesli harfler 0-10 ile belirtilmiştir. Toplamda 15 kişi bir sesli harfi 6 defa tekrarlamış ve konuşmacılar 0-89 numaralandırılmıştır.

Vehicle tanıma veri kümesi: Amaç verilen bir araç görüntüsünden alınan belli özellikler ile 4 farklı tip araçtan hangi sınıfa ait olduğunu bulmaktır.

Car veri kümesi: Car evaluation veri kümesi 1728 örnek ve 4 sınıf ve 6 özellikten oluşmaktadır. Sınıflar kabuledilemez-kabuledilebilir-iyi-çokiyi şeklindedir. Özellikler ise alım-bakım-kapılar-kişiler-çekiş gücü-güvenlik ile ilgilidir.

Çizelge 7.3'de veri kümelerinin özellik sınıf ve örnek sayıları belirtilmiştir. Çizelge 7.4'de Çoklu-KMLM-ÇKF uygulanması sonucunda elde edilen doğruluk ve hesaplama süresi değerleri verilmiştir. Çizelge 7.5'de ise 10 kez çapraz doğrulama sonuçları sunulmuştur. Hesaplama süresi 1800 sn. yi geçen denemelerde algoritma sonlandırılmış ve hücrede (_) işareti ile belirtilmiştir.

Çizelge 7.3: Çoklu sınıf veri kümelerine ait örnek ve özellik sayıları

Veri Kümeleri	Örnek Sayısı	Özellik sayısı	Sınıf sayısı
İris	150	4	3
Glass	214	10	6
Wine	178	13	3
Vowel	528	10	11
Vehicle	94	18	4
Car	1728	6	4

Elde edilen sonuçlar doğrultusunda, ÇKF de kümeleme yönteminin kullanımıyla çoklu sınıflandırma hesaplama süresinde önemli miktarda düşüş gözlenmiştir. Aynı zamanda 10 kez çaprazlama uygulaması sonrası bazı veri kümelerinde eğitim başarımında düşüş ve test başarım değerlerinde de artış elde edilerek eğitim verisine aşırı-uyum (over fitting) sorunu ortadan kaldırılmıştır.

Çizelge 7.4 Çoklu-KMLM-ÇKF ve Çoklu-ÇKF uygulama sonuçları

	Çoklu-KMLM-ÇKF		Çoklu-ÇKF	
Veri Kümesi	Doğruluk %	Süre sn	Doğruluk %	Süre sn
İris	98.66	1.96	100	29.87
Glass	100	2.71	100	100
Wine	100	0.79	100	39.11
Vowel	71.96	291.51	_	_
Vehicle	88	0.78	100	12.503
Car	71.06	473.64	_	_

Çizelge 7.5 Çoklu-KMLM-ÇKF ve Çoklu-ÇKF için 10-kez çapraz doğrulama sonuçları

	Çoklu-KMLM-ÇKF		ÇKF	
Veri Kümesi	eğitim	test	eğitim	test
İris	98	94	100	92
Glass	100	93	100	93
Wine	100	98	100	99
Vowel	36	34	_	_
Vehicle	87	69	100	62
Car	69.97	69.90	_	_

8. DESTEK VEKTÖR MAKİNALARINDA ÇOKYÜZLÜ KONİK FONKSİYONLAR İLE İKİLİ SINIFLANDIRMA

Bu bölümde çok yüzlü konik fonksiyonlarla ikili sınıflandırma ile bölüm 3.2.5 de anlatılan Destek Vektör Makinaları nın bir kombinasyonu yapılarak destek vektör makinalarında çokyüzlü konik fonksiyonlarla ikili sınıflandırma konusu çalışılmıştır. Aynı zamanda verimliliği arttırma amaçlı kümeleme yöntemi kullanılmıştır. Böyle bir yaklaşımın temel amacı destek vertör makinalarında kullanılan çekirdek fonksiyon kullanımına önemli bir alternatif sağlamaktır.

8.1'de, destek vektör makinalarında doğrular yada hiperdüzlemler yerine, çokyüzlü konik fonksiyonların kullanıldığı benzer bir ayrım yöntemi geliştirilmiştir. 8.2'de ise çokyüzlü konik fonksiyon ile ayrım sağlanamayan durumlar için bir hata fonksiyonu tanımlanmış ve bu hata fonksiyonunun minimizasyonunu içeren bir algoritma sunulmuştur 8.3'te yapılan çalışmalar açıklayıcı örneklerle açıklanmıştır ve bu algoritmaların verimliliğini arttırma amaçlı tepe noktası seçimlerinde kümeleme yönteminin uygulandığı başka bir algoritma daha tanımlanmıştır. ve 8.4'de gerçek hayat veri kümeleri üzerinde tasarlanan algoritmalar uygulanarak sonuçlar tablolarla belirtilmiştir.

8.1. Çokyüzlü konik fonksiyon ile ayrılabilme durumu

A ve B, $\mathbb{R}^n$ de tanımlı iki küme olsun.

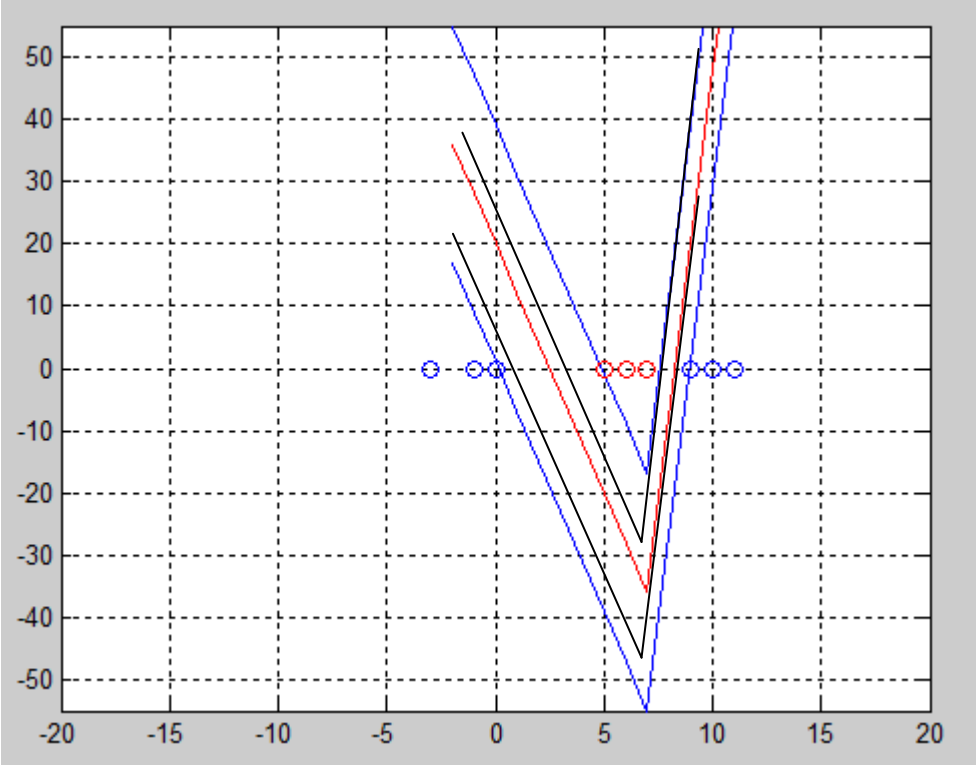
$$A=\left\{a^i \in \mathbb{R}^n : i \in I\right\}, B=\left\{b^j \in \mathbb{R}^n : j \in J\right\}$$

burada $i=\{1,..,m\}$, $j=\{1,..,p\}$ dir.

Önerme 8.1: A ve B kümeleri çokyüzlü konik fonksiyon ile ayrılabilirdir eğer en az bir $g(x)=g_{(w,\xi,\gamma,a)}(x)$ çokyüzlü konik fonksiyonu var ise öyle ki ;

$$g(a^i) \le 0 \quad \forall a^i \in A$$
$$g(b^j) > 0 \quad \forall b^j \in B$$
olsun.

Şekil 8.1 üzerinde görüldüğü gibi veri kümeleri, $a, w \in \mathbb{R}^n$, $\xi \in \mathbb{R}$ parametreleri sabit, sadece $\gamma \in [1, \infty)$ parametresi farklı (dolayısıyla $(a, -\gamma) \in \mathbb{R}^n \times \mathbb{R}$ tepe noktası farklı) çok sayıda çokyüzlü konik fonksiyon ile ayrılabilmektedir. Ancak daha iyi genelleştirme için örneklerin çokyüzlü konik fonksiyonlardan belli bir mesafe uzakta olmasıda istenir. Bu uzaklık, marjin olarak adlandırılır ve en büyüklenmeye çalışılır. En iyi ayırıcı çokyüzlü konik fonksiyon marjini en büyükleyen çokyüzlü konik fonksiyondur. Basit bir tanımla ayrım için veriyi birbirinden ayıran bu çokyüzlü konik fonksiyonlardan aralarında en geniş mesafeye sahip olanları seçmek en uygun yoldur.



Şekil 8.1. Veri kümesi ve çok yüzlü konik fonksiyonlar

Açıklayıcı örnek Şekil 8.3 üzerinde yer alan $g_1(x)$ ve $g_2(x)$ çokyüzlü konik fonksiyonları aralarında en büyük boşluğa sahip olanlardır. Bu iki çokyüzlü koniğin ortasında bulunan $g(x)$ çokyüzlü konik fonksiyonu ise iki sınıf veriyi

birbirinden ayıran çokyüzlü konik fonksiyondur. Bu $g(x)$ çokyüzlü konik fonksiyonuna "**optimal ayıran çokyüzlü konik fonksiyon"** adı verilir.

Burada $g_1(x)$ ve $g_2(x)$ çokyüzlü konikleri göz önüne alındığında, bu çokyüzlü konikler üzerindeki gözlemler "**destek vektör**" adını alır. Yani; $g_1(x) = 0, x \in A$ ve $g_2(y) = 0, y \in B$ vektörleri destek vektörlerdir.

Ele alınan çokyüzlü konik fonksiyonlar, aşağıdaki şekilde ifade edilir :

$$g_1(x) = w(x-a) + \xi \left\| x-a \right\|_1 - \gamma_1$$

$$g_2(x) = w(x-a) + \xi \left\| x-a \right\|_1 - \gamma_2$$

$$g(x) = w(x-a) + \xi \left\| x-a \right\|_1 - \gamma$$

İki fonksiyonun tepe noktaları $(a, -\gamma_1), (a, -\gamma_2)$, arası öklid uzaklığı $\gamma_2 > \gamma_1$ varsayılarak aşağıdaki şekilde hesaplanabilir:

$$\sqrt{(a-a)^2 + (-\gamma_2 + \gamma_1)^2} = \left| -\gamma_2 + \gamma_1 \right| = \gamma_2 - \gamma_1$$

Sonuç olarak elde edilen minimizasyon problem modeli aşağıda belirtilmiştir:

$$\min -(\gamma_2 - \gamma_1) \tag{8.1}$$

$$w(a^i - a) + \xi \left\| a^i - a \right\|_1 - \gamma_1 \le -1, \quad i = 1,..,m,$$

$$w(b^j - a) + \xi \left\| b^j - a \right\|_1 - \gamma_2 \ge 1, \quad j = 1,..,p,$$

$$\gamma_2 - \gamma_1 \ge 0,$$

$$w \in \mathbb{R}^n, \xi \in \mathbb{R}, \gamma_{1,2} \ge 1,$$

(8.1) minimizasyon problem çözümüyle elde edilen $w \in \mathbb{R}^n, \xi \in \mathbb{R}$, $\gamma_{1,2} \in [1, \infty)$ parametreleriyle oluşturulan "**optimal ayıran çokyüzlü konik fonksiyon"** aşağıdaki şekilde belirtilir:

$$g(x) = w(x-a) + \xi \left\| x-a \right\|_1 - (\frac{\gamma_1 + \gamma_2}{2})$$

Algoritma başlangıç adımında daha verimli sonuçlar elde etmek adına keyfi bir "*a"* merkez noktası seçmek yerine herbir "*a*" noktasına (8.1) minimizasyon problemi uygulanarak en küçük değerin elde edildiği "*a*" noktası merkez nokta olarak seçilmiştir.

Algoritma 5**: DVM-ÇKF 1**

A ve *B*, $\mathbb{R}^n$ de tanımlı iki küme olsun.

$$A = \left\{ a^i \in \mathbb{R}^n : i \in I \right\}, B = \left\{ b^j \in \mathbb{R}^n : j \in J \right\}$$

burada $i = \{1,..,m\}$, $j = \{1,..,p\}$ dir.

Başlangıç Adımı: Her $a_i \in A$ noktası için (P) altproblemini çöz.

Minimum (*P*) problem çözümüne ait a_i noktasını seç.

Adım 1: P altproblemini çöz.

$$(P) \quad \min -(\gamma_2 - \gamma_1)$$

$$w(a^i - a) + \xi \left\| a^i - a \right\|_1 - \gamma_1 \le -1, \quad i = 1,..,m,$$

$$w(b^j - a) + \xi \left\| b^j - a \right\|_1 - \gamma_2 \ge 1, \quad j = 1,..,p,$$

$$\gamma_2 - \gamma_1 \geq 0,$$

$$w \in \mathbb{R}^n, \xi \in \mathbb{R}, \gamma_{1,2} \geq 1,$$

$w, \xi, \gamma_1, \gamma_2$, P 'nin bir çözümü olsun.

$\gamma = \frac{\gamma_2 + \gamma_1}{2}$ yap.

Adım 2: A ve B kümelerini birbirinden ayıran $g(x)$ fonksiyonunu aşağıdaki şekilde tanımla,

$$g(x) = g_{(w,\xi,\gamma,a)}(x)$$

8.2. Çokyüzlü Konik Fonksiyon İle Ayrılamama Durumu

Verilerin çokyüzlü konik fonksiyon ile ayrılamama durumunda negatif olmayan ve hataları ifade eden $\varepsilon_{i,j}$ gevşek değişkenlerinin optimizasyon modeline eklenmesi sağlanarak soruna çözüm aranmıştır. Aynı algoritma normalde ayrımın sağlanabildiği kümelere de uygulanabilir, bu durumda tüm hata değerleri için $\varepsilon_i, \varepsilon_j = 0, i = 1,..m,\ j = 1,.., p$ sağlanır.

$$w(a^i - a) + \xi \left\| a^i - a \right\|_1 - \gamma_1 \leq -1 + \varepsilon_i, \quad i = 1,.., m,$$
$$w(b^j - a) + \xi \left\| b^j - a \right\|_1 - \gamma_2 \geq 1 - \varepsilon_j, \quad j = 1,.., p,$$

Burada $\varepsilon_i, \varepsilon_j > 0$ olan a^i ve b^j indisli veriler sırasıyla g_1 ve g_2 çokyüzlü konik fonksiyonların ters tarafında kalan yani yanlış sınıflandırılan verilerdir

Algoritma 6: **DVM-ÇKF 2**

A ve B, $\mathbb{R}^n$ de tanımlı iki küme olsun.

$$A = \left\{ a^i \in \mathbb{R}^n : i \in I \right\}, B = \left\{ b^j \in \mathbb{R}^n : j \in J \right\}$$

burada $i=\{1,..,m\}$, $j=\{1,..,p\}$ dir.

Başlangıç Adımı: Her $a_i \in A$ noktası için (P) altproblemini çöz.

Minimum (P) problem çözümüne ait a_i noktasını seç.

Adım 1: P altproblemini çöz.

$$(P) \quad \min(-(\gamma_2-\gamma_1)+\sum_{i=1}^{m}\varepsilon_i+\sum_{j=1}^{p}\varepsilon_j)$$

$$w(a^i-a)+\xi\left\|a^i-a\right\|_1-\gamma_1\le -1+\varepsilon_i, \quad i=1,..,m,$$

$$w(b^j-a)+\xi\left\|b^j-a\right\|_1-\gamma_2\ge 1-\varepsilon_j, \quad j=1,..,p,$$

$$\gamma_2-\gamma_1\ge 0,$$

$$\varepsilon_{i,j}\ge 0, w\in\mathbb{R}^n, \xi\in\mathbb{R}, \gamma_{1,2}\ge 1,$$

$w,\xi,\gamma_1,\gamma_2,\varepsilon_i,\varepsilon_j$,$i=1,..,m, j=1,...,p$, (P) 'nin bir çözümü olsun.

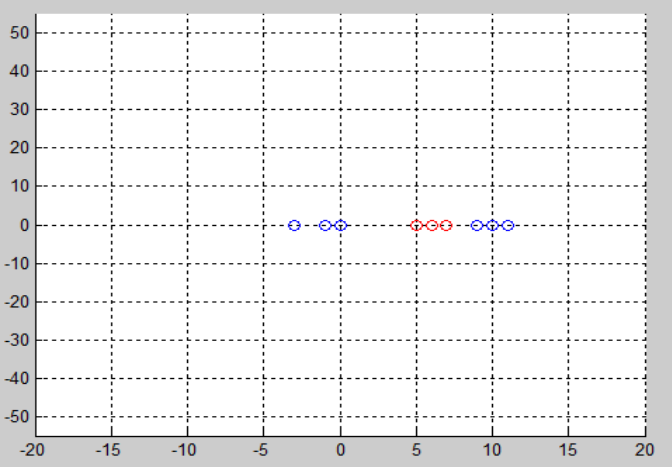
$\gamma=\frac{\gamma_2+\gamma_1}{2}$ yap.

Adım 2: A ve B kümelerini birbirinden ayıran $g(x)$ fonksiyonunu aşağıdaki şekilde tanımla,

$$g(x)=g_{(w,\xi,\gamma,a)}(x)$$

8.3 Açıklayıcı Örnekler

Örnek 8.1: $\mathbb{R}$'de tanımlı $A=\{5,6,7\}$ ve $B=\{-3,-1,0,9,10,11\}$ kümeleri ele alınsın. Şekil 8.2' de A noktaları kırmızı B noktaları mavi ile belirtilmiştir ve görüldüğü üzere A ve B noktaları arasında doğrusal bir ayrımın sağlanamayacağı açıktır .



Şekil 8.2: Veri kümesinin 2-boyutlu uzayda gösterimi

Burada A kümesini B kümesinden ayırmak için MATLAB ortamında DVM-ÇKF 2 algoritması uygulandığında aşağıdaki sonuçlar elde edilmiştir.

Algoritma merkez nokta olarak " $a=7$ " yi seçmiştir.

Algoritma içerisinde (P) minimizasyon problem çözümüyle elde edilen değerler:

$$w=10, \xi=18, \gamma_1=17, \gamma_2=55.$$

S_0 altseviye kümesi A kümesi elemanlarını kapsayacak şekilde, $a_i \in S_0, g_1(a_i) \le 0$, belirlenen çokyüzlü konik fonksiyon:

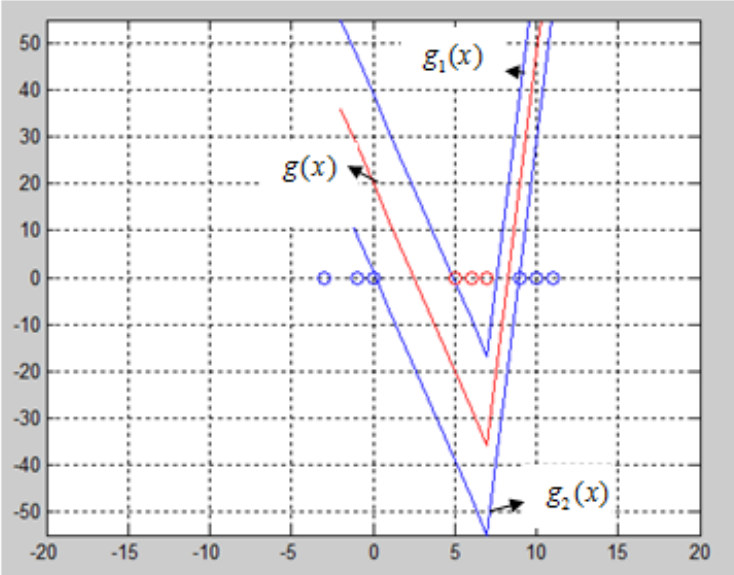
$$g_1(x)=10(x-7)+18\|x-7\|-17$$

S_0 altseviye kümesi B kümesi elemanlarını dışarıda bırakacak şekilde, $b_j \in S_0, g_2(b_j) \ge 0$, belirlenen çokyüzlü konik fonksiyon:

$$g_2(x) = 10(x-7) + 18\|x-7\| - 55$$

Elde edilen "**Optimal ayırıcı çokyüzlü konik fonksiyon**":

$$g(x) = 10(x-7) + 18\|x-7\| - 36$$



Şekil 8.3: Oluşturulan çokyüzlü konik fonksiyonların2-boyutlu uzayda gösterimi

Şekil 8.3'den de görüldüğü üzere elde edilen *optimal ayırıcı çokyüzlü konik fonksiyon* ile, A kümesi, g fonksiyonu S_0 alt seviye kümesi içerisinde, B kümesi ise g fonksiyonu S_0 alt seviye kümesi dışında kalmıştır ve dolayısıyla aşağıdaki eşitsizlikler sağlanmıştır.

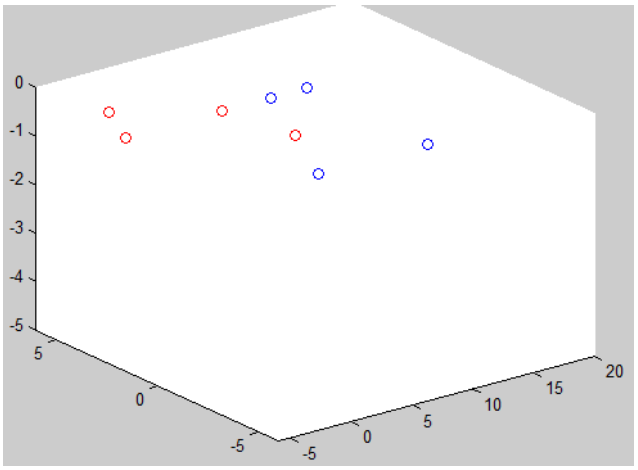
$$g(a^i) \le 0 \quad \forall a^i \in A$$
$$g(b^j) > 0 \quad \forall b^j \in B$$

Örnek 8.2: $\mathbb{R}^2$'de tanımlı $D = \{(-7,1),(2,-2),(-5,3),(1,1)\}$ ve

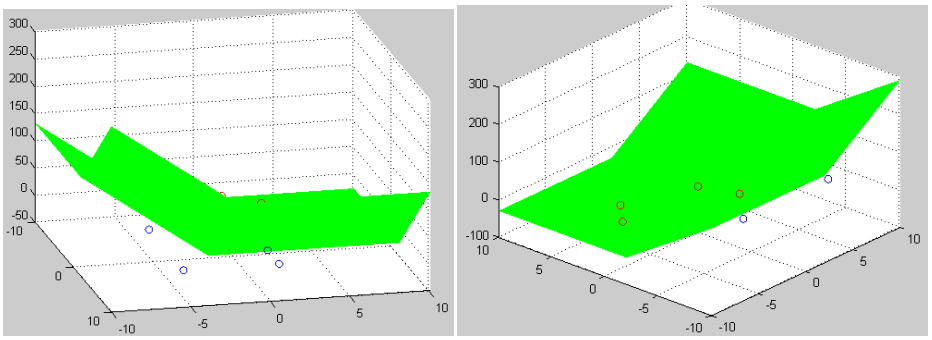
$G = \{(8,1),(5,1),(8,-5),(-1,-5)\}$ kümeleri ele alınsın.

Şekil 8.4' de D noktaları kırmızı G noktaları mavi ile belirtilmiştir ve görüldüğü üzere D ve G noktaları arasında doğrusal bir ayrımın sağlanamayacağı açıktır.

Elde edilen sonuçlar: $w=(1,-1), \xi=1, \gamma_1=1, \gamma_2=5, \gamma=\frac{5+1}{2}=3$



Şekil 8.4. Veri kümesinin 3-boyutlu uzayda gösterimi



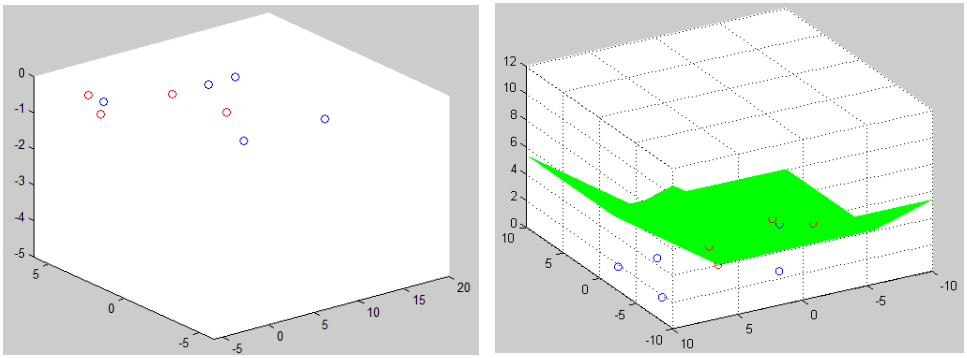
Şekil 8.5. 3-boyutlu uzayda optimal ayırıcı çokyüzlü konik fonksiyon gösterimi

G kümesine (-5,2) noktası eklendiğindeki veri kümesinin gösterimi ve algoritma sonucu elde edilen *g(x)* fonksiyonunun $\mathbb{R}^3$de grafiksel gösterimi şekil 8.6'da sunulmuştur ve elde edilen tüm değerler aşağıda belirtilmiştir.

$a=(2,-2), w=(0.33,-0.33), \xi=0.33, \gamma_1=1, \gamma_2=1, \gamma=1$

$\varepsilon_5=b^5$ noktasının hata değeri=2.0 , doğruluk=88.8

$g(x)=(0.33,-0.33)\times(x-(2,-2))+0.33\left\|x-(2,-2)\right\|_1-1$



Şekil 8.6. 3-boyutlu uzayda veri kümesi ve optimal ayırıcı çokyüzlü konik fonksiyon gösterimi

Sonuç olarak, destek vektör makinalarıyla doğrusal ayrımın sağlanamadığı veri kümelerinde, doğrular ya da hiperdüzlemler yerine çok yüzlü konik fonksiyonların kullanımıyla DVM-ÇKF yöntemi geliştirilmiş ve ayrımda başarı sağlanmıştır.

8.4 DVM-ÇKF Algoritmalarında Kümeleme Yönteminin Kullanımı

Tasarlanan bu algoritmalarda en fazla zamanın harcandığı kısmın tepe noktası seçiminde yaşandığı aşikardır. Çünkü her noktaya altproblem uygulanıp en iyi olan noktanın seçimi yapılmaktadır. Bu seçim doğruluk değerlerinde artış sağlamaktadır ancak hesaplama süresini uzattığı da göz ardı edilmemelidir. Bu nedenle bir önceki ikili ve çoklu sınıflandırma uygulamalarında kullandığımız kümeleme yöntemi DVM-ÇKF algoritmalarında da kullanılmıştır ve tasarlanan algoritma aşağıda sunulmuştur.

Algoritma 6: **DVM-ÇKF-KMLM**

A ve B, $\mathbb{R}^n$ de tanımlı iki küme olsun.

$$A=\left\{a^i \in \mathbb{R}^n : i \in I\right\}, B=\left\{b^j \in \mathbb{R}^n : j \in J\right\}$$

burada $i=\{1,..,m\}$, $j=\{1,..,p\}$ dir.

Başlangıç Adımı: *A* kümesinde kümeleme algoritmasını uygula. küme sayısı=*k* olsun. Herbir küme merkez noktası için, $a=a_s$, $s=1,2,..,k,$ için (*P*) altproblemini çöz.

A kümesinde yer alan *B*'de yer almayan *A* noktalarının sayısı=l_s ,$s=1,2,...,k,$ $a=a_{l_0}$, $l_0=\max\{l_s : s=1,2,...,k\}$

Adım 1: *P* altproblemini çöz.

$$(P)\quad \min(-(\gamma_2-\gamma_1)+\sum_{i=1}^{m}\varepsilon_i+\sum_{j=1}^{p}\varepsilon_j)$$

$$w(a^i-a)+\xi\left\|a^i-a\right\|_1-\gamma_1\le -1+\varepsilon_i,\quad i=1,..,m,$$

$$w(b^j-a)+\xi\left\|b^j-a\right\|_1-\gamma_2\ge 1-\varepsilon_j,\quad j=1,..,p,$$

$$\gamma_2-\gamma_1\ge 0,$$

$$\varepsilon_{i,j}\ge 0, w\in\mathbb{R}^n, \xi\in\mathbb{R}, \gamma_{1,2}\ge 1,$$

$w,\xi,\gamma_1,\gamma_2,\varepsilon_i,\varepsilon_j$,i=1,..,m, j=1,...,p , *(P)* 'nin bir çözümü olsun.

$\gamma=\frac{\gamma_2+\gamma_1}{2}$ yap.

Adım 2: *A* ve *B* kümelerini birbirinden ayıran *g(x)* fonksiyonunu aşağıdaki şekilde tanımla,

$$g(x)=g_{(w,\xi,\gamma,a)}(x)$$

8.5 Hesaplama Denemeleri

Bu bölümde 6.2 bölümünde açıklamaları verilen iki sınıflı gerçek hayat veri kümeleri (Bkz. Çizelge 6.2) üzerinde DVM-ÇKF algoritmaları uygulanmıştır.

Çizelge 8.1'de DVM-ÇKF 2 ve DVM-ÇKF-KMLM algoritma uygulamaları sonrası elde edilen doğruluk ve süreler verilmiştir. Sonuçlardan da görüldüğü gibi DVM-ÇKF-KMLM algoritması süre açısından önemli bir farkla DVM-ÇKF algoritmasından daha verimlidir Aynı zamanda doğrulukta da artışlar gözlemlenmiştir. Bu nedenle aynı tabloda DVM-ÇKF-KMLM için 10-kez çaprazlama sonucu elde edilen eğitim ve test değerleri de belirtilmiştir. Bu değerler de gösteriyor ki algoritmada yer alan doğrusal programlama üzerinde yapılan değişiklikler yani, B noktalarında hataya izin verme, sayesinde aşırı-uyum gözlenmemiştir.

Çizelge 8.1. DVM-ÇKF 2 ve DVM-ÇKF-KMLM uygulama sonuçları

	DVM-ÇKF 2		DVM-ÇKF-KMLM		DVM-ÇKF-KMLM (10-kez çapraz doğrulama)	
Veri Kümesi	Doğruluk %	Süre sn	Doğruluk %	Süre sn	eğitim	test
Diabetes	72.73	368 sn.	72.39	11.88 sn.	72.20	70.73
Ionosphere	75.6	656 sn.	84.33	10.19 sn.	81.47	80.72
Liver	64.95	296 sn.	64.63	5.43 sn.	63.18	64.99
WBCD	93.16	300 sn.	87.42	8.18 sn.	86.40	85.39
Heart	72.5	180 sn.	81.66	2.86 sn.	76.73	77.17
WBCP	64.9	130 sn.	76.80	2.11 sn.	75.30	76.68
Connectionist Bench	80.76	128 sn.	75.96	2.78 sn.	69.95	70.62
Haberman	74.18	166 sn.	74.18	3.94 sn.	74.17	74.21

9. SONUÇ

Günümüzde sınıflandırma birçok alanda çeşitli uygulamalara sahip olan bir yöntemdir. Sınıflandırma konusunda yapılan çalışmalarda elde edilen verimlilik kimi zaman, sadece zamandan kazanç sağlarken, kimi zaman hayati anlam bile taşıyabilmektedir. Örneğin, tıpta bir hastalık tanısında kullanılan sınıflandırma yöntemi sayesinde birçok hasta erken teşhis ile hayatına devam edebilmektedir. Bu yöntemin pek çok araştırmacı tarafından ele alınıp geliştirilmeye çalışılması da bu yönteme verilen önemin bir sonucudur. Bu yöntem üzerinde yapılan tüm araştırma ve geliştirmeler her uygulamada verimlilik sağlamasa da araştırmacıları yeni ve daha verimli çalışmalara yönlendirmektedir.

Bu tezde, sınıflandırma problemlerinin çözümü için matematiksel programlama içerisinde yer alan çokyüzlü konik fonksiyonlar temelli yeni algoritmalar tasarlanmıştır. Bunun yanısıra çokyüzlü konik fonksiyonlar tabanlı destek vektör makinaları yaklaşımı geliştirilmiştir.

İlk olarak ikili sınıflandırma konusu ele alınmıştır. İkili sınıflandırma için geliştirilen ilk algoritmada ÇKF algoritması temel alınmış ve zamandan kazanç sağlamak ve büyük veri kümelerinde verimlilik elde etmek amacıyla yine bir veri madenciliği yöntemi olan veri kümeleme yöntemi kullanılmıştır. Bu algoritma bir açıklayıcı örnek üzerinde denenmiş ve sonuçlar görsel olarak grafiklerle gösterilmiştir. Aynı zamanda ÇKF algoritması ile karşılaştırılarak elde edilen sonuçlar sunulmuştur.

İki kümenin ayrılmasını çok sayıda fonksiyon ile gerçekleştirme aşırı uyuma neden olabilmektedir. İkinci algoritmada, ilk algoritmada kullanılan kümeleme yöntemi ile merkez noktaların seçimlerinde elde edilen başarının yanısıra eğitim ile test başarı değerleri arası farkı azaltmak için algoritmada yer alan doğrusal programlama problemi yeniden tasarlanmıştır ve ayrım biraz daha gevşetilerek kesin ayrımdan kaçınılmış böylece eğitim başarısı düşsede test başarısında kimi örneklerde artış sağlanmıştır. Geliştirilen bu algoritma gerçek hayat veri kümeleri ve sentetik veri kümeleri üzerinde uygulanarak sonuçlar tablolarla belirtilmiştir.

Bölüm 7'de ikili sınıflandırma için tasarlanan çokyüzlü konik fonksiyonlar ve kümeleme tabanlı algoritma belli bir yaklaşım ile genişletilerek çoklu sınıflandırma problemlerinde de kullanılmasına olanak sağlanmıştır. Aynı zamanda karşılaştırma yapmak amaçlı ÇKF algoritması da aynı yaklaşım ile çoklu sınıflandırma için genişletilmiştir. Tasarlanan algoritmalar gerçek hayat veri kümeleri üzerinde uygulanmış ve sonuçlar tablolarla sunulmuştur.

Bölüm 8'de destek vektör makinaları yönteminde kesin ayrımın sağlandığı ve sağlanamadığı her iki durum için de düzlem veya hiperdüzlemler yerine çokyüzlü konik fonksiyonların kullanıldığı bir algoritma geliştirilmiştir aynı zamanda bu algoritmada da önceki bölümlerde olduğu gibi kümeleme yöntemi kullanılarak tasarlanan algoritmanın verimliliği arttırdığı gerçek hayat veri kümeleri üzerindeki uygulamalar sonucunda tablolarla gösterilmiştir.

Sunulan algoritmaların ve görsellerin yazılımı MATLAB paket programı kullanılarak gerçekleştirilmiştir.

KAYNAKLAR DİZİNİ

Bagirov A. M., 2005, Max-Min Separability, Optimization Methods and Software, 20:2-3, 277-296 pp.

Bagirov A. M., Rubinov A. M., Soukhoroukova N. V. and Yearwood J., 2003, Unsupervised and Supervised Data Classification via Nonsmooth and Global Optimizaiton, TOP 11(1), 1-75 pp.

Bagirov A. M., Ugon J., Webb D., Öztürk G. ve Kasimbeyli R., 2011, A Novel Piecewise Limear Classifier Based on Polyhedral Conic and Max-Min Separabilities, TOP, 1-22 pp.

Bagirov A.M. and Mardaneh K., 2006, Modified Global k-means Algorithm for Clustering in Gene Expression Data Sets, WISB '06 Proceedings of the 2006 workshop on Intelligent systems for bioinformatics – Vol: 73, 23-28 pp.

Bagirov. A. M. and Ugon. J., 2005, Supervised Data Classification via Max-Min Separability. In Continuous Optimisation: current trends and modern applications. Eds. Springer. Berlin, ch. 6., 175-208 pp.

Bennett K. P. and Demiriz A., 1998, Semi-supervised support vector machines, In D.A. Cohn M. S. Kearns. S. A. Solla. editor. Cambridge. MA, Advances in Neural In- formation Processing Systems -10-, 368-374 pp.

Bennett K. P. and Mangasarian O. L., 1992, Robust Linear Programming Discrimination of Two Linearly İnseparable Sets, Optimization Methods and Software, 1, 23-34 pp.

Bennett K. P. and Mangasarian O. L., 1993, Bilinear Separation of Two Sets in n-space, Computatioanl Optimization and Applications, November, Vol:2, Issue 3, 207-227 pp.

Bennett K. P. and Mangasarian O. L., 1994, Multicategory Discrimination via Linear Programming, Optimization Methods and Software, vol. 3, 27-39 pp.

Berry J. and Linoff G., 1997, Data Mining Techniques: For Marketing Sales and Customer Support, New York, USA, John Wiley & Sons, Inc., 643p.

Berry J., 1994, Database Marketing, Business Week, September 5, 56-62 pp.

Bhatia N., 2010, Survey of Nearest Neighbor Techniques, International Journal of Copmuter Science and Information Security, Vol:8, No:2, 302-305 pp.

KAYNAKLAR DİZİNİ (devam)

Bland C., 2002, The Discovery of multiple Level Profile Association Rules, Doctoral Thesis, Graduate School of Computer and Information Science of University of Missisipi, 43-50 pp.

Bradley P. S., Fayyad U. M. and Mangasarian O. L., 1999, Mathematical Programming for Data Mining: Formulations and Challenge, Informs Journal on Computing, 11(3): 217-238.

Bredensteiner E. J. and Bennett K. P., 1999, Multicategory Classifcation by Support Vector Machines, Computational Optimization and Application, 12:53-79.

Coşkun S. ve Kartal M., 2004, Lojisitk Regresyon Analizinin İncelenmesi ve Dişhekimliğinde bir Uygulaması, Cumhuriyet Ünv. Diş Hekimliği Dergisi, 7(1): 41-50.

Cybenko G., 1989, Approximation by superpositions of a sigmoidal function, Mathematical Control Signals System, 2:303-314.

Çakır M., 2005, Firma Başarısızlığının Dinamiklerinin Belirlenmesinde Makine Öğrenme Teknikleri:Ampirik Uygulamalar ve Karşılaştırmalı Analiz, Türkiye Cunhuriyet Merkez Bankası İstatistik genel Müdürlüğü, Ankara, Aralık, Uzmanlık Yeterlilik Tezi, 142 s.(yayınlanmamış).

Data Generator (http://www.datagenerator.com/) (Erişim tarihi Mayıs 2013)

David B., 1999, Data Mining Transformed, Information Week, 751:86-88.

Fayyad U., Piatetsky-Shapiro G. and Smyth P., 1996, From Data Mining to Knowledge Discovery in Databases, AI Magazine, 17(3): 37-54.

Frank E., and Witten I., 2000, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, San Francisco, Morgan Kaufmann Publishes, 120-129, 267-277.

Gasimov R. N., 2001, Characterization of the Benson Proper Efficiency and Scalarization in Nonconvex Vector Optimization, Multicriteria Decision Making in The New Millenium, 507:189-198.

Gasimov R. N. and Öztürk G., 2010, Separation via Polyhedral Conic Functions, Optimization Methods and Software, 21(4): 527-540.

Glover F., 1990, Improved Linear Programming Models for Discriminant Analysis, Decision Sciences, 21(4): 771-785.

KAYNAKLAR DİZİNİ (devam)

Gündoğan K. K., Alataş B. ve Karcı A., 2004, Mining Classification Rules by Using Genetic Algorithms with Non-random Initial Population and Uniform Operator, Turkish Journal of Electrical Engineering and Computer Sciences (ELEKTRİK), 12(1): 43-52.

Hand D. J., 1998, Data Mining: Statistics and More?, The American Statistician, 52: 112-118.

Hornik K., Stinchcombe and White H., 1989, Multilayer feedforward networks are universal approximators, Neural Networks, 2: 359–366.

Hornik K., 1991, Approximation capabilities of multilayer feedforward networks, Neural Networks, 4: 251–257 pp.

Hui S., Jha G., 2000, Application Data Mining for Customer Service Support, Information and Management, 38:1-13.

Jacobs P., 1999, Data Mining: What General Managers Need to Know, Harward Management, Update, 4(10):8.

Kaufman. L. and Rousseeuw. P.J., 1990, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley-interscience, New York, ISBN 0-471-87876-6, 347p.

Kitler R. and Wang W., 1998, The Emerging Role of Data Mining, Solid State Technology, 42(11):45.

Kohavi R., 1995, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1137-1143 pp.

Kusiak A., Kernstine K. H., Kern J. A., McLaughlin K. A. and Tseng T. L., 2000, Data Mining: Medical and Engineering Case Studies, Proceedings of the Industrail Engineering Research 2000 Conference, Cleveland, Ohio, May 21-23, 1-7 pp.

Mammone A., Turchi M. and Cristianini N., 2009, Support Vector Machines, Wires: Wiley's Interdisciplinary Reviews in Computational Statistics, June, 1: 283-289.

Mangasarian O. L., 1965, Linear and Nonlinear Separation of Patterns by Linear Programming, Operations Research 13: 444-452.

KAYNAKLAR DİZİNİ (devam)

Nabiyev V. V., 2005, Yapay Zeka: Problemler, Yöntemler, Algoritmalar, Ankara: Seçkin Yayıncılık, 2. Ed., 752s

Orsenigo C. and Vercellis C., 2007, Accurately Learning from few examples with a Polyhedral Classifier, Computational Optimization and Applications, 38(2): 235-247.

Ou G. and Murphey Y. L., 2007, Multi-Class Pattern Classifcation Using Neural Networks, Pattern Recognition. 40(1): 4-18.

Öztürk G., 2007, Sınıflandırma Problemlerinin Çözümü için Yeni Bir Matematiksel Programlama Çalışması, Doktora Tezi, Osmangazi Üniversitesi, Eskişehir, 133s.

Pawlak Z. and Skowron A., 1994, Rough Membership Functions, Advances in Dempster Shaefer Theory of Evidence, John Wiley & Sons, Inc., New York, Chichester, Brisbane, Toronto, Singapore, 251-271 pp.

Pawlak Z., 2002, Rough Sets, Decision Algorithms and Bayes Theorem, Europan Journal of Operation Research, 136: 181-189.

Richard M. D. and Lippmann R., 1991, Neural network classifiers estimate Bayesian a posteriori probabilities, Neural Comput., 3: 461–483.

Shah S. and Kursak A., 2004, Data Mining and Genetic Algorithms based gene/SNP Selection, Artificial Intelligence in Medicine, 31:183-196.

Smith F. W., 1968, Pattern Classifier Design by Linear Programming, IEEE Transactions on Computers, C-17(4): 367-372.

Tax D. M. J. and Duin R. P. W., 2002, Using Two-Class Classifiers for Multiclass Classification, Proc. ICPR2002, Quebec City, Canada, August, 4p.

UC Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/) (Erişim tarihi Mayıs 2013)

Zadeh L. A., 1965, Fuzzy Sets, Information and Control, 8: 338-353.

ÖZGEÇMİŞ

09.11.1983 tarihinde Muğla'da doğmuştur. İlk, orta ve lise öğrenimini yine aynı ilde tamamladıktan sonra 2002 yılında Ege Üniversitesi Fen Fakültesi Matematik Bölümüne girmeye hak kaznamıştır. Bu bölümden 2006 yılında mezun olduktan sonra aynı yıl içerisinde Ege Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim dalı Bilgisayar Bilimlerinde yüksek lisans eğitimine başlayıp 2008 yılında tamamlamıştır. Sonrasında aynı üniversitede 2008 yılında Matematik Anabilim dalında doktora öğrenimine başlamıştır.

2007-2012 yıllarında ingilizce eğitim veren İzmir Ekonomi Üniversitesi Mühendislik ve Bilgisayar Bilimleri Fakültesinde araştırma görevlisi olarak çalışmıştır. Aynı zamanda 2007-2008'de TÜBİTAK yurtiçi yükseklisans bursiyeri ve 2008-2013 yıllarında TÜBİTAK yurtiçi doktora bursiyeri olmaya hak kazanmıştır. Bunun yanısıra 2010 yılında Dokuz Eylül Üniveristesi Buca Eğitim Fakültesi'nden pedagojik formasyon sertifikasını almıştır.

Tezden yapılan çalışmalar aşağıda sunulmuştur:

Makaleler

Satı U. N., Ordin B., (2013), A Mathematical Programming Approach for Classification Problems, SCI sınıftan bir dergiye gönderilecek.

Uluslararası Konferans Yayınları

- Ordin B., **Uylaş, N.** (2013), A mathematical Programming Approach For Binary Classification Problems, 26th Europan Conference on Operational Research, Rome, ITALY

- Ordin B., **Uylaş, N.** (2010), Methods based on mathematical optimization for semi-supervised data classification, 24th European Conference on Operational Research, Lisbon, PORTUGAL

- Ordin B., **Uylaş, N.** (2010), Linear Separability: Quasisecant Method and Application to Semisupervised Data Classification, The ISI proceedings of 24th MEC-EUROPT 2010-Continious Optimization and Information-based Technologies in the Financial Sector, pp.264-269, Izmir, Turkey, 2010.

Ulusal Konferans Yayınları

- Ordin B., **Uylaş, N.,** Kaya O., (2008), Genelleştirilmiş sınıf Atama

Problemi için bir Genetik Algoritma Yaklaşımı, YA & EM Ulusal Kongresi, İstanbul, TÜRKİYE

- Ordin B., **Uylaş, N.** (2007),Değişken Modellerde Denge Fiyatlarının Araştırılması için bir Global Optimizasyon Modeli, YA & EM 2007 (Yöneylem Araştırması ve Endüstri Mühendisliği 27. Ulusal Kongresi), İzmir, TÜRKİYE

- Ordin B., **Uylaş N.** (2013), İkili Sınıflandırma için Çokyüzlü Konik Fonksiyonlar ve Kümeleme Tabanlı Bir Yaklaşım, YA&EM 2013 (Yöneylem Araştırması ve Endüstri Mühendisliği 27. Ulusal Kongresi), İstanbul, TÜRKİYE

.

EKLER

EK 1. Program Kodları

EK 1. PROGRAM KODLARI

a) Çokyüzlü Konik Fonksiyonlar Algoritması (ÇKF)

```
load yeni.txt;
tic;
m=length(yeni(:,1));
sinifno=0;
anai=1;
R=yeni;
dng=1;
c=0;
sinifsayisi=1;

while sinifno<sinifsayisi

n=2; %boyut
s=0;
sil=0;
clear yeni;
clear G;
clear d;
yeni=R;
m=length(R(:,1));

for i=1:m
   if R(i,n+1)~=sinifno
      s=s+1;
      d(s)=i;
   for j=1:n+1
    G(s,j)=R(i,j);
   end
   end
end

if(s==0)
   d=[];
   G=[];
   merkezler=[];
   ye=[];
   mk=merkezler;
   y2=ye;
   display('%n GERİ KALANLAR AYNI SINIFTAN %n')
   dogru11=sinifno;
   return;
end
 yeni(d,:)=[];

while length(yeni(:,1))==0.0
   sinifno=sinifno+1.0;
    if sinifno>sinifsayisi break;end
   s=0;
   clear d;
   clear G;
   yeni=R;
```

```
    for i=1:m
    if R(i,n+1)~=sinifno
        s=s+1;
        d(s)=i;
    for j=1:n+1
     G(s,j)=R(i,j);
    end
    end
    end
   if(s==0)
    d=[];
    G=[];
    merkezler=[];
    ye=[];
    mk=merkezler;
    y2=ye;
     display('%n GERİ KALANLAR AYNI SINIFTAN %n')
    dogru11=sinifno;
    return;
   end
   yeni(d,:)=[];

end

if sinifno>sinifsayisi break;end


kalan=length(yeni(:,1));
while kalan>0

m=length(yeni(:,1));
p= length(G(:,1));
max=0;
for bas=1:m
    sayi(anai,bas)=0;
for j=1:n+1
t(bas,j)=yeni(bas,j);
end
end

for bas=1:m %herbir nokta için değerlendirme

for say=1:m
top=0;

for j=1:n
ak(say,j)=yeni(say,j)-t(bas,j);
end

for j=n+1:n+2
if j==n+1

for k=1:n
top=top+abs(yeni(say,k)-t(bas,k));
ak(say,j)=top;
end

end

if j==n+2
```

```
ak(say,j)=-1;
end

end

for j=n+3:n+2+m
if j-say==n+2
ak(say,j)=-1;
else
ak(say,j)=0;
end

end

end


for say=1:p
top=0;

for j=1:n
bk(say,j)=-(G(say,j)-t(bas,j));
end

for j=n+1:n+2

if j==n+1
for k=1:n
top=top+abs(G(say,k)-t(bas,k));
bk(say,j)=-top;
end
end
 if j==n+2
bk(say,j)=1;
 end
end

for j=n+3:n+2+m
bk(say,j)=0;
end
end

for i=1:m
    for j=1:n+2+m
    C(i,j)=ak(i,j);
    end
end
for i=m+1:m+p
    for j=1:n+2+m
        C(i,j)=bk(i-m,j);
    end
end

 for h=1:m+p
     b(h)=-1;
 end

for o=1:n+2+m
    if o<=n+2 f(o)=0;
else
```

```
    f(o)=1;
    end
end
for i=1:n
    lb(i)=-1;
    ub(i)=1;
end
lb(n+1)=0;
ub(n+1)=1;
lb(n+2)=1;
ub(n+2)=inf;
for i=n+2+1:n+2+m
   lb(i)=0;
   ub(i)=inf;
end
beq=[];
Ceq=[];

[ans,fval,exitflag,output]=linprog(f,C,b,Ceq,beq,lb,ub,[]) %returns a structure output that
contains information about the optimization.
if exitflag== 1

son(anai,bas)= f*ans;

for d=1:m
top=0;
 for j=1:n+2
top=top+(ak(d,j)*ans(j));
end
if top<=0.0
   sayi(anai,bas)=sayi(anai,bas)+1;
end
end

if sayi(anai,bas)>=max %max nokta ayıran tepe noktası indexi maxbas
    max=sayi(anai,bas);
    maxbas=bas;
end

for j=1:n+2+m
    sonuc(bas,j)=ans(j);
end
end
end%herbir nokta için değerlendirildi.

if max~=0

for i=1:n+3
    if i<=n+2
ye(dng,i)=sonuc(maxbas,i);%bulunan polyhedral konik fonksiyon parametreleri
    else
        ye(dng,i)=sinifno;%son sütun sınıfları gösteriyor.
         end
end


merkez(anai)=maxbas; %merkez indexleri
for j=1:n
merkezler(dng,j)=yeni(maxbas,j); %merkezlerin koordinatları
end
```

```
merkezler(dng,n+1)=sinifno;

for say=1:m
top=0;
for j=1:n
ak(say,j)=yeni(say,j)-t(maxbas,j);
end

for j=n+1:n+2
if j==n+1
for k=1:n
top=top+abs(yeni(say,k)-t(maxbas,k));
ak(say,j)=top;
end
end
if j==n+2
ak(say,j)=-1;
end
end
end
ayrilansayisi(anai)=0;

k=0;
for i=1:m
   top(i)=0;
   for j=1:n+2
      top(i)=top(i)+ak(i,j)*ye(dng,j);
   end
   if top(i)<=0
      k=k+1;
      d(k)=i;
      ayrilansayisi(anai)= ayrilansayisi(anai)+1;

   end

end

 yeni(d,:)=[];

for i=1:ayrilansayisi(anai)
ayrilanlar(anai,i)=d(i);
end

anai=anai+1;
dng=dng+1;
kalan=length(yeni(:,1));

if kalan>=0
clear ak;
clear bk
clear C
clear f
clear b
clear top
clear l
clear t
clear d
clear bas;
clear maxbas;
end
```

```
else
    c=c+1
    fprintf('ayrilamayan sinif: %d', sinifno)
    kalan=0;
    ayrilmayanlar=yeni;
    break;
end
end
sinifno=sinifno+1.0;
clear sonuc;
clear ayrilansayisi;
clear ans;

clear yeni;
clear t;
end
mk=merkezler;
y2=ye;
toc;
```

b) Algoritma 2: Kümeleme Tabanlı ÇKF algoritması 1 (KMLM-ÇKF 1)

```
tic;
close all;clc;

load D.txt;
% normal=D;
% D=mat2gray(D);
% D(:,5)=normal(:,5);
% bak=D;

m=length(D(:,1));
sinifno=0.0; %en küçük sınıf numarası
anai=1;
kalanlar=0;
sinifsayisi=1.0 %en büyük sınıf numarası
clustersayisi=20;

while sinifno<sinifsayisi

n=13;
s=0;
sil=0;
clear D;
clear G;

load D.txt;
% normal=D;
% D=mat2gray(D);
% D(:,5)=normal(:,5);
 m=length(D(:,1));

for i=1:m
    if D(i,n+1)~=sinifno
        s=s+1;
        d(s)=i;
    for j=1:n+1
     G(s,j)=D(i,j);
```

```
    end
    end
end
  D(d,:)=[];

  while length(D(:,1))== 0.0
      sinifno=sinifno+1.0;
    if sinifno>sinifsayisi
        break;
    end
    clear D;
    clear d;
load D.txt;
m=length(D(:,1));
 s=0;
for i=1:m
    if D(i,n+1)~=sinifno
        s=s+1;
        d(s)=i;
    for j=1:n+1
     G(s,j)=D(i,j);
    end
    end
end
 D(d,:)=[];
  end


    while(length(D(:,1))<clustersayisi)
        if sinifno>sinifsayisi
        break;
        end
kalan=length(D(:,1));
for i=1+kalanlar:kalanlar+kalan
    for j=1:n+1
yeni(i,j)=D(i-kalanlar,j); %kalanlar yeni matrisine atandı.
    end
end
kalanlar=kalanlar+kalan;
clear D;
load D.txt;
% normal=D;
% D=mat2gray(D);
% D(:,5)=normal(:,5);
m=length(D(:,1));
 sinifno=sinifno+1.0;
 if sinifno>sinifsayisi break; end
 s=0;
for i=1:m
    if D(i,n+1)~=sinifno
        s=s+1;
        d(s)=i;
    for j=1:n+1
     G(s,j)=D(i,j);
    end
    end
end
  D(d,:)=[];
end
```

```
 if sinifno>sinifsayisi break; end
[IDX,t]=kmeans(D,clustersayisi);%clusterlar belirleniyor
%[IDX, Cluster, Err] = kmedoid2(D, clustersayisi, 10000)


tl=length(t(:,1));
m=length(D(:,1));
p= length(G(:,1));
for bas=1:tl
   sayi(anai,bas)=0;
end

for bas=1:tl %herbir cluster için değerlendirme

      m=length(D(:,1));
      if m==0 continue; end
for say=1:m
top=0;

for j=1:n
ak(say,j)=D(say,j)-t(bas,j);
end

for j=n+1:n+2
if j==n+1

for k=1:n
top=top+abs(D(say,k)-t(bas,k));
ak(say,j)=top;
end

end

if j==n+2
ak(say,j)=-1;
end

end

for j=n+3:n+2+m
if j-say==n+2
ak(say,j)=-1;
else
ak(say,j)=0;
end

end

end

for say=1:p
top=0;

for j=1:n
bk(say,j)=-(G(say,j)-t(bas,j));
end

for j=n+1:n+2

if j==n+1
```

```
for k=1:n
top=top+abs(G(say,k)-t(bas,k));
bk(say,j)=-top;
end
end
 if j==n+2
bk(say,j)=1;
 end
end

for j=n+3:n+2+m
bk(say,j)=0;
end
end

for i=1:m
   for j=1:n+2+m
   C(i,j)=ak(i,j);
   end
end
for i=m+1:m+p
   for j=1:n+2+m
      C(i,j)=bk(i-m,j);
   end
end

 for h=1:m+p
    b(h)=-1;
 end


for o=1:n+2+m
   if o<=n+2 f(o)=0.0;
else
   f(o)=1.0;
   end
end
for i=1:n
   lb(i)=-inf;
   ub(i)=inf;
end
lb(n+1)=-inf;
ub(n+1)=inf;
lb(n+2)=1.0;
ub(n+2)=inf;
for i=n+2+1:n+2+m
  lb(i)=0.0;
  ub(i)=inf;
end
beq=[];
Ceq=[];


 [ans,fval,exitflag,output]=linprog(f,C,b,Ceq,beq,lb,ub,[]) %returns a structure output that
contains information about the optimization.
if exitflag == 1
   display('%n NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN %n');
    fprintf('sinif== %d ', sinifno);

son(anai,bas)= f*ans;
```

```
sayi(anai,bas)=0;
for d=1:m
top=0;
 for j=1:n+2
top=top+(ak(d,j)*ans(j));
end
if top<=0
   sayi(anai,bas)=sayi(anai,bas)+1;
end
end

for j=1:n+2+m
    sonuc(bas,j)=ans(j);
end

for i=1:n+3
    if i<=n+2
y(anai,i)=sonuc(bas,i);%bulunan polyhedral konik fonksiyon parametreleri
    else
         y(anai,i)=sinifno;%son sütun sınıfları gösteriyor.
          end
end

merkez(anai)=bas; %merkez indexleri
for j=1:n+1
merkezler(anai,j)=t(bas,j); %merkezlerin koordinatları
end

for say=1:m
top=0;
for j=1:n
ak(say,j)=D(say,j)-t(bas,j);
end

for j=n+1:n+2
if j==n+1
for k=1:n
top=top+abs(D(say,k)-t(bas,k));
ak(say,j)=top;
end
end
if j==n+2
ak(say,j)=-1;
end
end
end

ayrilansayisi(anai)=0;

k=0;
for i=1:m
    top(i)=0;
    for j=1:n+2
        top(i)=top(i)+ak(i,j)*y(anai,j);
    end
    if top(i)<=0.0
        k=k+1;
        d(k)=i;
        ayrilansayisi(anai)= ayrilansayisi(anai)+1;
```

```
    end

end

if k>0
 D(d,:)=[];
end

for i=1:ayrilansayisi(anai)
ayrilanlar(anai,i)=d(i);
end

anai=anai+1;

if bas<=tl
clear ak;
clear bk
clear C
clear f
clear b
clear top
clear l
clear d
end


end

end %herbir cluster için pcf yapıldı.

kalan=length(D(:,1));


for i=1+kalanlar:kalanlar+kalan
    for j=1:n+1
yeni(i,j)=D(i-kalanlar,j); %kalanlar yeni matrisine atandı.
    end
end
kalanlar=kalanlar+kalan;
clear sonuc;
clear ans;
gkalan=length(G(:,1));
clear kalan;
clear t
sinifno=sinifno+1.0;
end
display('%n PCF E GEÇİLDİ %n');
yeni1=[yeni;G]; %%for 2 dimensions
fprintf ('kalanlar= %d \n', kalanlar)

if kalanlar>0
gidenler=yeni1;
[dogru11,mk,y2]=pcf(yeni1);
pcfp=[y;y2];
pcfm=[merkezler;mk];
else
    pcfp=y;
    pcfm=merkezler;
end
```

```
toc;
```

c) Algoritma 3: Kümeleme Tabanlı ÇKF Algoritması 2

```
tic;
close all;clc
load D.txt;
m=length(D(:,1));
sinifno=1.0; %en küçük sınıf numarası
anai=1;
kalanlar=0;
sinifsayisi=2.0; %en büyük sınıf numarası
clustersayisi=40;
n=3;
s=0;
sil=0;
 m=length(D(:,1));
for i=1:m
   if D(i,n+1)~=sinifno
      s=s+1;
      d(s)=i;
   for j=1:n+1
    G(s,j)=D(i,j);
   end
   end
end
D(d,:)=[];
[IDX,t]=kmeans(D,clustersayisi);%clusterlar belirleniyor
%[IDX, Cluster, Err] = kmedoid2(D, clustersayisi, 10000)
%t=Cluster;
tl=length(t(:,1))
m=length(D(:,1));
p= length(G(:,1));
for bas=1:tl
   sayi(anai,bas)=0;
end
a=0;
for bas=1:tl %herbir cluster için değerlendirme
      m=length(D(:,1));
      if m==0 continue; end
for say=1:m
top=0;
for j=1:n
ak(say,j)=D(say,j)-t(bas,j);
end
for j=n+1:n+2
if j==n+1
for k=1:n
top=top+abs(D(say,k)-t(bas,k));
ak(say,j)=top;
end
end
if j==n+2
ak(say,j)=-1;
end
end
for j=n+3:n+2+m
if j-say==n+2
ak(say,j)=-1;
else
```

```
ak(say,j)=0;
end
end
for j=n+3+m:n+2+m+p
    ak(say,j)=0;
end
end
for say=1:p
top=0;
for j=1:n
bk(say,j)=-(G(say,j)-t(bas,j));
end
for j=n+1:n+2
if j==n+1
for k=1:n
top=top+abs(G(say,k)-t(bas,k));
bk(say,j)=-top;
end
end
 if j==n+2
bk(say,j)=1;
 end
end
for j=n+3:n+2+m
bk(say,j)=0;
end
for j=n+3+m:n+2+m+p
if j-say==n+2+m
bk(say,j)=-1;
else
bk(say,j)=0;
end
end

end

for i=1:m
    for j=1:n+2+m+p
    C(i,j)=ak(i,j);
    end
end
for i=m+1:m+p
    for j=1:n+2+m+p
        C(i,j)=bk(i-m,j);
    end
end

 for h=1:m+p
     b(h)=-1;
 end

for o=1:n+2+m
    if o<=n+2 f(o)=0.0;
else
    f(o)=1/m;
    end
end
for o=n+3+m:n+2+m+p
     f(o)=10/p;
    end
```

```
for i=1:n
    lb(i)=-1;
    ub(i)=1;
end
lb(n+1)=0;
ub(n+1)=inf;
lb(n+2)=1;
ub(n+2)=inf;
for i=n+3:n+2+m+p
   lb(i)=0;
   ub(i)=inf;
end
beq=[];
Ceq=[];
 options = optimset('LargeScale', 'off', 'Simplex', 'on');
options = optimset(options,'Maxiter',15000);
 [ans,fval,exitflag,output]=linprog(f,C,b,Ceq,beq,lb,ub,[],options)
if exitflag == 1
    a=a+1;
    display('%n NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN %n');
     fprintf('sinif== %d ', sinifno);
son(anai,bas)= f*ans;
sayi(anai,bas)=0;
for d=1:m
top=0;
 for j=1:n+2
top=top+(ak(d,j)*ans(j));
end
if top<=0
   sayi(anai,bas)=sayi(anai,bas)+1;
end
end

for j=1:n+2+m+p
    sonuc(bas,j)=ans(j);
end
for i=1:n+3
    if i<=n+2
y(anai,i)=sonuc(bas,i);%bulunan polyhedral konik fonksiyon parametreleri
    else
         y(anai,i)=sinifno;%son sütun sınıfları gösteriyor.
          end
end

merkez(anai)=bas; %merkez indexleri
for j=1:n+1
merkezler(anai,j)=t(bas,j); %merkezlerin koordinatları
end
for say=1:m
top=0;
for j=1:n
ak(say,j)=D(say,j)-t(bas,j);
end

for j=n+1:n+2
if j==n+1
for k=1:n
top=top+abs(D(say,k)-t(bas,k));
ak(say,j)=top;
end
```

```
end
if j==n+2
ak(say,j)=-1;
end
end
end
ayrilansayisi(anai)=0;
k=0;
for i=1:m
   top(i)=0;
   for j=1:n+2
      top(i)=top(i)+ak(i,j)*y(anai,j);
   end
   if top(i)<=0.0
      k=k+1;
      d(k)=i;
      ayrilansayisi(anai)= ayrilansayisi(anai)+1;
   end
end

if k>0
 D(d,:)=[];
end
for i=1:ayrilansayisi(anai)
ayrilanlar(anai,i)=d(i);
end
anai=anai+1;
if bas<tl
clear ak;
clear bk
clear C
clear f
clear b
clear top
clear l
clear d
end
end
sboyut=length(D(:,1));
if sboyut==0  break;
end
end %herbir cluster için pcf yapıldı.
if a==0
   generalization=0
   generalizationt=0;
   y2=[];
   return;
end

for i=1:n+3
   y2(:,i)=y(:,i);
   end
pcfp=y2;
toc;
```

d) Algoritma 4:Çoklu Sınıflandırmada kümeleme tabanlı çokyüzlü konik fonksiyon uygulaması (Çoklu-KMLM-ÇKF)

```
load yeni.txt;
m=length(yeni(:,1));
```

```
sinifno=1,0;
anai=1;
R=yeni;
dng=1;
c=0;
anai=1;
kalanlar=0;
sinifsayisi=2,0;
dogru11=0;
clustersayisi=10;
tic;
while sinifno<=sinifsayisi
n=10;
s=0;
sil=0;
clear yeni;
clear G;
clear d;
yeni=R;
m=length(R(:,1));
for i=1:m
    if R(i,n+1)~=sinifno
        s=s+1;
        d(s)=i;
    for j=1:n+1
     G(s,j)=R(i,j);
    end
    end
end
 yeni(d,:)=[];
while length(yeni(:,1))==0.0
    sinifno=sinifno+1.0;
     if sinifno>sinifsayisi break;end
    s=0;
    clear d;
    clear G;
    yeni=R;
    for i=1:m
    if R(i,n+1)~=sinifno
        s=s+1;
        d(s)=i;
    for j=1:n+1
     G(s,j)=R(i,j);
    end
    end
    end

   yeni(d,:)=[];

end

if sinifno>sinifsayisi break;end

kalan=length(yeni(:,1));
m=length(yeni(:,1));
p= length(G(:,1));
[IDX,t]=kmeans(yeni,clustersayisi);
max=0;
tl=length(t(:,1));
m=length(yeni(:,1));
```

```
p= length(G(:,1));
%max1=0;
for bas=1:tl
   sayi(anai,bas)=0;
end
for bas=1:tl %herbir cluster için değerlendirme
      m=length(yeni(:,1));
      if m==0 continue; end
for say=1:m
top=0;

for j=1:n
ak(say,j)=yeni(say,j)-t(bas,j);
end
for j=n+1:n+2
if j==n+1
for k=1:n
top=top+abs(yeni(say,k)-t(bas,k));
ak(say,j)=top;
end
end
if j==n+2
ak(say,j)=-1;
end
end
for j=n+3:n+2+m
if j-say==n+2
ak(say,j)=-1;
else
ak(say,j)=0;
end
end
end
for say=1:p
top=0;
for j=1:n
bk(say,j)=-(G(say,j)-t(bas,j));
end
for j=n+1:n+2
if j==n+1
for k=1:n
top=top+abs(G(say,k)-t(bas,k));
bk(say,j)=-top;
end
end
 if j==n+2
bk(say,j)=1;
 end
end
for j=n+3:n+2+m
bk(say,j)=0;
end
end
for i=1:m
   for j=1:n+2+m
   C(i,j)=ak(i,j);
   end
end
for i=m+1:m+p
   for j=1:n+2+m
```

```
            C(i,j)=bk(i-m,j);
        end
    end
     for h=1:m+p
        b(h)=-1;
     end
    for o=1:n+2+m
        if o<=n+2 f(o)=0.0;
    else
        f(o)=1.0;
        end
    end
    for i=1:n
        lb(i)=-inf;
        ub(i)=inf;
    end
    lb(n+1)=-inf;
    ub(n+1)=inf;
    lb(n+2)=1.0;
    ub(n+2)=inf;
    for i=n+2+1:n+2+m
       lb(i)=0.0;
       ub(i)=inf;
    end
    beq=[];
    Ceq=[];
     [ans,fval,exitflag,output]=linprog(f,C,b,Ceq,beq,lb,ub,[]) %returns a structure output that
contains information about the optimization.
    if exitflag == 1
        display('%n NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN %n');
         fprintf('sinif== %d ', sinifno);

    son(anai,bas)= f*ans;

    sayi(anai,bas)=0;
    for d=1:m
    top=0;
     for j=1:n+2
    top=top+(ak(d,j)*ans(j));
    end
    if top<=0
       sayi(anai,bas)=sayi(anai,bas)+1;
    end
    end

    for j=1:n+2+m
        sonuc(bas,j)=ans(j);
    end

    for i=1:n+3
        if i<=n+2
    y(anai,i)=sonuc(bas,i);%bulunan polyhedral konik fonksiyon parametreleri
        else
             y(anai,i)=sinifno;%son sütun sınıfları gösteriyor.
              end
    end

    for j=1:n+1
    merkezler(anai,j)=t(bas,j); %merkezlerin koordinatları
    end
```

```
for say=1:m
top=0;
for j=1:n
ak(say,j)=yeni(say,j)-t(bas,j);
end

for j=n+1:n+2
if j==n+1
for k=1:n
top=top+abs(yeni(say,k)-t(bas,k));
ak(say,j)=top;
end
end
if j==n+2
ak(say,j)=-1;
end
end
end

ayrilansayisi(anai)=0;

k=0;
for i=1:m
   top(i)=0;
   for j=1:n+2
      top(i)=top(i)+ak(i,j)*y(anai,j);
   end
   if top(i)<=0.0
      k=k+1;
      d(k)=i;
      ayrilansayisi(anai)= ayrilansayisi(anai)+1;
   end
end

if k>0
 yeni(d,:)=[];
end
for i=1:ayrilansayisi(anai)
ayrilanlar(anai,i)=d(i);
end
anai=anai+1;
if bas<=tl
clear ak;
clear bk
clear C
clear f
clear b
clear top
clear l
clear d
end

end
kume=bas
end %herbir cluster için pcf yapıldı.
sinifno=sinifno+1;
end
toc;
```

e) Algoritma 5: Destek Vektör Makinalarında Çokyüzlü Konik Fonksiyonlar (DVM-ÇKF 1)

```
load D.txt;
load G.txt;
n=2
v=0;
m=length(D(:,1));
kalan=m;
anai=1;

 tic;
m=length(D(:,1));
p= length(G(:,1));
min=10000;
 for bas=1:m %herbir nokta için değerlendirme
for j=1:n
 t(bas,j)=D(bas,j);
end
 end
 for bas=1:m
for say=1:m
top=0;
for j=1:n
ak(say,j)=D(say,j)-t(bas,j);
end
    for j=n+1:n+3
if j==n+1
for k=1:n
top=top+abs(D(say,k)-t(bas,k));
ak(say,j)=top;
end
end
if j==n+2
ak(say,j)=-1;
end
if j==n+3
   ak(say,j)=0;
end
    end

    for k=n+4:n+3+m+p
      if k-say==n+3
         ak(say,k)=-1;
      else
         ak(say,k)=0;
      end

    end
end

for say=1:p
top=0;

for j=1:n
bk(say,j)=-(G(say,j)-t(bas,j));
end
    for j=n+1:n+3
if j==n+1
for k=1:n
```

```
top=top+abs(G(say,k)-t(bas,k));
bk(say,j)=-top;
end
end
 if j==n+2
bk(say,j)=0;
 end
 if j==n+3
    bk(say,j)=1;
end

   end

   for k=n+4:n+3+m
      bk(say,k)=0;
   end

   for k=n+4+m:n+3+m+p
     if k-say==n+3+m
        bk(say,k)=-1;
     else bk(say,k)=0;
     end
   end
end

for i=1:m
   for j=1:n+3+m+p
   C(i,j)=ak(i,j);
   end
end
for i=m+1:m+p
   for j=1:n+3+m+p
      C(i,j)=bk(i-m,j);
   end
end

 for j=1:n+3+m+p
    if j<n+1
C(m+p+1,j)=0;
    end
       if j==n+2
       C(m+p+1,j)=1;
       end
       if j==n+3
       C(m+p+1,j)=-1;
       end
       if j>n+3
         C(m+p+1,j)=0;
       end
    end
 for h=1:m+p
    b(h)=-1;
 end
 b(m+p+1)=-1;
for o=1:n+3
   if o<=n+1
      f(o)=0;
   end
if o==n+2
   f(o)=1;
```

```
end
    if o==n+3
        f(o)=-1;
    end
end

for o=n+4:n+3+m+p
    f(o)=1;
end

for i=1:n
    lb(i)=-1;
    ub(i)=1;
end
lb(n+1)=0;
ub(n+1)=1;
lb(n+2)=1;
ub(n+2)=inf;
lb(n+3)=1;
ub(n+3)=inf;
for o=n+4:n+3+m+p
    lb(o)=0;
    ub(o)=inf;
end
beq=[];
Ceq=[];
options = optimset('LargeScale', 'off', 'Simplex', 'on');
options = optimset(options,'Maxiter',100000);
 [ans,fval,exitflag,output]=linprog(f,C,b,Ceq,beq,lb,ub,[],options)

if exitflag==1
    v=v+1;
    y(v)=fval;
%if bas==1 min=fval; end
if fval<min
    min=fval;
    sonuc=ans;
    tepe=bas
end
end
 end

 for i=1:n+1
    song(i)=sonuc(i)
end
song(n+2)=(sonuc(n+3)+ sonuc(n+2))/2
dogru=0;
load T.txt
s=length(T(:,1));
  for say=1:s
      top(say)=0;
    sinif(say)=0;


toplam=0;
for j=1:n
tk(say,j)=T(say,j)-t(tepe,j);
end

for k=1:n
```

```
toplam=toplam+abs(T(say,k)-t(tepe,k));
end
tk(say,n+1)=toplam;
tk(say,n+2)=-1;

 for j=1:n+2
        top(say)=top(say)+(tk(say,j)*song(j));

 end
if top(say)<=0
   sinif(say)=1;
else sinif(say)=2;
end %sinif sınıflar matrisini temsil ediyor
 end
 for i=1:s
    if sinif(i)==T(i,n+1) dogru=dogru+1;
    end
 end
 generalization=(100*dogru)/s;

 [generalization2,song2,sonuc1,tepe1,sinif2,dogru1]=svmhata2();
 load D1.txt;
 if generalization>generalization2 %veri kümeleri yer değiştirdiği için training
genelleştirmesi en iyi olan alınıyor
    ayiricifonk=song;
    genellestirme=generalization;
    son=sonuc;
    tepe=tepe;
 else
    ayiricifonk=song2;
    genellestirme=generalization2;
    son=sonuc1;
    tepe=tepe1;
    t=D1;
 end
toc;
```

f) Algoritma 6: Destek vektör Makinalarında Çokyüzlü Konik Fonksiyonlar ve kümeleme (DVM-ÇKF 2)

```
close all;clc;
tic;
load yeni.txt;
n=8;
s=0;
m=length(yeni(:,1));
R=yeni;
sinifno=1.0;
for i=1:m
   if R(i,n+1)~=sinifno
      s=s+1;
      d(s)=i;
   for j=1:n+1
    G(s,j)=R(i,j);
   end
   end
end
R(d,:)=[];
D=R
```

```
n=8;
v=0;

kalan=m;
anai=1;


 tic;
m=length(D(:,1));
p= length(G(:,1));
min=10000;
clustersayisi=2;

[IDX,t]=kmeans(D,clustersayisi);%clusterlar belirleniyor
tl=length(t(:,1));
m=length(D(:,1));
p= length(G(:,1));

 for bas=1:tl%herbir cluster için çalıştırılıo
for say=1:m
top=0;
for j=1:n
ak(say,j)=D(say,j)-t(bas,j);
end
    for j=n+1:n+3
if j==n+1
for k=1:n
top=top+abs(D(say,k)-t(bas,k));
ak(say,j)=top;
end
end
if j==n+2
ak(say,j)=-1;
end
if j==n+3
    ak(say,j)=0;
end
    end

    for k=n+4:n+3+m+p
      if k-say==n+3
          ak(say,k)=-1;
      else
          ak(say,k)=0;
      end

    end
end
for say=1:p
top=0;
for j=1:n
bk(say,j)=-(G(say,j)-t(bas,j));
end
    for j=n+1:n+3
if j==n+1
for k=1:n
top=top+abs(G(say,k)-t(bas,k));
bk(say,j)=-top;
end
end
```

```
 if j==n+2
bk(say,j)=0;
 end
 if j==n+3
    bk(say,j)=1;
 end
   end
   for k=n+4:n+3+m
      bk(say,k)=0;
   end
   for k=n+4+m:n+3+m+p
     if k-say==n+3+m
        bk(say,k)=-1;
     else bk(say,k)=0;
     end
   end
end
for i=1:m
   for j=1:n+3+m+p
   C(i,j)=ak(i,j);
   end
end
for i=m+1:m+p
   for j=1:n+3+m+p
      C(i,j)=bk(i-m,j);
   end
end
 for j=1:n+3
    if j<n+1
C(m+p+1,j)=0;
    end
       if j==n+2
       C(m+p+1,j)=1;
       end
       if j==n+3
       C(m+p+1,j)=-1;
       end
       if j>n+3
         C(m+p+1,j)=0;
       end
    end

 for h=1:m+p
    b(h)=-1;
 end
 b(m+p+1)=0;
for o=1:n+3
   if o<=n+1
      f(o)=0;
   end
if o==n+2
   f(o)=1;
end
   if o==n+3
      f(o)=-1;
   end
end
for o=n+4:n+3+m+p
   f(o)=1;
end
```

```
for i=1:n
    lb(i)=-10;
    ub(i)=10;
end
lb(n+1)=0.0;
ub(n+1)=inf;
lb(n+2)=1;
ub(n+2)=inf;
lb(n+3)=1;
ub(n+3)=inf;
for o=n+4:n+3+m+p
    lb(o)=0;
    ub(o)=inf;
end
beq=[];
Ceq=[];
[ans,fval,exitflag,output]=linprog(f,C,b,Ceq,beq,lb,ub,[])


if exitflag==1
    v=v+1;
    y(v)=fval;
if fval<min
    min=fval;
    sonuc=ans;
    tepe=bas
end
end
 end
 for i=1:n+1
    song(i)=sonuc(i)
end
song(n+2)=(sonuc(n+3)-sonuc(n+2))/2
toc;
```