Sınıflandırma Problemleri İçin Matematiksel Programlama Temelli Çözüm Yaklaşımları

Müge Acar
Yüksek Lisans Tezi

Endüstri Mühendisliği Ana Bilim Dalı
Temmuz 2015

JÜRİ VE ENSTİTÜ ONAYI

Müge Acar'ın "**Sınıflandırma Problemleri İçin Matematiksel Programlama Temelli Çözüm Yaklaşımları"** başlıklı **Endüstri Mühendisliği** Anabilim Dalındaki, Yüksek Lisans Tezi 20.07.2015 tarihinde aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

		Adı Soyadı	**İmza**
Üye (Tez Danışmanı):		**Prof. Dr. REFAİL KASIMBEYLİ**	**……..**
Üye	**:**	**Doç. Dr. ŞAFAK KIRIŞ**	**……..**
Üye	**:**	**Yard. Doç. Dr. GÜRKAN ÖZTÜRK**	**……..**

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …….. tarih ve …….. sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET
Yüksek Lisans Tezi

SINIFLANDIRMA PROBLEMLERİ İÇİN MATEMATİKSEL PROGRAMLAMA TEMELLİ ÇÖZÜM YAKLAŞIMLARI

Müge ACAR

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Refail KASIMBEYLİ

2015, 50 Sayfa

Yeni teknolojiler, araçlar ve yöntemler insan hayatını kolaylaştırmak amacıyla geliştirilmiş ve geliştirilmeye de devam etmektedir. Örneğin görüntü tanıma yöntemiyle kimlik tespiti, bir hastalığın teşhis edilmesi, el yazılarının metine dönüştürülmesi gibi konular araştırmacıların ilgisini çekmiş ve bu konularda daha hızlı ve doğru sonuç verecek yöntemler geliştirmeye yönlenmiştir. Bu çalışmada bu amaçlara hizmet edebilecek Çok Yüzlü Konik Fonksiyonlar algoritmalarına yeni matematiksel modeller eklenerek sınıflandırıcıların oluşturulması, özellikle de bu yaklaşım temelli yeni modeller geliştirilerek yaklaşımlarının incelenmesi ve yeni bir yaklaşım önerilmesi, son olarak da Çok Yüzlü Konik Fonksiyonlar temelli algoritmaların kümeleme temelli yaklaşımlarla geliştirilmesi üzerine çalışılmıştır. Süre ve başarı oranlarında verimlilik sağlanmaya çalışılmıştır.

Anahtar Kelimeler: Matematiksel Programlama, Veri Madenciliği, Sınıflandırma, Kümeleme Algoritmaları

ABSTRACT

Master Of Science Thesis

MATEMATICAL PROGRAMMING BASED SOLUTION APPROACHES FOR CLASSIFICATION PROBLEMS

Müge ACAR

Anadolu University

Graduate School of Sciences

Industrial Engineering Program

Supervisor: Prof. Dr. Refail KASIMBEYLİ

2015, 50 Pages

Technology and new approaches seaches to make how human life can be easier than before. Diagnosis of a disease, image processing, detecting spam mails before it received are some basic issues that some researchers are studying on and motivated to make new developments about. In this study we carried about generating new matematical models based polyhedral conic functions algorithms, seaching for classification. Also in this study we searched for generating clustering based polyhedral conic functions algorithms in the purpose of making development of process time and success percents.

Keywords: Mathematical programming, Data Mining, Classification, Clusturing

TEŞEKKÜR

Kendime birçok anlamda örnek aldığım bu çalışmamda beni her zaman motive eden, ondan daha öğrenecek çok şeyim olan sayın hocam Prof. Dr. Refail KASIMBEYLİ'ye, beni iş ve özel hayatımda da destekleyen hepsini çok sevdiğim çalışma arkadaşlarıma,

Her zaman ve her koşulda beni ilk önceliği yapan, hiçbir konuda desteğini esirgemeyen, hep yanımda olup, sabrıyla beni motive eden biricik eşim Özgür ACAR'a

Beni bu günlere fedakarlık ve üstün emeğiyle getiren, hep daha iyi bir insan ve başarılı bir birey olmam için hiçbir şeyi esirgemeyen, her zaman destekleyen canım annem Mübeccel SOYUÖZ'e

Ve son olarak da annesini, daha doğmadan bu tez aşamasında destekleyen minik oğlum Çınar ACAR'a

Teşekkürü bir borç bilir, yürekten sevgilerimi sunarım…

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1 GİRİŞ

İnsan doğası gereği hep yaşadığı dünyayı iyileştirmeye çalışma eğilimindedir. Bilim adamları tarafından geliştirilen teknolojiler, araçlar ve yöntemler ile insanların hayatını kolaylaştırmak amacıyla geliştirilmiş ve geliştirilmeye devam etmektedir. Örnek vermek gerekirse, herhangi bir hastalığın teşhis edilmesi, görüntü tanıma yöntemiyle kimlik tespiti, el yazılarının metine dönüştürülmesi gibi konular araştırmacıların ilgisini çekmiş ve araştırmacılar bu konularda daha hızlı ve doğru sonuç verecek yöntemler geliştirmeye yönlenmişlerdir. Bahsedilen görüntü tanıma yöntemiyle kimlik tespiti; askeri alanda, gizlilik arz eden, yüksek güvenlik gerektiren kurumlarda zaman ve güvenlik açısından daima bir gereksinim olmuş ve kurumlar araştırmacıları bu gereksinimler doğrultusunda ilerletmiştir. El yazılarının metne dönüştürülmesi alanı her türlü bilginin elektronik ortamda depolanmaya çalışıldığı günümüzde önemli bir gereksinim olarak görülüp, üzerine yapılan çalışmalar devam etmektedir. Bahsedilen, insan hayatını kolaylaştıran bu çalışma alanları her zaman birer gereksinim olarak sistemleştirilmiş ve geliştirilmeye de devam edecektir. Bu konularda yapılan araştırmalarda kullanılan başarılı yöntemlerden biri de sınıflandırma yaklaşımlarıdır.

Bu yaklaşımlar üzerine yapılan çalışmalar, Fisher'ın ayırma analizi ile başlayıp günümüzde de hala geliştirilmektedir. 1950'lerden itibaren ise sınıflandırma yaklaşımlarında matematiksel modelleme temelli yaklaşımlar geliştirilmeye başlanmıştır.

Tez kapsamında yapılan çalışmada kullanılacak yöntemlerin tespitinde literatür incelendiğinde farklı yaklaşımlara dayanan birçok algoritma karşımıza çıkmaktadır. Bu çalışmada, iki sınıflı sınıflandırma problemlerinde bir çözüm yöntemi olan Öztürk ve Gasimov [1] tarafından ortaya atılan Çok Yüzlü Konik Fonksiyonlar (ÇKF) olarak adlandırılan matematiksel model temelli bir algoritma temel alınarak, başarı oranlarının arttırılması ve çalışma sürelerinin kısaltılması üzerine yeni matematiksel modeller geliştirilmeye çalışılmış ve kullanılan modeller birbirleriyle kıyaslanmıştır. İkinci bölümde ise, algoritmada büyük öneme sahip başlangıç tepe noktasını daha etkin belirleyebilmek için literatürdeki farklı

yöntemler araştırılmış, farklı kümeleme algoritmaları kullanılmış ve bu algoritmalar da birbiriyle kıyaslanarak başarı oranı ve çalışma süresi olarak en iyi sonuç veren algoritma ve modeller kullanılarak yeni bir algoritma geliştirilmiştir.

Üçüncü bölümde, geliştirilen algoritmanın literatürdeki diğer algoritmalarla, başarı oranları ve çalışma süreleri açısından kıyaslanmış ve elde edilen sonuçlar tez kapsamında sunulmuştur.

Bu çalışma ile ilgili geliştirilen modeller ve bu modelleri içeren farklı kümeleme algoritmaları kullanan iç içe tasarlanmış olan algoritmaların genel değerlendirmeleri ve açıklamaları son bölüm olan sonuç ve öneriler bölümünde belirtilmiştir.

2 SINIFLANDIRMA PROBLEMLERİNİN ÇÖZÜMÜ İÇİN KULLANILAN YÖNTEMLER

Bu bölümde genel olarak ikili sınıflandırma probleminin literatürdeki çözüm yaklaşımlarına değinilmiştir. Ayrıca bu çözüm yaklaşımlarında kullanılan matematiksel modeller incelenmiş ve sonuçlar sunulmuştur.

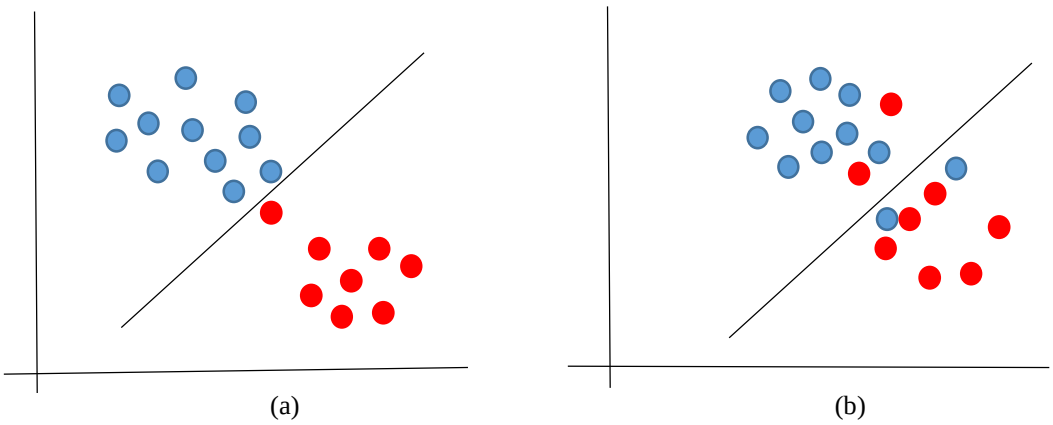
2.1 Giriş

Günümüz teknolojisine göre veri artık çok kolay elde edilip, yığınlar halinde saklanabilen, üzerine analizler yapılabilen bir nesne haline gelmiştir. Büyük boyutlardaki verilerin anlamlı bilgilere dönüştürülmesi olarak tanımlanan veri madenciliğinde en çok kullanılan tekniklerden birisi de sınıflandırma ve kümeleme teknikleridir.

Öztürk'e göre bir sınıflandırma problemi bir veri kümesinden seçilen eğitim kümesini kullanan belirli bir tanıma sisteminin, yani sınıflandırıcının geliştirilmesidir [2]. Bennett ve Mangasarian ise sınıflandırma problemlerini, eldeki verilerin belli özelliklerine göre alt kümelere atanması işlemi şeklinde tanımlamışlardır[3]. Sınıflandırma problemi, birden fazla sınıfa ait olduğu bilinen

verilerin hangi sınıfa ait olduğunu belli bir başarı oranıyla belirleyebilen bir tanıma sisteminin - sınıflandırıcının oluşturulması problemidir. Veri kümesindeki sınıfların sayısına göre problem iki sınıflı, üç sınıflı vb şeklinde tanımlanabilmektedir.

Matematiksel programlama temelli algoritmalar eldeki veri kümesinden seçilen ve eğitim kümesi olarak adlandırılan bir küme üzerinde "eğitilerek" oluşturdukları sınıflandırıcıyı daha sonra, genellikle rastgele oluşturulan test kümesi üzerinde sınarlar. Böylece algoritmanın oluşturduğu sınıflandırıcının başarı oranı belirlenmiş olur. Eğitim kümesinde doğru sınıflandırılan veri oranına eğitim kümesinin, test kümesindeki doğru sınıflandırılan veri oranına ise test kümesinin başarı oranı denir. Şekil 2.1'de iki sınıftan oluşan veriler ve bu verilerin bir sınıflandırıcıyla sınıflara ayrılması gösterilmiştir. Bir doğrusal ayırma işleminde sınıflandırıcılar doğrusal bir fonksiyon yardımıyla ayrılmıştır. Şekil 2.1 (a)'da veri kümesindeki noktalar %100 doğru ayrılmışken, Şekil 2.1 (b)'de ise yanlış ayrılan noktaların en küçüklenme örneğidir.



Şekil 2.1 Doğrusal Ayırma (a) Bir sınıflandırıcı aracılığıyla tüm noktaların birbirinden ayrılması sağlanmıştır. (b) Bir sınıflandırıcı aracılığıyla tüm noktalar yanlış noktalar en aza indirilerek ayrılması sağlanmıştır.

Sınıflandırma problemlerinin temelleri 1930'larda Fisher tarafından atılmıştır. Kullanılan diğer yöntemler; Bayes sınıflandırma, sinir ağları, genetik algoritmalar, en yakın komşu ve bulanık mantık, karar ağaçları, geri yayılımlı sinir ağları sayılabilir [1, 3, 4, 5, 6, 7, 8,]. Literatürde en çok tercih edilen yöntemlerden biri de destek vektör makineleridir [9].

Sınıflandırma problemleri için geliştirilen yöntemlerin değerlendirilmesindeki kriterler; başarı oranları, problem boyutları, iterasyon sayısı ve iterasyon süresi olarak belirtilebilir. 1960'lı yıllardan itibaren matematiksel programlama temelli algoritmalar sayılan bu kriterlere göre kullanılmaya başlanmış, diğer yaklaşımlarla rekabet edebilir sonuçlara ulaşılmıştır.

2.2 Matematiksel Programlama Yöntemleri

Yöntemlerin değerlendirilmesinde başarı oranı yüksekliğiyle öne çıkan matematiksel programlama temelli algoritmalar ile ilgili literatürde bulunan çalışmalar Bennet ve Mangasarian tarafından 1991 yılında geliştirilen gürbüz doğrusal programlama algoritmasıdır [3]. Bu çalışmada doğrusal fonksiyonlarla veriler sınıflara ayrılırken yanlış sınıflandırılan verileri en aza indirmek amaçlanmıştır.

Bir diğer çalışma ise Astorino ve Gaudioso'nun 2002 yılında birden fazla hiper düzlemle yaptığı sınıflandırma çalışmasıdır [8]. Bu çalışmada bir tolerans değeri belirlenmiş ve algoritma o tolerans değerine geldiğinde sonlanmaktadır.

Geliştirilen bir diğer yöntem ise Öztürk ve Gasımov 'ün 2006 yılında yapmış olduğu çalışması olan Çok Yüzlü Konik Fonksiyonlar temelli algoritmadır. Bu çalışmada sınıflandırmayı yapan konik fonksiyonlar tanımlanmıştır, başarı oranları eğitim kümelerinde %100 olarak elde edilmiştir [1].

Üney ve Türkay'ın 2005 yılında yaptığı çalışmada ise (hyperbox) hiperkutu adını verdikleri kümeler ile, çok sınıflı sınıflandırma problemi için bir matematiksel model geliştirilmiştir [4]. Amaçları ayrılan hatalı veri sayısı ve hiperkutu sayısını enküçüklemek olmuştur.

2.3 Çok Yüzlü Konik Fonksiyonlar ile İki Sınıflı Sınıflandırma

Bu tez kapsamında iki sınıflı sınıflandırma problemleri için bir konik yüzey, kümelerden birine ait maksimum sayıda noktaları "iç kısmında", diğer kümeye ait

noktaları ise dış kısmında tutacak şekilde oluşturulacaktır. İncelenen algoritmada eğitim kümesindeki noktaları hedef alarak öncelikle rassal bir başlangıç noktası belirlenmektedir. Bu başlangıç noktası oluşturulacak konik yüzeyin tepe noktası olacaktır. Algoritma bu tepe noktasını kullanarak, oluşturacağı konik yüzeyin içinde bir kümenin olabildiğince çok ayrılması istenen noktaları içine alan ve diğer noktaların dışarda kalmasını sağlayan uygun konik yüzeyleri oluşturarak sınıflandırma işlemini gerçekleştirecektir..

A ve B kümeleri R^n' de verilmiş iki küme iken algoritmanın adımları aşağıdaki gibidir:

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

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

Adım 0: $l = 1, I_l = I, A_l = A$ şeklinde belirleyip *Adım1*' e gidilir.

Adım 1: a^l noktası A_l noktasının herhangi bir noktası olmak üzere, P_l alt problemini çözülür.

$$(P_l) \quad \min\left(\frac{ye_m}{m}\right) \tag{2.1}$$

$$\omega(a^i - a^l) + \xi\|a^i - a^l\|_1 - \gamma + 1 \leq y_i \qquad \forall i \in I_l \tag{2.2}$$

$$-\omega(b^j - a^l) + \xi\|b^j - a^l\|_1 + \gamma - 1 \leq 0 \qquad \forall j \in J \tag{2.3}$$

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

Bu matematiksel modelin çözüm çıktıları P_l fonksiyonu için ω^l , ξ^l , γ^l, y^l parametrelerini vermektedir. A kümesindeki noktaları bitene kadar elde edilen l tane fonksiyon için $g_l(x)$ fonksiyonu bulunur.

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

Adım 2:

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

$Eğer\ A_l \neq \emptyset\ ise\ Adım1'$e gidilir.

Adım 3:

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

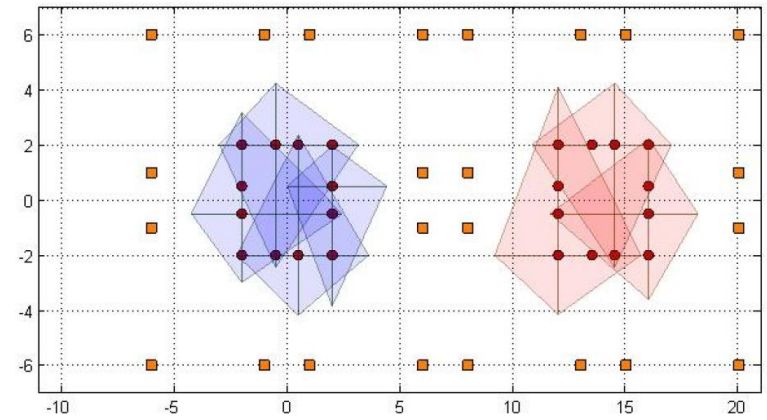
fonksiyonu tespit edilir ve durulur.

Şekil 2.2 ve Şekil 2.3 te çokyüzlü konik fonksiyonların iki boyutlu ve üç boyutlu ayırma şekilleri gösterilmiştir.

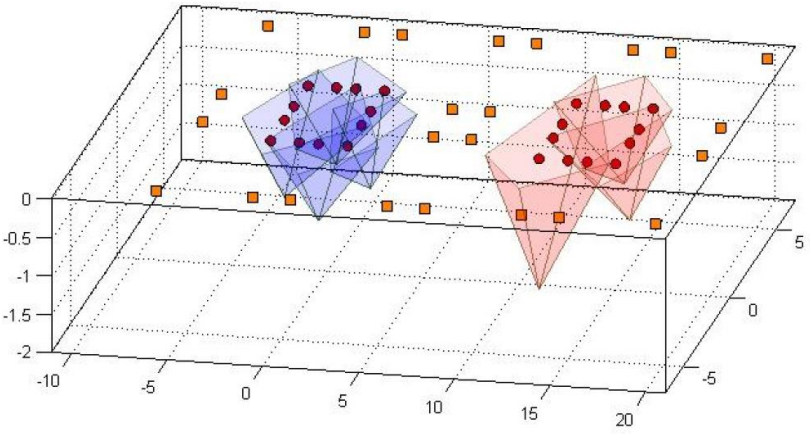
Amaç algoritmada eğitim kümesinde eğitilen sınıflandırıcıların test kümesinde de başarılı olmasını sağlamaktır.

Bu algoritma ile ayrılması istenen noktaların tamamı kesinlikle küme dışında bırakılmadığı için %100 e varan eğitim kümesi başarısı elde edilmektedir. Ancak algoritma son iterasyonlarda çalışırken, kümeler sadece bir ya da iki noktayı kapsadığı için test kümesi başarısını belli oranda düşürmektedir. Bu sebeple olabildiğinde test başarısını arttırmak bu tez kapsamında amaç olarak belirlenmiştir.

Bu algoritma son iterasyonlarda çalışırken, gürültü noktalar olarak bilinen noktaları ayırırken iterasyon sayısını arttırarak ayırmak zorunda kaldığı için toplam iterasyon sayısı da gereksiz yere artmaktadır. Bu durumda fazla iterasyonda oluşturulan konilerin içine B noktalarının girmesi ihtimali oluşmaktadır. Bu tez kapsamında iterasyon sayısını azaltmak için algoritma ve matematiksel modeller önerilmesi amaçlanmıştır.



Şekil 2.2 Çok Yüzlü Konik Fonksiyonlarda iki boyutlu ayırma [2]



Şekil 2.3 Çok Yüzlü Konik Fonksiyonlarda Üç Boyutlu Ayırma [2]

2.4 Kullanılan Çok Yüzlü Konik Fonksiyonlar Algoritmaları

Kullanılan matematiksel programlama yöntemi tüm problemlerde etkin sonuç vererek en iyiyi bulma imkanı verdiği için çalışmada öncelikle modeller üzerine odaklanılmıştır. Bölüm 2.4.4'te bahsedilecek ve bu tez kapsamında özgün olarak geliştirilmiş olan yapay değişkenli modelin ilk motivasyon kaynağı olan ve literatürde de kullanılmış olan bazı modeller aşağıda belirtilmiştir. [10,11,12]

2.5 Z model

Bu model Çiftçi [10], Çimen [11], tarafından kümeleme amaçlı kullanılmış ve başarılı kümeleme sonuçlarına ulaşılmıştır. Satı' nın çalışmasında ise, farklı bir katsayı kullanılarak sınıflandırma problemlerine entegre edilmiştir [12]. O çalışmada da başarılı sonuçlar elde edilmiştir. Bu tez kapsamında ise başarısı denenen ilk model olarak yer verilmiştir.

Algoritma A noktaları içinden öncelikle bir tepe noktası (A noktalarının skaler olarak toplamlarının en büyük değeri olarak) seçilir. Bir matematiksel model aracılığıyla, seçilen tepe noktasından B noktalarından içinde olabildiği kadar az, A noktaları maksimum olarak içeride kalacak şekilde çok yüzlü konik fonksiyonlar oluşturur. Bu fonksiyonlar, A noktalarının tamamı ayrılıp bitinceye kadar, farklı tepe noktaları seçilerek oluşturulur. Elde edilmek istenen fonksiyon ise bulunan tüm fonksiyonların noktasal en küçüğü hesaplanarak bulunur.

A ve B kümeleri R^n ' de verilmiş kümeler olsun:

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

$B = \{b^j \in R^n : j \in J\}$ $\qquad J = \{1, \ldots, n\}$

Adım 0: $l = 1, I_l = I, A_l = A$ şeklinde belirleyip *Adım1'e* gidilir.

Adım 1: a_l noktası A_l noktasının herhangi bir noktası olmak üzere, P_l alt problemini çözülür.

$$(P_l) \quad \min\left(\frac{ye_m}{m} + \frac{ze_n}{n}\right) \tag{2.4}$$

$$\omega(a^i - a^l) + \xi\left\|a^i - a^l\right\|_1 - \gamma + 1 \leq y_i \qquad \forall i \in I_l \tag{2.5}$$

$$-\omega(b^j - a^l) + \xi\left\|b^j - a^l\right\|_1 + \gamma - 1 \leq z_j \qquad \forall j \in J \tag{2.6}$$

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

Matematiksel modelin çözümü 2.4 denklemindeki P_l fonksiyonunun ω^l, ξ^l, γ^l, y^l, z^l parametreleri olmaktadır. A noktaları bitene kadar elde edilen l tane fonksiyon için $g_l(x)$ fonksiyonu bulunur.

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

Adım 2:

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

$Eğer\ A_l \neq \emptyset\ ise\ Adım1'$e gidilir.

Adım 3:

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

fonksiyonu tespit edilir ve durulur.

Z modelde elde edilmek istenen B noktalarının çizilen konilerin içine alınması durumunda başarı oranlarının nasıl değiştiğinin gözlemlenmesidir.

2.5.1 Epsilon Model

Algoritma A noktaları içinden öncelikle bir tepe noktası (A noktalarının skaler olarak toplamlarının en büyük değeri olarak) seçilir. Bir matematiksel model aracılığıyla, seçilen tepe noktasından B noktalarından içinde olabildiği kadar az, A

noktaları maksimum olarak içeride kalacak şekilde çok yüzlü konik fonksiyonlar oluşturur. Bu fonksiyonlar, A noktalarının tamamı ayrılıp bitinceye kadar, farklı tepe noktaları seçilerek oluşturulur. Elde edilmek istenen fonksiyon ise bulunan tüm fonksiyonların noktasal en küçüğü hesaplanarak bulunur. e sabit bir sayı olmak üzere,

A ve B kümeleri R^n ' de verilmiş kümeler olsun:

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

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

Adım 0: $l = 1, I_l = I, A_l = A$ şeklinde belirleyip *Adım1'* e gidilir.

Adım 1: a_l noktası A_l noktasının herhangi bir noktası olmak üzere, P_l alt problemini çözülür.

$$(P_l) \quad \min\left(\frac{ye_m}{m} + \frac{ze_n}{n}\right) \tag{2.7}$$

$$\omega(a^i - a^l) + \xi\left\|a^i - a^l\right\|_1 - \gamma + 1 \leq y_i \qquad \forall i \in I_l \tag{2.8}$$

$$-\omega(b^j - a^l) + \xi\left\|b^j - a^l\right\|_1 + \gamma - 1 \leq \varepsilon \qquad \forall j \in J \tag{2.9}$$

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

Matematiksel modelin çözümü 2.7 denklemindeki P_l fonksiyonunun ω^l , ξ^l , γ^l, y^l parametreleri olmaktadır. A noktaları bitene kadar elde edilen l tane fonksiyon için $g_l(x)$ fonksiyonu bulunur.

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

Adım 2:

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

$Eğer\ A_l \neq \emptyset\ ise\ Adım1'$e gidilir.

Adım 3:

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

fonksiyonu tespit edilir ve durulur.

İncelenen epsilon modelde amaç konilerin içine alınmayan B noktalarını belli bir marjinle sınırlayarak, belli oranda hataya izin verebilme ihtimaliyle test kümesindeki başarı oranlarını incelemektir.

2.5.2 Ağırlıklandırılmış Model

Algoritma A noktaları içinden öncelikle bir tepe noktası (A noktalarının skaler olarak toplamlarının en büyük değeri olarak) seçilir. Bir matematiksel model aracılığıyla, seçilen tepe noktasından B noktalarından içinde olabildiği kadar az, A noktaları maksimum olarak içeride kalacak şekilde çok yüzlü konik fonksiyonlar oluşturur. Bu fonksiyonlar, A noktalarının tamamı ayrılıp bitinceye kadar, farklı

tepe noktaları seçilerek oluşturulur. Elde edilmek istenen fonksiyon ise bulunan tüm fonksiyonların noktasal en küçüğü hesaplanarak bulunur.

A ve B kümeleri R^n' de verilmiş kümeler olsun:

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

$$B = \{b^j \in R^n : j \in J\} \qquad J = \{1, \ldots, n\}$$

Adım 0: $l = 1, I_l = I, A_l = A$ şeklinde belirleyip *Adım1'e* gidilir.

Adım 1: a_l noktası A_l noktasının herhangi bir noktası olmak üzere, P_l alt problemini çözülür.

$$(P_l) \quad \min\left(k\frac{ye_m}{m} + l\frac{ze_n}{n}\right) \tag{2.10}$$

$$\omega'(a^i - a^l) + \xi\left\|a^i - a^l\right\|_1 - \gamma + 1 \leq y_i \qquad \forall i \in I_l \tag{2.11}$$

$$-\omega'(b^j - a^l) + \xi\left\|b^j - a^l\right\|_1 + \gamma - 1 \leq z_j \qquad \forall j \in J \tag{2.12}$$

$$y = (y_1, \ldots, y_m) \in R_+^n \ , \qquad z = (z_1, \ldots, z_n) \in R_+^n,$$

$$\omega \in R^n, \qquad \xi \in R, \qquad \gamma \geq 1, k, l = (1,2 \ldots 10) \in Z$$

Matematiksel modelin çözümü 2.10 denklemindeki P_l fonksiyonunun ω^l , ξ^l , γ^l, y^l, z^l parametreleri olmaktadır. A noktaları bitene kadar elde edilen l tane fonksiyon için $g_l(x)$ fonksiyonu bulunur.

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

Adım 2:

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

$Eğer\ A_l \neq \emptyset\ ise\ Adım1'$e gidilir.

Adım 3:

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

fonksiyonu tespit edilir ve durulur.

Ağırlıklandırılmış modelde elde edilemek istenen, konilerle ayrılması istenen ve istenmeyen noktaları amaç fonksiyonunda belli oranlarda ağırlıklandırarak, algoritmayı A noktalarını içine almak isteme amacıyla B noktalarını dışarıda bırakmak isteme amacını dengelemektir.

2.5.3 Yapay Değişkenli Model

Algoritma A noktaları içinden öncelikle bir tepe noktası (A noktalarının skaler olarak toplamlarının en büyük değeri olarak) seçilir. Bir matematiksel model aracılığıyla, seçilen tepe noktasından B noktalarından içinde olabildiği kadar az, A noktaları maksimum olarak içeride kalacak şekilde çok yüzlü konik fonksiyonlar oluşturur. Bu fonksiyonlar, A noktalarının tamamı ayrılıp bitinceye kadar, farklı tepe noktaları seçilerek oluşturulur. Elde edilmek istenen fonksiyon ise bulunan tüm fonksiyonların noktasal en küçüğü hesaplanarak bulunur.

A ve B kümeleri R^n' de verilmiş kümeler olsun:

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

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

Algoritmanın adımlarını şu şekilde sıralayabiliriz.

Adım 0: $l = 1, I_l = I, A_l = A$ şeklinde belirleyip *Adım1'* e gidilir.

Adım 1: a_l noktası A_l noktasının herhangi bir noktası olmak üzere, P_l alt problemini çöz.

(P_l) $\min\left(\sum_i^m \frac{u_i}{m} + \sum_j^n \frac{v_j}{n}\right)$ (2.13)

$\omega(a^i - a^l) + \xi\|a^i - a^l\|_1 - \gamma + 1 \leq y_i$ $\forall i \in I_l$ (2.14)

$-\omega(b^j - a^l) + \xi\|b^j - a^l\|_1 + \gamma - 1 \leq z_j$ $\forall j \in J$ (2.15)

$\sum_{j=1}^J v_j \leq \phi$ (2.16)

$y_i/M \leq u_i \leq My_i$ $\forall i \in I$ (2.17)

$z_j/M \leq v_j \leq Mz_j$ $\forall j \in J$ (2.18)

$$y = (y_1, \dots, y_m) \in R_+^n, \quad z = (z_1, \dots, z_n) \in R_+^n, \quad \omega \in R^n,$$

$$\xi \in R, \qquad \gamma \geq 1, \qquad \phi \geq 0$$

Matematiksel modelin çözümü 2.13 denklemindeki P_l fonksiyonunun ω^l , ξ^l , γ^l, y^l, z^l parametreleri olmaktadır. A noktaları bitene kadar elde edilen l tane fonksiyon için $g_l(x)$ fonksiyonu bulunur.

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

Adım 2:

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

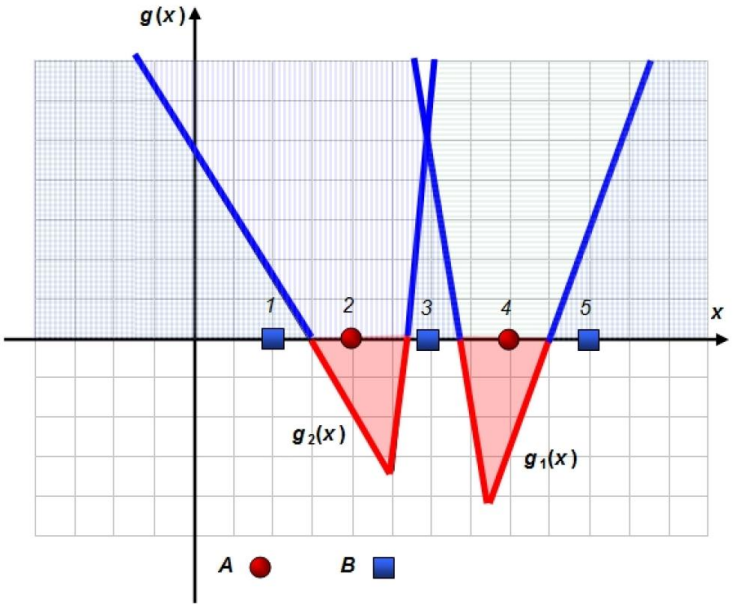
$Eğer\ A_l \neq \emptyset\ ise\ Adım1'e$ gidilir.

Adım 3:

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

fonksiyonu tespit edilir ve durulur.

Geliştirilen 2.13 denklemi amaç fonksiyonu olan matematiksel modelde amaç, konilerin içine giren B noktalarını sayısal olarak sınırlandırmak ve bu sayede testteki başarı oranını dengelemek yönünde oluşturulmuştur. Test başarıları modelde gösterilen *y* ve *z* karar değişkenleri ile sağlanmaktadır. Bu karar değişkenleri modelde mesafe belirtmektedir. Belirtilen mesafelerin detaylı gösterimi Şekil 2.4' te gösterilmiştir. Öztürk tarafından verilen Şekil 2.4'te belirtlen geometrik yazım kullanılarak açıklanmıştır. (bknz [2] sayfa 80)



Şekil 2.4 Çok Yüzlü Konik Fonksiyonların Karar Değişkeni Mesafeleri [2]

z ve *y* olarak belirlenmiş değişkenler, konilerin içindeki hatalı noktalarına ait vektörel bir uzaklığı temsil eden değişkenler *u* ve *v* olarak yapay değişkenlere dönüştürüldüklerinde hatalı olan noktaların uzaklık değişkeni olan *z* değerini *u* değişkeni için 1 hatasız olan negatif değerleri için ise 0 olarak belirlenmiştir. Aynı zamanda mevcut modelde amaç fonksiyonunda hatalı olan noktalara ait uzaklıkların toplamını en küçükleyen bir amaç fonksiyonu yerine hatalı olan nokta sayısını en küçükleyen (u ve v karar değişkenlerinden oluşan) bir amaç fonksiyonu kullanılmıştır.

2.5.4 Hesapsal Sonuçlar

Hesapsal sonuçlarda tez kapsamında araştırılan modeller göz önüne alındığında incelenen hangi modelin hangi yönlerden daha etkin olduğunu hesaplamak amacıyla farklı veri kümelerinde dört farklı model çalıştırılmıştır. Bu modeller kendi aralarında karşılaştırılmış ve etkin olan model tez kapsamında geliştirilmeye devam edilmiştir. Sonuçlar Çizelge 2.1 de karşılaştırmalı olarak verilmiştir.

Bu model verilerinin karşılaştırılmasında WBCD, Fertility, Liver, Ion verileri kullanılmıştır. Bu verilerin içerik bilgileri Bölüm 3.3 te detaylı olarak anlatılmıştır.

Çizelge 2.1 Modellerin Başarı Oranları

10 kez çapraz doğrulama

	WBCD		Fertility		Liver		Ion	
	Eğitim(%)	Test(%)	Eğitim(%)	Test(%)	Eğitim(%)	Test(%)	Eğitim(%)	Test(%)
Z model	80,34	71,56	96,98	86,00	52,16	49,48	100,00	69,04
Epsilon Model	68,27	62,48	-	-	-	-	-	-
A.Model 100/1	77,16	70,83	98,25	83,00	56,59	46,66	100,00	76,34
A. Model 60/40	77,58	70,94	96,64	84,00	55,34	58,82	100,00	77,14
Yapay Değişkenli Model	99,04	96,34	98,47	86,00	58,61	60,01	-	-
Algortima3	98,21	98,55	98,21	98,55	-	-	98,21	98,55
PCF Algoritması	100	100	100	90,45	100	68,40	100	95,76

Geliştirilen modeller, verilere göre analiz edildiğinde modellerin birbiri üzerine etkinliğine bakılmış ve yapay değişkenli modelin oldukça başarılı sonuçlar verdiği görülmüştür. Ancak yapay değişkenli modelin, yapay değişkenlerinden dolayı modelin iterasyon süresini çok arttırdığı gibi büyük boyutlu verilerde bazen sonuç vermemektedir. Buna göre modeller arasında ikinci derecede en iyi olan model Z modeldir. Z modelin, ÇKF algoritmasında kullanılan modelle farkı oluşturulan konik fonksiyonların, ayırmak istenen noktaları kapsamasını sağlayan sıkı kısıtı gevşeterek konik fonksiyonların hatalı noktaları da kapsamasına izin vererek test başarısını yükseltmektir.

Tez kapsamında geliştirilmek istenen algoritma Z model kullanılarak literatürde rekabet edecek başarı oranlarına ulaşmakta yeterli gelmediğinden, tez kapsamında yapılan çalışmalar ÇKF algoritmasında geliştirilecek yönler olarak tabir edebileceğimiz, test ve eğitim kümesinin başarı oranlarının birbirine yaklaştırılmasını, işlem sürelerinin kısaltılmasını geliştirmek için çalışmalar bu doğrultudadır.

Tez kapsamındaki çalışmanın devamında, yapay değişkenli modelin çok boyutlu veriler üzerinde nasıl etkinleştirilebileceği araştırılmıştır.

2.5.5 Sonuçlar

Sonuç olarak tez kapsamında yapılan çalışmanın 2. bölümünde ÇKF algoritması temel alınarak, algoritma geliştirilmeye çalışılmıştır. Algoritmanın zayıf yönleri tespit edilmiş ve bu yönler üzerinde çalışılmıştır.

Öncelikle ÇKF algoritması yapısı itibariyle sıkı kısıtları olan bir algoritmadır. Bu sıkı kısıtlar öncelikle gevşetilerek alınan sonuçlar gözlemlenmiştir. ÇKF algoritmasında başarı oranlarının birbirine yaklaştırılması için gereken koşulun, hatalı noktaları çok yüzlü konik fonksiyonlar kapsamayacak şekilde bir yol izlendiğinde çok başarılı eğitim kümesi sonuçları elde ederken, test kümesinde başarı oranı düşüklüğü gözlenmiştir. Bu durum iki başarı oranının arasındaki farkı arttırmaktadır. Bu sebeple ilk olarak Z modelde bu kısıt değiştirilerek hatalı noktalarında çok yüzlü konik fonksiyonlar tarafından kapsanmasına izin vermek için bir karar değişkeni değişken tanımlanmış ve amaç fonksiyonuna bu değer en küçüklenmesi için eklenmiştir. Sonuç olarak başarı oranları ÇKF algoritmasına göre oldukça düşük çıkmıştır. Bu durumun sebebi hatalı noktaların oluşturulan konik fonksiyonlar tarafından daha fazla kapsanma eğiliminden kaynaklanmaktadır.

İkinci olarak belirtilen etkiyi aşmak için Epsilon model olarak tanımladığımız model denenmiş, modelin sonuçları gözlenmiştir. Epsilon modelin kurulmasındaki amaç, konik fonksiyonların kapsadığı hatalı nokta değerlerini belli bir değerde sınırlandırma eğilimi olmuştur. Bu model sonucunda ise başarı oranları daha da düşmüştür. Başarı oranlarının düşme sebebi ise, konik fonksiyonların kapsadığı hatalı noktaların yanında, iterasyon sayısı artmış ama başarılı konik fonksiyonlar oluşturulamamıştır. Ayrılmak istenilen noktalar fonksiyonların dışında kalma eğilimine girmiştir. Bu modelin sonucunda ise amaç fonksiyonu değerlerini ağırlıklandırmanın bu soruna çözüm olabileceği düşünülmüş ve buna göre ağırlıklandırılmış model denenmiştir. Bu modelin geliştirilme amacı ise konik fonksiyonların kapsamaması istenen ve kapsaması istenen hatalı noktaların karar

değişkenlerini ifade eden amaç fonksiyonu değerlerini ağırlıklandırarak, her bir konik fonksiyonun hata eğilimini dengelemek olmuştur. Bir sonraki iterasyonda konik fonksiyonların kapsaması mümkün olmayan hata noktalarının karar değişkeni ifadelerinin ağırlıkları az, konik fonksiyonların kapsadığı hata noktalarının karar değişkeni ifadelerinin ağırlıklarının ise daha fazla oranda ağırlık verilerek model denenmiştir. Farklı ağırlıklarla denemeler yapılmış ve bunların sonucunda ise, başarı oranlarının çok değişmediği gözlenmiştir. Daha sonra bu modelin de zayıf yönleri düşünülerek geliştirilen son model oluşturulmuştur.

Yapay değişkenli modelin geliştirilme çalışmasının motivasyon noktası hem epsilon model hem de ağırlıklandırılmış model olmuştur. Algoritmanın model aracılığıyla konik fonksiyonların kapsadığı hatalı nokta sayısını kontrol edebilmesi için yöntemler üzerine düşünülmüştür. Kısıtlarda belirtilen karar değişkenleri noktaların konik fonksiyonlara olan uzaklıklarını belirten değişkenlerdir. Amaç fonksiyonunda hatalı noktaları belirten karar değişkenleri ifadeleri birbirine eklendiğinde aslında toplam hatalı nokta sayısını belirtmemektedir. Toplam hatalı nokta sayısını belirtmek için her bir uzaklık belirten karar değişkeni bir ikil değişken olarak yeni kısıtlar altında tanımlanmış ve hatalı nokta sayıları farklı değişkenlerle belirtilmiştir. Daha sonra konik fonksiyonların kapsamasını istemediğimiz karar değişkenine ait değeri belli parametrelerle sınırlandırarak model oluşturulmuş. Aynı zamanda amaç fonksiyonunda kullanılan hatalı noktaların toplam uzaklıklarını en küçüklemek yerine toplamdaki hatalı nokta sayısını en küçüklemek olarak amaç fonksiyonu yenilenmiştir. Bu model diğer denenmiş tüm modellere göre daha etkin sonuçlar vermiş hatta literatürle kıyaslandığında başarılı olmuştur. Ancak sadece küçük verilerde sonuçlara ulaşılabilmiştir. Veri özellik sayısı 10'u aştığında sonuç vermemiştir. Bu sebeple tez kapsamında kullanılan diğer model olan Z model ile daha etkin bir ÇKF algoritması çalışması yapılması için çalışılmıştır. İzleyen bölümde ise bu algoritmadaki diğer bir geliştirilmeye uygun olan rassal seçilen konik fonksiyonların tepe noktalarını nasıl daha etkin seçilebilir sorusuna yanıt aranmış ve kümeleme algoritmaları araştırılmıştır.

3 İKİLİ SINIFLANDIRMA PROBLEMLERİNDE KÜMELEME TEMELLİ YAKLAŞIMLAR

3.1 Giriş

İkili sınıflandırma problemlerinin çözümünde geliştirilen yöntemlerde elde edilen sonuçlara göre verimliliği arttırmak için farklı yöntemler geliştirerek çözüm aranması araştırmacıların çokça yöneldiği bir çalışma anlayışı olmuştur. En çok kullanılan temel yaklaşımlardan biri de kümeleme temelli yaklaşımlar olmuştur. Bu tez kapsamında kullanılan sınıflandırma algoritmasında, iyileştirilecek yönlerin belirlenmesi aşamasında, konik fonksiyonların başlangıç noktalarının belirlenmesi aşamasının, sonuçların verimliliğinde önemli rol oynadığı, bu tepe noktalarının rassal olarak belirlenmesi yerine kullanılabilecek daha etkin yöntemler arasında kümeleme yaklaşımları ve bu yaklaşıma ait çözüm yöntemleri üzerinde durulmuştur.

3.2 Literatür Özeti

İkili sınıflandırma problemlerinin çözümünde her bir iterasyon için başlangıç noktası seçiminde kullanılacak olan yöntemin seçilmesinde literatürde birçok çalışmaya rastlanabilir. Kümeleme yaklaşımlarından en bilinen yöntem MacQuen [13] tarafından literatüre kazandırılmış olan k-ortalamalar algoritmasıdır. Bagirov'un [14] geliştirmiş olduğu bütünsel k-ortalamalar algoritması ile başarılı sonuçlar elde edilmiştir. Çiftçi [10], çalışmasında kümeleme için kullandığı çok yüzlü konik fonksiyonlar algoritmasında rassal olan tepe noktası seçimi için kümeleme yaklaşımlarından k-ort yöntemini benimsemiş ve bu yöntemle başarılı sonuçlar elde etmiştir. Aynı zamanda Çimen [11] yaptığı çalışmada geliştirilmiş bütünsel k-ortalamalar kümeleme algoritmalarını denemiş ve yine başarılı sonuçlar elde edilmiştir. Satı [12] yaptığı çalışmasında çokyüzlü konik fonksiyonlar

algoritmasında yine kümeleme algoritmalarından k-ortalamalar algoritmasını kullanarak iterasyon süresinde iyileştirmeler yapmıştır. Bu çalışmalarda elde edilmiş olan başarılı sonuçların motivasyonu ile çok yüzlü konik fonksiyonlar algoritmasının model aşamasında denenen ve algoritmanın performansını, çözüm süresini ve büyük veri kümelerinde de kullanım gibi yönlerini geliştirebilmesi aşamasında incelenen çalışmalar ışığında da kümeleme yaklaşımlarının farklı çözüm yöntemlerinin denenmesi üzerine bu tez kapsamında araştırmalar yapılmıştır.

3.3 Kümeleme

Kümeleme bir veri kümesini belli özelliklerine göre gruplamak olarak tanımlanabilir. Kümeleme birçok alanda verileri kümelemek için kullanılabilir. Birçok farklı temelli yönteme kümeleme yapmak mümkündür. Bunlar ilişki temelli kümeleme, merkez nokta temelli kümeleme, dağılım temelli kümeleme, yoğunluk temelli kümeleme olarak literatürde sınıflandırılmıştır.

3.3.1 İlişki Temelli Kümeleme Algoritmaları

Hiyerarşik yöntemler nesneleri dendrogram denilen ağaç yapısı şeklinde gruplandırma temeline dayanır. Hiyerarşik yöntemler k değerinden bağımsızdır. Ağaç yapısı oluşturma işleminin ne zaman durdurulacağını belirten eşik değeri parametresine gereksinimleri vardır.

3.3.2 Yoğunluk Temelli Algoritmalar

Yoğunluk tabanlı yöntemler, nesnelerin doğal dağılımını yoğunluk fonksiyonu aracılığı ile tespit ederek eşik yoğunluğunu aşan bölgeleri küme olarak adlandırırlar. Düzgün şekilli olmayan kümeleri bulma başarısı ve istisnalardan etkilenmemesi ile başarılı kümeleme yöntemlerindendir.

3.3.3 Matematiksel Model Temelli Algoritmalar

Eldeki verileri bir matematiksel model ile ifade etmeye çalışırlar. Model tabanlı yöntemler iki temel yaklaşımı kullanırlar; istatistik yaklaşım ve yapay zekâ yaklaşımıdır.

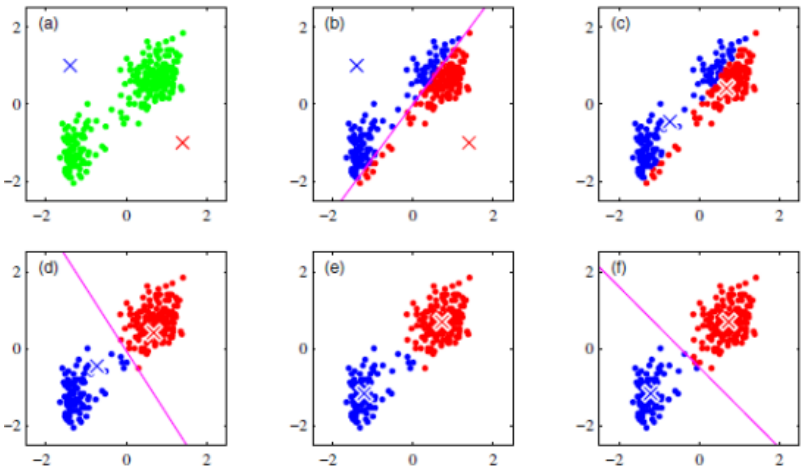
3.3.4 Ağırlık Merkezi Temelli Algoritmalar

Bölümleme yöntemleri, n adet nesneden oluşan veri tabanını giriş parametresi olarak belirlenen k adet bölüme (k<=n) ayırma temeline dayanır. Veri tabanındaki her bir eleman farklılık fonksiyonuna göre k adet bölümden birine dâhil edilir. Bu bölümlerden her biri bir küme olarak adlandırılır.

Bu kümeleme yöntemleri incelendiğinde Çok Yüzlü Konik Fonksiyonlar algoritmasında başlangıç tepe noktalarının belirlenmesinde kullanılacak yöntem olarak merkez nokta temelli kümeleme algoritmalarından birinin uygun olacağına karar verilmiştir. Bu kararın sebebi ise kullanılacak kümeleme yöntemi için gerekli olan esas olarak hangi noktaların hangi kümeye ait olacağına karar vermek değil, bu noktaları en az sayıda çevreleyecek fonksiyonu elde etmeyi sağlayacak merkez noktalarını elde etmektir. Bu sebeple merkez nokta algoritmaları bu çalışmada kullanılmak üzere uygun bulunmuştur.

k-ort Algoritması: k-ort algoritması, tüm veri kümesini k tane küme merkezine olabilecek en yakın uzaklıkların karesini atayarak, bu merkez noktalarını bulmayı amaçlamaktadır. Bu algoritma uygulaması oldukça basit ve sık kullanılan bir algoritmadır. Algoritmada öncelikle k (elde edilecek küme sayısı) parametre olarak girilmelidir. Birinci adımda algoritma k adet merkez noktasını rassal olarak belirler, daha sonra diğer kalan noktalar için, belirlenen merkez noktalara öklid uzaklığına göre en yakın merkez noktaya göre atar. Fakat elde edilen çözüm rassal olarak seçildiği için her bir kümedeki nokta için Öklid uzaklığına göre küme noktaları tekrar hesaplanarak en iyi merkez noktaya karar verilir. Bu adım küme merkezleri

değişmeyene kadar devam eder. Şekil 3.1'te k-ort algoritmasının şekilsel gösterimi detaylı olarak belirtilmiştir.



Şekil 3.1 k-ort algoritması ile kümeleme [15]

Algoritmanın adımları daha ayrıntılı olarak açıklamak gerekirse,

Eldeki veriler $(x_1, x_2, \dots, x_n)$ n adet d boyutlu olsun. k küme sayısı olsun ve k≤n olmak üzere kümeler $S = \{S_1, S_2, \dots, S_k\}$

Adım 1: Küme sayısı kadar rassal merkez noktası belirlenir.

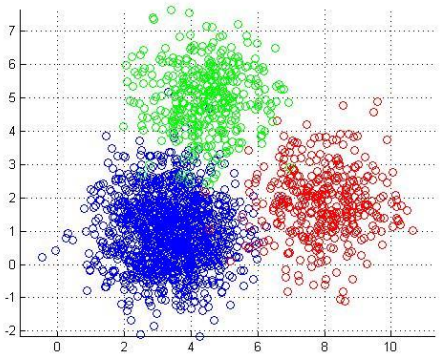
Adım 2: Öklid uzaklığına göre her noktayı en yakın merkeze atanır.

$$s_i^t = \{x_p : \|x_p - m_i\| \leq \|x_p^t - m_j^t\| \forall\, 1 \leq j \leq k\}$$

Adım 3: Yeni küme merkezleri seçilerek küme merkezlerine göre tüm noktaların uzaklıkları hesaplanır.

$$m_i^{(t+1)} = \frac{1}{|s_i^t|} \sum_{x_j \in s_i^t} x_j$$

Adım 4: Hesaplanan bu değer daha iyisi bulunana kadar tüm merkezler için devam eder. Yoksa durulur.



Şekil 3.2 k-ort algortimasıyla kümelemede gürültü noktalar [15]

Bu algoritma hızlı bir şekilde çalışır. Performansı büyük veri kümelerinde yüksektir. En iyi çözümü garanti etmez. Yeni küme merkezlerini seçerken kümenin tüm elemanlarına göre ortalama aldığı için gürültü noktaların etkisinde kalır. Bu da küme merkezlerinin en iyi çözümü garanti etmemesinde başlıca nedenlerden biridir. Şekil 3.2'de k-ort algoritması kullanılarak kümelemenin sonucu örnek olarak gösterilmiştir.

k-medoid Algoritması: k medoid algoritması temel mantığı olarak k ort algoritmasına benzese de küçük bir farklılık ile bazı yöntemlerde daha etkin rol oynadığını söyleyebiliriz. k-ort algoritması gibi öncelikle k adet merkez küme noktası rassal olarak seçilir. Daha sonra kalan noktalar yine Öklid uzaklığına göre merkez noktaya yakınlık koşuluna göre merkezlere atanır. Şekil 3.3'de k-medoid algoritmasıyla kümeleme ayrıntılı olarak gösterilmiştir.

Algoritmanın adımları:

Adım 1: Küme sayısı kadar rassal merkez noktası belirlenir.

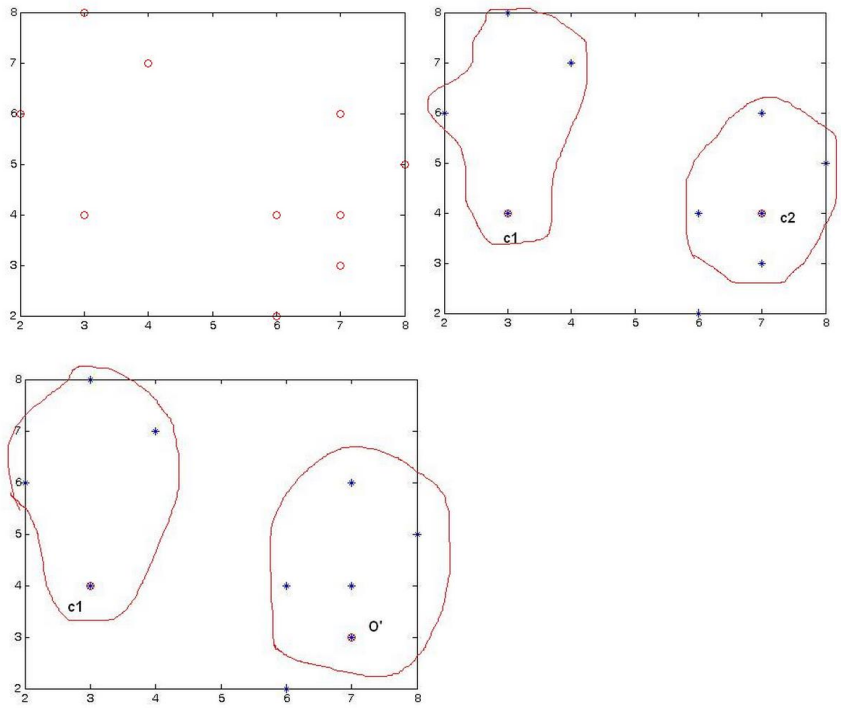
Adım 2: Öklid uzaklığına göre her noktayı en yakın merkeze atanır.

$$s_i^t = \{x_p : \|x_p - m_i\| \leq \|x_p^t - m_j^t\| \forall\ 1 \leq j \leq k\}$$

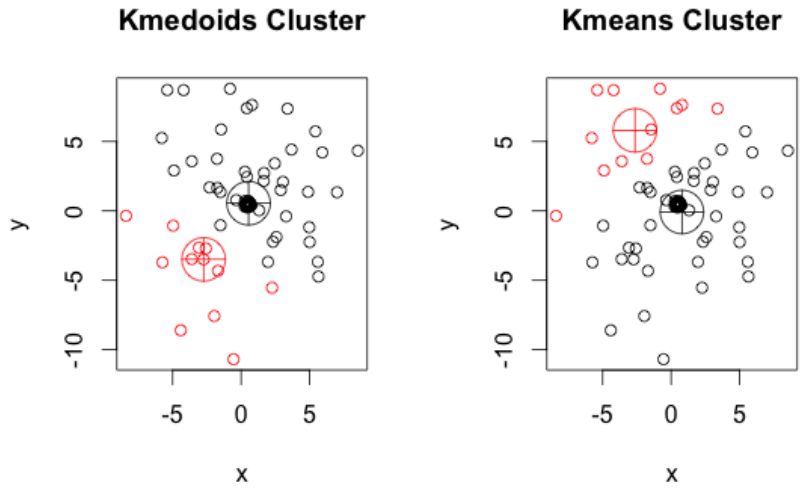
Adım 3: Yeni küme merkezleri hesaplanır.

$$m_i^{(t+1)} = \frac{1}{|s_i^t|} \sum_{x_j \in S_i^t} x_j$$

Adım 4: Hesaplanan bu değer daha iyisi bulunana kadar tüm merkezler için devam eder. Yoksa durulur.



Şekil 3.3 k-medoid Algortiması ile Kümeleme [16]



Şekil 3.4 k-medoid ve k-ort Algoritmalarının Farkı [15]

Rassal atanan merkezlerin iyileştirmesi amaçlı k-ort algoritması tüm noktaların Öklid uzaklıklarının ortalamalarına göre en iyi merkezi seçerken, k medyan algoritması sadece uzaklıkları dikkate alarak daha iyi olan merkez noktasını seçer. Bahsedilen farklılık bu iki algoritmada farklı sonuçlar elde etmeye neden olur. Bu farklı sonucun etkisi ise gürültü noktalarının etkisi olmaktadır. k-medoid

algoritması en iyi merkezi seçerken ortalamaları değil sadece uzaklıkları dikkate aldığı için merkeze olan öklid uzaklığı fazla olan bir nokta, az olan bir noktayla aynı oranda etkilenmediği için seçilen en iyi merkez noktası gürültü noktaya çok fazla yaklaşmaz ve böylelikle gürültü noktalardan k-ort algoritması kadar etkilenmez. Şekil 3.4'de iki algoritmanın farkı belirtilmiştir.

3.4 Geliştirilen Kümeleme Temelli ÇKF Algoritmaları

Önceki bölümde anlatılmış olan iki farklı algoritma da ÇKF algoritmasında rassal olarak belirlenen ÇKF algoritmasındaki tepe noktalarını bulmayı amaçlamıştır. Bu hedefe yönelik olarak k-ort ÇKF algoritmasının ikili sınıflandırma problemlerine uyarlanmıştır. Adımları ayrıntılı olarak belirtmek gerekirse,

3.4.1 k-ort ÇKF Algoritması

Bu algoritma ÇKF Algoritmasını temel alarak başlangıç çözüm için ihtiyaç duyulan konilerin tepe noktalarının rassal olarak belirlenmesi yerine k-ort Algoritmasıyla belirleyerek, daha hızlı bir çözüm üretmektedir. Bu algoritmanın adımlarının açıklaması ise;

A ve B kümeleri R^n ' de verilmiş kümeler olsun:

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

$B = \{b^j \in R^n : j \in J\}$ $\qquad J = \{1, \ldots, n\}$

Adım 0: Rassal olarak k adet merkez noktası belirlenir.

Adım 1: Her bir nokta için merkez noktaları arasındaki uzaklık aşağıda belirtilen formüle göre (öklid uzaklığına göre) belirlenir.

$$s_i^t = \{x_p : \|x_p - m_i\| \leq \|x_p^t - m_j^t\| \forall\ 1 \leq j \leq k\}$$

Adım 2: Belirlenen merkez noktaları her bir küme için tekrar rassal olarak seçilir ve yeni merkez noktası için yine her bir noktaya göre öklid uzaklığı alınır. Ortalama öklid uzaklıklarının ortalaması ile kıyaslanır. Daha küçük olan değer yeni merkez noktası ile eskisiyle değiştirilir. Eğer daha küçük olan değer eski nokta ise yeni bir rassal merkez noktası seçilerek tekrar eski nokta ile değiştirilir. Tüm noktalar bitip, en iyi değer değişmeyinceye kadar merkez noktaları iyileştirilir.

$$m_i^{(t+1)} = \frac{1}{\left|s_i^t\right|} \sum_{x_j \in S_i^t} x_j$$

Adım 3: Belirlenen bir küme için merkez noktası

$l = 1, I_l = I, A_l = A$ noktaları şeklinde belirleyip *Adım1' e* gidilir.

Adım 4: a_l noktası seçilen merkez noktası olarak P_1 alt problemini çözülür.

$$(P_l) \quad \min\left(\frac{ye_m}{m} + \frac{ze_n}{n}\right) \tag{3.1}$$

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

$$-\omega\left(b^j - a^l\right) + \xi\left\|b^j - a^l\right\|_1 + \gamma - 1 \leq z_j \qquad \forall j \in J \tag{3.3}$$

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

Bu matematiksel modelin çözümü P_l fonksiyonunun ω^l , ξ^l , γ^l, y^l parametreleri olmaktadır. A noktaları bitene kadar elde edilen l tane fonksiyon için $g_l(x)$ fonksiyonu bulunur.

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

Adım 5:

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

$Eğer\ A_l \neq \emptyset\ ise\ Adım1'$e gidilir.

Adım 6:

$g(x)$ fonksiyonu tespit edilir ve durulur.

Bu tez çalışmasında geliştirilmiş k-ort bütünleşik algoritması kullanılırken en iyi küme sayısı bulunmamış, tıpkı k-ortalamalar algoritmasında olduğu gibi kümeleme problemi çözülmeden önce küme sayısı belirlenmiştir.

3.4.2 k-medoid ÇKF Algoritması

Bu algoritma da tıpkı k-ort algoritmasının ÇKF algoritmasındaki rassal tepe noktası seçiminde kullanıldığı gibi kullanılmıştır. Rassal olarak belirlenen ÇKF tepe noktalarını kümeleme algoritmalarıyla çözümündeki verimliliği ölçmek ayrıca hangi algoritmanın hangi kriterlere göre daha iyi sonuç verdiği incelenmiştir.

K-medoid çkf Algoritmasının adımları:

A ve B kümeleri R^n ' de verilmiş kümeler olsun:

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

$B = \{b^j \in R^n : j \in J\}$ $\qquad J = \{1, \ldots, n\}$

Adım 0: Rassal olarak k adet merkez noktası belirlenir.

Adım 1: Her bir nokta için merkez noktaları arasındaki uzaklık aşağıda belirtilen formüle göre (öklid uzaklığına göre) belirlenir.

$$s_i^t = \{x_p : \|x_p - m_i\| \leq \|x_p^t - m_j^t\| \forall\ 1 \leq j \leq k\}$$

Adım 2: Belirlenen merkez noktaları herbir küme için tekrar rassal olarak seçilir ve yeni merkez noktası için yine herbir noktaya göre öklid uzaklığı alınır. Ortalama öklid uzaklıkları kıyaslanır. Daha küçük olan değer yeni merkez noktası ile eskisiyle değiştirilir. Eğer daha küçük olan değer eski nokta ise yeni bir rassal merkez noktası seçilerek tekrar eski nokta ile değiştirilir. Tüm noktalar bitip, en iyi değer değişmeyinceye kadar merkez noktaları iyileştirilir.

$$m_i^{(t+1)} = \sum_{x_j \in S_i^t} x_j$$

Adım 3: Belirlenen bir küme için merkez noktası

$l = 1, I_l = I, A_l = A$ noktaları şeklinde belirleyip *Adım1*'e git

Adım 4: a_l noktası seçilen merkez noktası olarak P_1 alt problemini çöz.

$$(P_l) \quad \min\left(\frac{ye_m}{m} + \frac{ze_n}{n}\right) \tag{3.4}$$

$$\omega(a^i - a^l) + \xi\|a^i - a^l\|_1 - \gamma + 1 \leq y_i \qquad \forall i \in I_l \tag{3.5}$$

$$-\omega\left(b^j - a^l\right) + \xi\left\|b^j - a^l\right\|_1 + \gamma - 1 \leq z_j \qquad \forall j \in J \qquad (3.6)$$

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

Bu matematiksel modelin çözümü P_l fonksiyonunun ω^l , ξ^l , γ^l, y^l parametreleri olmaktadır. A noktaları bitene kadar elde edilen l tane fonksiyon için $g_l(x)$ fonksiyonu bulunur.

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

Adım 5:

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

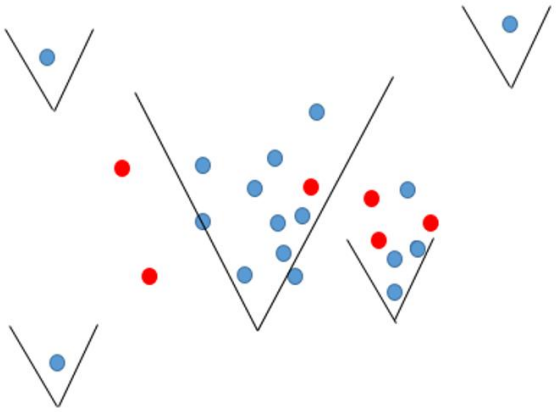
$Eğer\ A_l \neq \emptyset\ ise\ Adım1'e$ gidilir.

Adım 6:

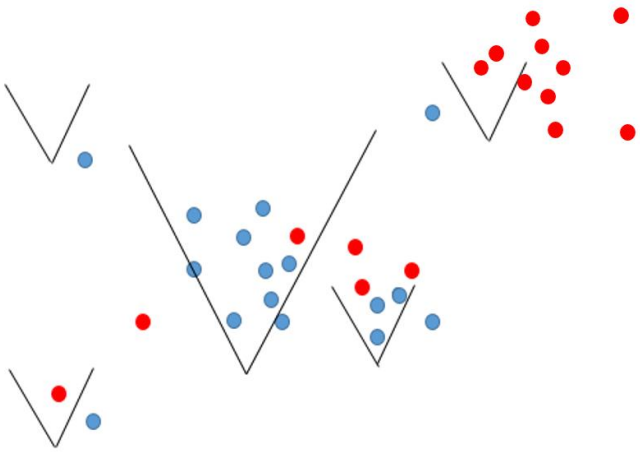
$g(x)$ fonksiyonu tespit edilir ve durulur.

3.4.3 Geliştirilen Toleranslı ÇKF Algoritması

Kümeleme algoritmaları ne kadar etkin tepe noktası seçerse seçsin, eğer veri kümesindeki tüm noktaları (gürültü noktalar özellikle) kapsama eğiliminde olursa A gürültü noktalarının durumu değişmez. Bu da başarı oranını değiştirmeyebilir. Bu sebeple geliştirilen bu algoritmada gürültü A noktalarından ödün vererek, test kümesinin başarı oranı arttırılmaya çalışılmıştır. Şekil 3.5 ve Şekil 3.6'da eğitim ve test kümelerinde A gürültü noktaların ÇKF algoritmasıyla nasıl ayrıldığı şekilsel olarak gösterilmiştir. Bu etkiyle Geliştirilen Toleranslı ÇKF algoritmasının avantajları şekilsel olarak belirtilmiştir.



Şekil 3.5 Eğitim Kümesi ÇKF



Şekil 3.6 Test Kümesi ÇKF

Adım 0: $l = 1, I_l = I, A_l = A$ şeklinde belirleyip *Adım1*'e gidilir.

Adım 1: a_l noktası A_l noktasının herhangi bir noktası olmak üzere, P_1 alt problemini çöz.

$$(P_l) \quad \min\left(\frac{ye_m}{m} + \frac{ze_n}{n}\right) \tag{3.7}$$

$$\omega\left(a^i - a^l\right) + \xi\left\|a^i - a^l\right\|_1 - \gamma + 1 \leq y_i \qquad \forall i \in I_l \tag{3.8}$$

$$-\omega\left(b^j - a^l\right) + \xi\left\|b^j - a^l\right\|_1 + \gamma - 1 \leq z_j \qquad \forall j \in J \tag{3.9}$$

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

Bu matematiksel modelin çözümü P_l fonksiyonunun ω^l, ξ^l, γ^l, y^l parametreleri olmaktadır. A noktaları bitene kadar elde edilen l tane fonksiyon için $g_l(x)$ fonksiyonu bulunur.

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

Adım 2:

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

$Eğer\ \{Toplam\ nokta - ayrılan\ nokta\ sayısı\} > tolerans\ değeri\ ise\ Adım1'e$ gidilir.

Adım 3:

$g(x)$ fonksiyonu tespit edilir ve durulur.

3.5 Hesapsal Sonuçlar

k-ort ÇKF algoritması ve k-medoid ÇKF algoritması farklı örneklere sahip çeşitli veri setlerinde denenmiş ve sonuçlar Çizelge 3.5'te belirtilmiştir. Veri setleri *UCI* Makine Öğrenmesi Veri Tabanları' ndan [**12**] elde edilmiştir. Test edilen veri setleri WBCD, Ionosphere, Fertility, Liver isimleriyle kaynaktaki aynı isimlerle belirtilmiştir.

Ionoshphere Veri Kümesi: Veritabanında bulunan veriler Labrador yarımdasında bulunan bir sistem tarafından toplanmıştır. Bu sistem toplam 6,4 kilovatlık iletim gücü olan 16 tane yüksek frekanslı antenin bir dizisinden oluşmaktadır. İyonosferdeki serbest elektronlar bu çalışmanın hedefidir. Radarın "iyi" olarak nitelendirdiği dönütler iyonosferde bazı yapı tiplerinin kanıtıdır. "Kötü" dönütler ise sinyalleri iyonosferi geçip gittiği için herhangi bir yapının kanıtını göstermez. Kanıtları, titreşim süresi, titreşim sayısı olan sinyaller bir otokorelasyon fonksiyonu kullanılarak işlenir.

Wisconsin göğüs kanseri tedavi süreci veri kümesi: Wisconsin Üniveristesi Hastanesi'nden elde edilen göğüs kanseri hastalarını iki yıl boyunca tedavi süreçlerinin izlenmesi ile ilgili veriler yer almaktadır. Verilerde tedaviye olumlu ve olumsuz yanıt veren hastaların verileri bulunmaktadır.

BUPA karaciğer bozuklukları: BUPA veri kümesi altı tane sayısal niceliği olan 345 tane bekar erkeğe ilişkin verileri içermektedir. Niteliklerin beşi, karaciğer bozukluğu nedeniyle ile ilgili olduğu düşünülen kan testleri; diğeri de her gün içilen alkollü içecek sayısıdır.

Üreme Veri Seti: 100 gönüllüden alınan WHO 2010 kriterlerine göre erkeklerdeki üreme etkinliğini etkileyen 10 farklı faktörün, üremeye etkisinin gözlemlendiği bir veri kümesidir.

Çizelge 3.1 Kullanılan Veri Kümelerinin Özellik ve Veri Sayıları

Veri Kümeleri	Veri Sayısı	Özellik
WBCD	683	9
Fertility	100	10
Ionosphere	351	34
Liver	345	6

Detayları da belirtilen verilerin toplam sayıları ve özellik sayıları da aşağıdaki Çizelge 3.1'de verilmiştir.

Her bir veri seti için geliştirilmiş olan iki algoritma da farklı k sayılarıyla öncelikle işlem süreleri bakımından verinin tamamı ile çalıştırılarak işlem süreleri ve başarı oranları gözlemlenmiş, daha sonra ise 10 kez çapraz doğrulama yöntemi ile test edilmiştir. Test edilen verilen hem kendi aralarında kıyaslanmış hem de literatürde kullanılan benzer algoritmaların sonuçları ile de kıyaslanmıştır.

Çizelge 3.2 k-ort ÇKF Algoritmasının Literatür Kıyaslaması

10 kez çapraz doğrulama sonuçları

	k-ort ÇKF		Algoritma 3		ÇKF	
Veri Kümeleri	Eğitim Kümesi (%)	Test Kümesi (%)	Eğitim Kümesi (%)	Test Kümesi (%)	Eğitim Kümesi (%)	Test Kümesi (%)
WBCD	97,67	95,71	98,21	98,55	100,00	100,00
Fertility	98,13	83,00	90,22	80,23	100,00	90,45
Ionosphere	97,07	74,58	98,10	94,28	100,00	95,76
Liver	58,62	51,08	-	-	100,00	68,40

Çizelge 3.3 k-medoid ÇKF Algoritmasının Literatür Kıyaslaması

10 kez çapraz doğrulama sonuçları

	k medoid ÇKF		Algoritma 3		ÇKF	
Veri Kümeleri	Eğitim Kümesi (%)	Test Kümesi (%)	Eğitim Kümesi (%)	Test Kümesi (%)	Eğitim Kümesi (%)	Test Kümesi (%)
WBCD	97,64	96,00	98,21	98,55	100,00	100,00
Fertility	97,68	86,00	90,22	80,23	100,00	90,45
Ionosphere	96,58	74,60	98,10	94,28	100,00	95,76
Liver	59,91	56,14	-	-	100,00	68,40

Çizelge 3.4 Geliştirilen Toleranslı ÇKF Algoritmasının Literatür Kıyaslaması

10 kez çapraz doğrulama sonuçları

	Toleranslı bütünleşik		Algoritma 3		ÇKF	
Veri Kümeleri	Eğitim Kümesi (%)	Test Kümesi (%)	Eğitim Kümesi (%)	Test Kümesi (%)	Eğitim Kümesi (%)	Test Kümesi (%)
WBCD	97,67	95,71	98,21	98,55	100,00	100,00
Fertility	95,16	85,00	90,22	80,23	100,00	90,45
Ionosphere	94,87	84,44	98,10	94,28	100,00	95,76
Liver	64,07	59,09	-	-	100,00	68,40

Eldeki tüm veri kümeleri geliştirilen k-ort ÇKF algoritmasıyla, k medoid ÇKF algoritmasıyla ve toleranslı bütünleşik algoritma ile 10 kez çapraz doğrulama yöntemine göre çalıştırılmıştır. Sonuçlar Çizelge 3.2, Çizelge 3.3, Çizelge 3.4 te ayrıntılı olarak verilmiştir.

k-ort ÇKF algoritmasını Satı'nın çalışması [10] Algoritma 3 ile kıyaslandığında elde edilen sonuçlar fertility veri kümesinde hem daha iyi bir eğitim kümesi başarısı, hem de daha iyi bir test başarısı elde edilmiştir. Kasımbeyli ve Öztürk'ün çalışması [1] ÇKF algoritması ile kıyaslandığında ise eğitim başarısı, test başarısı düşmüştür.

k-medoid algoritmasının başarı oranların kıyasladığımızda Satı'nın çalışması olan [16] Algoritma 3 ile kıyasladığımızda elde edilen sonuçlar fertility veri kümesinde hem daha iyi bir eğitim kümesi başarısı, hem de daha iyi bir test başarısı elde edilmiştir. Kasımbeyli ve Öztürk'ün çalışması olan [1] ÇKF algoritması ile kıyaslandığında ise eğitim başarısı, test başarısı da düşmüştür.

3.6 Sonuçlar

Toleranslı bütünleşik algoritması literatürde bulunan sonuçlarla kıyaslandığında ise fertility veri kümesinde daha iyi sonuçlara ulaşılmıştır.

Çizelge 3.2, Çizelge 3.3, Çizelge 3.4' teki sonuçlara göre geliştirilen tüm algoritmaları birbiriyle kıyasladığımızda, genel olarak k-ort ÇKF ve k-medoid ÇKF algoritmalarının birbirine yakın ama her seferinde k-medoid ÇKF algoritmasının eğitim başarı oranlarının birbirine daha yakınsadığını gözlemlemekteyiz. k-medoid ÇKF ve k-ort ÇKF algoritmalarının temeldeki en büyük farkı bir veri setindeki gürültü (gürültü) noktalardan daha az etkilenerek, eğitim ve test kümesi arasındaki tahmin kabiliyetini arttırması belirtilebilir. Sonuçlardan da gözlemleyebiliyoruz ki, k-medoid ÇKF algoritması aynı veri kümesinde k-ort ÇKF algoritmasına göre test başarısı tahmin etmede daha başarılıdır. Gürültü noktalardan daha az oranda etkilendiği için gereksiz konik fonksiyonlar çizerek test başarısını daha az yanıltmaktadır. Aynı zamanda literatürdeki veri yapılarını genel olarak inceleyip, bir veri kümesinde de literatürden daha iyi sonuç bulmak, k-medoid ÇKF algoritmasının gürültü noktası daha fazla olan gerçek hayat problemlerinde k-ort ÇKF algoritmasına göre daha başarılı olabileceği ihtimalini arttırmıştır. Bu sebeple sentetik veri kümeleri hazırlanarak ortaya atılan bu düşünce kanıtlanmaya çalışılmıştır.

Toleranslı bütünleşik algoritmanın sonuçlarını değerlendirilecek olunursa, bu algoritmanın da amacı tıpkı k-medoid ÇKF algoritmada olduğu gibi gürültü noktaların vermiş olduğu test kümesi ve eğitim kümesinin birbirinden farkının azaltılmasına yardımcı bir mantığa sahip olmasıdır. k-ort ÇKF algoritmasına göre her veri tipinde daha iyi sonuçların bu algoritmayla elde edildiği gözlenmiştir. Literatürdeki bu verilerden hareketle k-medoid ÇKF algoritması gibi toleranslı bütünleşik algoritmanın da aynı şekilde gürültü noktalarının yoğunlukta olduğu yani güncel hayat problemlerinde k-ort ÇKF algoritmasına göre daha başarılı olduğunu yapay verilerle kanıtlamak için bu algoritmanın da içinde olduğu çalışmalar yapılmış ve sonuçları tez kapsamına eklenmiştir.

4 SONUÇLAR VE ÖNERİLER

ÇKF algoritmasında geliştirilecek yön olan işlem süresinin çok fazla olması kümeleme algoritmalarının etkinliğiyle giderilemeye çalışılmıştır. Aynı zamanda literatürde ÇKF algoritmasını geliştirmek için yapılan algoritmalarla kıyaslandığında bir veri kümesinde literatürden daha iyi sonuçlar elde edilmiştir. Bu veri kümeleri incelendiğinde bazılarında gürültü noktalarının fazlalığı algoritmaların genel anlamda handikapı olduğu gözlemlenmiş ve bu yönde geliştirilen k-medoid ÇKF ve toleranslı bütünleşik algoritmalar da gürültü noktalı verilerde literatüre göre daha iyi sonuçlar çıkardığı ve birbirleri arasında gürültü noktalardan daha az etkilenen k-medoid ÇKF algoritmasının daha başarılı olduğu kanıtlanmıştır. Aynı zamanda geliştirilen toleranslı bütünleşik algoritma ile yine eğitim ve test kümesi arasında tahmin gücünü arttırabildiği ve iki başarı oranı arasındaki farkı azalttığı gözlemlenmiştir.

Eğitim başarısının yüksek olduğu durumlar aslında algoritmanın genel olarak gürültü noktalara takıldığı ve bu noktaları kapsamak adına test kümesinde olmaması gereken noktaları ayırdığı sonucuna varılmıştır. Bu gürültü noktaların önceden çıkarılarak eğitim kümesinin sadece gürültü olmayan noktalar üzerinde bir işlem görmesi eğitim başarısını düşüp, test başarısını da yükseltebileceği kanısı oluşmuştur. Bu sebeple Astorino ve Gaudiosso [8] yaptığı çalışmalarda kullandığı tolerans değeri prensibinden esinlenerek, yeni bir genişletilmiş ÇKF algoritması oluşturulmuştur.

Bu tez kapsamında temel alınan amaç ÇKF algoritmasındaki test başarısının eğitim kümesindeki başarı ile uzaklığının nedenlerini bulmak, bu nedenlere çözüm aramak olmuştur. Bu nedenler araştırılırken, yapılan her bir veri kümesindeki deneyde eğitim kümesindeki tüm noktaları ayırmaya çalışan açgözlü yaklaşım yerine, noktaları ayırdıkça, dışarıda kalan noktaların bir kontrolünü yaparak dışarıda algoritmayı çalıştırmaya değecek kadar nokta olup olmadığını kontrol edilmiştir. Bu ÇKF algoritmasına yeni bir bakış açısı getirmiş aynı zamanda başarılı sonuçlar vermiştir. ÇKF algoritmasındaki test başarısının eğitim kümesindeki başarı ile uzaklığının nedenlerinden bir başkası ise, kullanılan modeldeki sıkı kısıtlardır. Yapay değişkenli model kullanılarak, algoritmanın başarı farkı

ayarlanmaya çalışılmış ve başarı oranları arasındaki fark düşürülmüştür. Fakat veri kümelerinin boyutları ve veri sayıları arttıkça modelin performansında düşüş yaşanmış ve bazı veri kümelerine çözüm bulamamıştır. Bu sebeple her veride kullanışlı olmayabilmektedir. Fakat farklı model ve farklı kümeleme yöntemleri kullanılarak ÇKF algoritması geliştirilmiş ve farklı bir bakış açısı kazandırılmıştır.

İleriki çalışmalarda üzerinde durulacak noktalar, yapay değişkenli modelin performansını arttırabilecek yönlerle ilgili çalışmak ve farklı yöntemlerle etkin tepe noktası seçimi üzerine olacaktır.

KAYNAKÇA

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

[2] Öztürk G., *Sınıflandırma problemleri için yeni bir matematiksel programlama yaklaşımı*, Doktora Tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2007.

[3] Bennett K.P. ve Mangasarian O.L., *Robust linear programming discrimination of two linearly inseparable sets*, Optimization Methods and Software, pp. 1, 23-34,, 1992.

[4] Üney F. ve Türkay M., *A mixed-integer programming approach to multiclass data classification problem,* European Journal Of Operational Research, 173(3), pp. 910-920, 2006.

[5] Rastogi R. ve Shim K., *A decision tree classifier that integrates*, Data Mining and Kowledge Discovery,4, pp. 315-344, 2000.

[6] Han J. ve Kamber M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, 2001.

[7] Gehrke J., Ramakrishnan R., ve Ganti V., *RainForest a framework for fast decision tree,* Data Mining and Knowledge Discovery, pp. 127-162, 2000.

[8] Astorino A. ve Gaudioso M., *Polyhedral separability thorough Succesive LP*, Journal of Optimization Theory and Applications, 112(2), pp. 265-293, 2002.

[9] Cortes C. ve Vapnik V., *Support vector networks*, Machine Learning, 20, pp. 173-297, 1995.

[10] Çiftçi M. T., *Büyük boyutlu sınıflandırma problemleri için matematiksel programlama yaklaşımları*, Yüksek Lisans Tezi, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2011.

[11] Çimen E, *Çok yüzlü konik fonksiyonlar temelli sınıflandırma yaklaşımları ile hareket tanıma*, Yüksek Lisans Tezi, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2013.

[12] Satı N. U., *A Binary Classification Algorithm Based on Polyhedral Conic Functions,* Düzce University Journal of Science and Technology, 3, pp. 152-161, 2015.

[13] MacQueen J. B., *Some methods for classification and analysis of multivariate observations*, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press, pp. 281-297, 1967.

[14] Bagirov A.M., *Modified global k-means algorithm for minimum sum-of-squares clustering problems,* Pattern Recognition, 41(10), pp. 3192-3199, 2008.

[15] Anonim, *K-Means Clusteing*, 2013, https://en.wikipedia.org/wiki/K-means_clustering

[16] Anonim, *K-Medoids*, 2013, https://en.wikipedia.org/wiki/K-medoids

[17] Anonim, *UCI machine learning repository*, 2015, http://archive.ics.uci.edu/ml/

[18] Mangasarian O. L., *Mathematical programming in data mining,* Data Mining and Knowledge Discovery, vol. 1, pp. 183-201, 1997.

[19] Öztürk G., Bagirov A. M. ve Kasimbeyli R., *An incremental piecewise linear classifier based on polyhedral conic seperation*, Machine Learning (to appear), 2014.

[20] Sarıman G., *Veri Madenciliğinde Kümeleme Teknikleri Üzerine Bir Çalışma: K-Means ve K-Medoids Kümeleme Algoritmalarının Karşılaştırılması*, Süleyman Demirel Üniversitesi FBE Dergisi, 15-3, pp. 192-202, 2011.

[21] Alpaydin E., *Introduction to Machine Learning*, Massachusetts: MIT Press, 2005.

[22] Astorino A., Fuduli A.ve Gaudioso M., *Margin maximization in spherical separation*, Computational Optimization and Applications, v.53, pp. 301-322, 2012.

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

[24] Bagirov A. M., Kasimbeyli R., Öztürk G. ve Ugon J, *Piecewise Linear Classifiers Based on Nonsmooth Optimization Approaches*, Optimization in Science and Engineering, Ed. Themistocles M. Rassias, Christodoulos A. Floudas, Sergiy Butenko, ISBN:978-1-4939-0807-3 (Print) 978-1-49-0808-0 (Online), In Honor of the 60th Birthday of Panos M. Pardalos, Springer, pp 1-32, 2014.

[25] Çiftçi M. T., Öztürk G., *Clustering Based Polyhedral Conic Functions Algorıthm in Classıficaıon*, Journal of Industrial and Management Optimization, Volume 11, Number 3, July 2015