T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENME PROBLEMLERİNDE KONVEKS OLMAYAN OPTİMİZASYON MODELLERİNİN İKİ KONVEKS FONKSİYONUN FARKI VE İKİNCİ DERECE KONİK PROGRAMLAMA İLE MODELLENMESİ

Duygu ÜÇÜNCÜ

DOKTORA TEZİ
Matematik Anabilim Dalı
Matematik Programı

Danışman
Prof. Dr. Erdal GÜL

Eş-Danışman
Prof. Dr. Süreyya AKYÜZ

Mart, 2024

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENME PROBLEMLERİNDE KONVEKS OLMAYAN OPTİMİZASYON MODELLERİNİN İKİ KONVEKS FONKSİYONUN FARKI VE İKİNCİ DERECE KONİK PROGRAMLAMA İLE MODELLENMESİ

Duygu ÜÇÜNCÜ tarafından hazırlanan tez çalışması 01.03.2024 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı Matematik Programı **DOKTORA TEZİ** olarak kabul edilmiştir.

Prof. Dr. Erdal GÜL
Yıldız Teknik Üniversitesi
Danışman

Prof. Dr. Süreyya AKYÜZ
Bahçeşehir Üniversitesi
Eş-Danışman

Jüri Üyeleri

Prof. Dr. Erdal GÜL, Danışman
Yıldız Teknik Üniversitesi _____

Prof. Dr. Canan ÇELİK KARAASLANLI, Üye
Yıldız Teknik Üniversitesi _____

Prof. Dr. Atabey KAYGUN, Üye
İstanbul Teknik Üniversitesi _____

Prof. Dr. Ömer GÖK, Üye
Yıldız Teknik Üniversitesi _____

Prof. Dr. Birsen EYGİ ERDOĞAN, Üye
Marmara Üniversitesi _____

Danışmanım Prof. Dr. Erdal GÜL sorumluluğunda tarafımca hazırlanan "Makine Öğrenme Problemlerinde Konveks Olmayan Optimizasyon Modellerinin İki Konveks Fonksiyonun Farkı ve İkinci Derece Konik Programlama ile Modellenmesi" başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Duygu ÜÇÜNCÜ

İmza

“Bilim deyince, onda hakikat diye öne sürdüğü önermelerin pekin olmasını ister; pekinlik ise en mükemmel şekliyle matematikte bulunur. O halde bilim o disiplindir ki; önermeleri matematikle ifade edilir. O zaman matematiği kullanmayan disiplinler bilimin dışında kalacaklardır."

Mustafa Kemal ATATÜRK

Canım Babamın anısına...

TEŞEKKÜR

Üzerimde büyük emekleri olan ve her adımımda yanımda olmuş olan annem Tülay ÜÇÜNCÜ ve babam Ünal ÜÇÜNCÜ'ye teşekkür ederek başlamak isterim. Bana olan güvenleri süreç boyunca moralimi ve motivasyonumu yüksek tutmuştur. Elbette, her zaman yanımda olmak için gayret gösteren kardeşim Deniz ÜÇÜNCÜ'ye de desteği için teşekkür etmek isterim.

Doktora tez danışmanlarım Prof. Dr. Erdal GÜL ve Prof. Dr. Süreyya AKYÜZ'e yönlendirmeleri, destekleri ve yardımları için teşekkürü bir borç bilirim. Çalışma boyunca verdikleri rehberliğin paha biçilmez olduğunu söylemeden geçmem mümkün olamaz. Uzun zamandan beri beraber çalıştığım Profesör GÜL hocama, zor zamanlarımda gösterdiği anlayış ve devamlı desteği için teşekkür ederim. Profesör AKYÜZ hocama, harika bir danışman ve akıl hocası olduğu ve akademik araştırma alanında bana kattıkları için ayrıca derin bir minnettarlık duyduğumu belirtmek isterim. Bu aşamaya gelmemde emeği geçen tüm akademisyenlere de teşekkür ederim. Tez izleme komitesi üyeleri Prof. Dr. Canan ÇELİK KARAASLANLI ve Prof. Dr. Atabey KAYGUN hocalarıma cömertçe bilgi ve uzmanlık paylaşımları nedeniyle ayrıca teşekkür etmek isterim.

Bu fırsatı aynı zamanda bu süre zarfında bana hep destek olan ve moral veren arkadaşlarıma teşekkür etmek için kullanmak isterim. Doktora yıllarım, kişisel hayatımda bazı talihsiz olaylarla çakıştı. En önemlisi babamın tarifsiz ve yeri doldurulamayacak kaybıydı. Arkadaşlarımın sevgisi ve desteği olmadan asla iyileşemezdim.

Yıllar içindeki öğretmenlerimin çaba ve desteğini de es geçmek istemem, bu çabalar akademik kimliğimi ve ilgi alanlarımı şekillendirmede kritik bir rol oynadı. Mesleği sınıf öğretmenliği olan ve ilk okul yıllarımda uygulamalı anlatımlarıyla temelimi kuran babama ve tüm öğretmenlerime teşekkür ederim.

Son olarak, çalışma arkadaşlarım Dr. Pınar KARADAYI ATAŞ ve özellikle Dr. Erkut ARICAN'a tezin bilgisayar ayağındaki destek ve yardımlarından ötürü teşekkür ederim.

Duygu ÜÇÜNCÜ

İÇİNDEKİLER

SİMGE LİSTESİ

l_0	0-norm
l_1	1-norm
E	Bir topluluk
C	Bir kümeleme çözümü
I	Birim matris
CM	Birlikte ilişkilendirme matrisi
δ	Eşik değeri
$\nabla f(x)$	f fonksiyonunun gradyanı
$\nabla^2 f(x)$	f fonksiyonunun Hessian matrisi
$\nabla f(x)^T$	f fonksiyonunun x noktasındaki Jocobian'ı
C_i	$i-inci$ kümeleme çözümünü
$w_{i,j}$	i ve j arasındaki köşe noktası ağırlıkları
C^2	İki kere sürekli türevlenebilen
$\mathcal{K}_1$	İkinci dereceden koni
e	Kolon vektörü
$\mathcal{N}(x^*)$	Komşuluk
N_v	Köşe Noktaları
X, Y	Kümeler(Clusters)
$\mathcal{L}(x,u,v)$	Lagrange fonksiyonu
$g(u,v)$	Lagrange dual fonksiyonu
$\mathcal{K}_m$	Lorentz konisi
D,H,G	Matris
$\succeq$	Matris Eşitsizliği

$p(x), p(y)$	Marjinal dağılım fonksiyonları
$\sim$	Normalleştirilmiş matris
$SNMI$	Normalleştirilmiş karşılıklı bilgi toplamı
$\|.\|_2$	Öklid Normu
λ_j	Özdeğer
q_j	Özvektör
α,θ	Parametre
$\mathbb{R}$	Reel Sayılar
S_+^n	Simetrik pozitif yarı tanımlı
C^1	Sürekli türevlenebilen
k	Topluluk boyutu
N	Topluluktaki kümeleme çözümleri sayısı
L	Topluluk kütüphanesi
$d(i,j)$	Uzaklık fonksiyonu
x,y,v	Vektör
$\boldsymbol{z}$, $\boldsymbol{w}$	Vektör
MM	Yeni tanımlanmış dahili geçerlilik indeksi
$I(X,Y)$	X ve Y kümeleri arasındaki karşılıklı bilgi fonksiyonu
$H(x), H(y)$	X ve Y kümelemeleri arasındaki karşılıklı bilgi fonksiyonları
$NMI(X,Y)$	X ve Y kümelemeleri arasındaki normalleştirilmiş karşılıklı bilgi
$\|x\|_0$	x'in 0 normu
$\|x\|_1$	x'in 1 normu

KISALTMA LİSTESİ

DVM	Destek Vektör Makineleri
DKKP	Disiplinli Konveks-Konkav Programlama
EAMKSO	En Az Mutlak Küçültme ve Seçim Operatörü
HGBA	Hiper Grafik Bölümleme Algoritması
İNM	İç Nokta Metodu
İDKP	İkinci dereceden Konik Programlama
KBY	Kanıt Biriktirme Yöntemi
KKT	Karush-Kuhn-Tucker
KF	Konveks Fark
KFA	Konveks Fark Algoritması
KFP	Konveks Fark Programlama
KP	Kuadratik Programlama
LP	Lineer Programlama
NKB	Normalleştirilmiş Karşılıklı Bilgi
NKBT	Normalleştirilmiş Karşılıklı Bilgi Toplamı
ö.k.	Öyle ki
TAF	Toplanmış Amaç Fonksiyonu
YTP	Yarı-Tanımlı Programlama

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

Makine Öğrenme Problemlerinde Konveks Olmayan Optimizasyon Modellerinin İki Konveks Fonksiyonun Farkı ve İkinci Derece Konik Programlama ile Modellenmesi

Duygu ÜÇÜNCÜ

Matematik Anabilim Dalı
Doktora Tezi

Danışman: Prof. Dr. Erdal GÜL
Eş-Danışman: Prof. Dr. Süreyya AKYÜZ

Makine öğrenmesi algoritmalarının tüm disiplinlerde kullanılmaya başlanması ve yapay zekanın yaygınlaşması ile birlikte literatürde bu yöntemlerin performanslarını arttırıcı yeni matematiksel modellemelere ihtiyaç duyulmuş ve optimizasyon teorisinin kullanıldığı modellemeler popülerliğini gittikçe arttırmıştır. Optimizasyon teorisinde de bilindiği gibi eğer problem konveks değil ise global çözüme ulaşmak her zaman mümkün olmamaktadır. Makine öğrenimi algoritmalarından sınıflandırma ya da kümeleme yapacak optimizasyon modeline ait amaç fonksiyonunun konveks olmadığı durumlarda modellenerek iki konveks fonksiyonun farkı ya da ikinci dereceden konik programlama algoritmaları ile global çözüme yakınsamalar yapılmaktadır.

Topluluk öğrenmesi, birden fazla öğrenme modelini bir araya getirerek bireysel modellerin performansını artırma amacına dayanmaktadır. Bu yaklaşım, farklı modellerin öğrenme ve tahmin güçlerini birleştirerek, bireysel modellerin karşılaşabileceği zayıflıkları ve sınırlamaları aşmayı amaçlamaktadır. Özellikle karmaşık veri yapıları ve zorlu tahmin problemleri söz konusu olduğunda, topluluk öğrenmesi tek bir modelin performansını aşan sonuçlar üretebilmektedir. Bu nedenle, topluluk öğrenmesi, yapay zeka ve makine öğrenimi alanlarında giderek artan bir ilgiyle karşılanmakta ve pek çok disiplinde başarıyla uygulanmaktadır. Topluluk öğrenmesinin bu önemi, bu tezde ele alınan ve optimizasyon teorisine dayanan

kümeleme problemlerinde de kendini göstermektedir. Kümeleme algoritmalarının performansını ve doğruluğunu artırmada topluluk öğrenmesi yöntemlerinin etkinliği, bu çalışmanın merkezinde yer almaktadır.

Denetimsiz öğrenme algoritmalarından biri olan kümeleme, etiketlenmemiş benzer nesneleri aynı gruba yerleştirmek için kullanılır. Gruptaki en iyi modellerin seçimi, topluluk kitaplığındaki gereksiz çözümlerin genel tahmin doğruluğunu azaltacağı gerçeğinden dolayı topluluk öğrenme algoritmalarının genel performansında çok önemli bir role sahiptir. Bu nedenle, bu tür adayların elenmesi hesabına, topluluğun en iyi alt kümesini seçerken doğruluk ve çeşitlilik dikkate alınmalıdır.

Bu tezde topluluk öğrenimi ile denetimsiz öğrenme problemlerinden kümeleme problemi optimizasyon teorisi kullanılarak modellenmiştir. Başka bir deyişle, topluluk içinde optimum doğruluk ve çeşitlilik sağlayan en iyi modellerin seçildiği bir alt kümenin bulunması problemi bir optimizasyon modeli tarafından tanımlanmıştır. Topluluğu oluşturacak kümeleme modelleri literatürde de sıkça tercih edilen k-ortalama algoritması ile kurulmuştur. Bu tezde geliştirilen yeni topluluk küme seçimi modelleri ve önerilen algoritmaların performansları, farklı alanlardan gerçek veri kümeleri üzerinde literatürdeki ilgili kümeleme algoritmaları ile karşılaştırılmıştır. Önerilen tezde kullanılacak olan yöntemler konveks fark programlama ve ikinci dereceden konik programlama algoritmalarıdır. Geliştirilmiş topluluk seçiminin, literatürdeki mevcut topluluk seçim algoritmalarından çoğunlukla daha iyi performans gösterdiği elde edilmiştir.

Anahtar Kelimeler: Optimizasyon, makine öğrenmesi, topluluk budama, ikinci derecede konik programlama, iki konveks fonksiyonun farkı

ABSTRACT

Difference of Convex Functions Programming and Second-Order Conic Programming Modelling of Non-Convex Optimization Problems in Machine Learning

Duygu ÜÇÜNCÜ

Department of Mathematics
Doctor of Philosophy Thesis

Supervisor: Prof. Dr. Erdal GÜL
Co-supervisor: Prof. Dr. Süreyya AKYÜZ

With the increasing use of machine learning algorithms in all disciplines and the widespread adoption of artificial intelligence, new mathematical modeling techniques have been needed to enhance the performance of these methods. As a result, modeling techniques that utilize optimization theory have become increasingly popular. As is known in optimization theory, if a problem is not convex, reaching a global solution may not always be possible. In cases where the objective function of an optimization model that performs classification or clustering through machine learning algorithms is not convex, global convergence can be achieved through algorithms by modeling the objective function as the difference of two convex functions or as the second-order cone programming.

Ensemble learning, which combines multiple learning models, aims to enhance the performance of individual models. This approach merges the learning and predictive strengths of different models, aiming to overcome weaknesses and limitations that individual models may encounter. Especially in complex data structures and challenging prediction problems, ensemble learning can produce results surpassing the performance of a single model. This growing interest in ensemble learning in the fields of artificial intelligence and machine learning is also evident in clustering problems based on optimization theory, which is the focus of this thesis. The effectiveness of ensemble learning methods in improving the performance and

accuracy of clustering algorithms is central to this work.

Clustering, one of the unsupervised learning algorithms, is used to group similar unlabeled objects into the same group. The selection of the best models in the ensemble has a crucial role in the overall performance of ensemble learning algorithms due to the fact that redundant solutions in the ensemble library will reduce the overall prediction accuracy. Therefore, accuracy and diversity must be considered when selecting the best subset of the ensemble to account for the elimination of such candidates.

In this thesis, the clustering problem, one of the ensemble learning and unsupervised learning problems, is modeled using optimization theory. In other words, a collective decision was determined by an optimization model, by an aggregated clustering solution that provides optimum accuracy and diversity in the ensemble. The k-means algorithm is used as the base learner due to its popularity in the clustering literature. The new ensemble cluster selection models developed in this thesis and the performance of the proposed algorithm are compared with different clustering algorithms on real datasets from different fields. The proposed methods used in the thesis are difference of convex programming and second-order conic programming algorithms. Improved ensemble selection has often been shown to outperform existing ensemble selection algorithms in the literature.

Keywords: Optimization, machine learning, ensemble pruning, second order conic programming, difference of convex functions

1
GİRİŞ

Son yıllarda, makine öğrenimi ile optimizasyonun kesişimi, akademik araştırmalarda önemli ölçüde ilgi görmüştür. Bu artan ilgi, verilerin artan karmaşıklığı ve hacmi ile daha etkili algoritmaların analiz ve tahmin yapma ihtiyacından kaynaklanmaktadır. Bu alandaki öncü çalışmalardan biri, Bottou ve arkadaşları [1] tarafından sunulmuştur. Bu çalışmada, makine öğreniminde kullanılan optimizasyon tekniklerine kapsamlı bir bakış sunulmakta ve özellikle büyük ölçekli veri ortamlarında makine öğrenimi algoritmalarının başarısının, optimizasyon yöntemlerinin etkinliğine büyük ölçüde bağlı olduğu savunulmaktadır. Bottou ve arkadaşları, büyük veri kümelerinin işlenmesinde stokastik gradyan inişinin ve türevlerinin önemini vurgulamaktadırlar.

Son yarım yüzyılda, bilgi işlem ve dijital teknolojiler, araçlar ve hizmetlerin dijital versiyonlarına geçişle büyük dönüşümlere öncülük etmiştir. Bilgisayarların kullanımının yaygınlaşmasıyla birlikte, uygulama alanları da genişlemiştir. Bilgisayarların gücü, her tür bilginin dijital olarak temsil edilebilmesi ve işlenebilmesi gerçeğinden kaynaklanmaktadır. Özellikle, veritabanları sayesinde, bilgisayarlar sadece işleme gücü değil, aynı zamanda geniş bilgi havuzları olarak da işlev görmektedir [2].

Bilim ve teknolojideki ilerlemeler, büyük veri setlerinin ortaya çıkmasına ve bu verilerin tek başına anlamlı olmamasından dolayı veri madenciliği ve akıllı sistemlere olan ihtiyacın artmasına sebep olmuştur. Makine öğrenimi, bilgisayarların algılayıcı verisi ya da veritabanları gibi veri türlerine dayalı öğrenimini olanaklı kılan algoritmaların tasarım ve geliştirme süreçlerini konu edinen bir bilim dalıdır. Kümeleme (clustering), sınıflandırma (classification) ve birliktelik kuralları (association rules) bu alanda iyi bilinen tekniklerdir. Makine öğreniminin temel fikri, önceki gözlemleri kullanarak gelecekteki eylemleri tahmin etmektir. Yeterli miktarda veri ve parametre varlığında, makine öğrenimi, insanların manuel olarak göremediği veya belirleyemediği kısıtlamaları, kuralları ve yapıları belirleyebilmektedir. Makine öğrenimi alanındaki temel tekniklerden ikisi denetimli ve denetimsiz öğrenmedir. Denetimli öğrenme, veri kümesi üzerinde, yani x

bağımsız değişken (girdi) ve bunlara karşılık gelen $y \in \mathbb{Z}$ bağımlı değişken (çıktı) değerlerinin bilindiği bir makine öğrenimi türüdür. Başka bir deyişle, model, istenen çıktıyı zaten bildiği örneklerden öğrenmektedir. Denetimli öğrenmenin amacı, yeni, görünmeyen x verileri için y değerlerini doğru bir şekilde tahmin edebilen bir model oluşturmaktır. Denetimli öğrenmenin en yaygın örnekleri, sınıflandırma ve regresyon problemleridir. Sınıflandırma problemlerinde bağımlı değişkenler tamsayı değerleri alırken regresyon problemlerinde ise sürekli (reel) değerler almaktadır. Denetimli öğrenme, etiketlenmiş veri kümesi $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ üzerinde çalışır. Burada, x_i girdi vektörlerini ve y_i karşılık gelen çıktı etiketlerini temsil eder. Model, $f(x) = y$ formundaki bir fonksiyonu tahmin etmeyi amaçlar. Buradaki hedef, tahmin edilen $\hat{y}$ ve gerçek y arasındaki farkı, genellikle bir kayıp fonksiyonunu $L(y, \hat{y})$ minimize ederek en aza indirir. Bu süreç, modelin doğruluğunu maksimize ederken, genellikle çapraz doğrulama gibi yöntemlerle doğrulanır. Denetimsiz öğrenme ise, etiketlenmemiş veri kümesini $\{x_1, x_2, \ldots, x_n\}$ veri noktalarının birbirlerine olan benzerlikleri üzerinden faydalanarak gruplamayı amaçlar. Kümeleme bu gruplama yöntemlerinden birisidir ve veri noktalarının birbirlerine olan yakınlıklarını çeşitli uzaklık metriklerini kullanarak k gruba ayırır. Kümeleme algoritmasında ki hedef fonksiyon, kümeler içi varyansı minimize ederken kümeler arası uzaklığı maksimize eder.

Topluluk öğrenmesi, modern makine öğreniminin en önemli unsurlarından biri haline gelmiştir. Birden fazla makine öğrenimi modelinin birleştirilmesiyle oluşan bu yaklaşım, bireysel modellerin güçlü yönlerini bir araya getirerek genel performansı arttırmaktadır [3]. Bu yöntem, özellikle tahmin ve sınıflandırma problemlerinde, modelin kararlılığını ve doğruluğunu önemli ölçüde iyileştirebilir. Ayrıca, topluluk öğrenme, çeşitli veri setleri ve senaryolar üzerinde daha iyi genelleme yapabilme yeteneğiyle de dikkat çekmektedir.

Topluluk yöntemlerinin etkinliği, büyük ölçüde topluluk seti içindeki öğrenme modellerinin çeşitliliğine ve doğruluğuna bağlıdır. Topluluklar, doğruluk ve çeşitlilik ikilemini tatmin etmek için farklı öğrenme algoritmaları eklemek, temel öğrenici parametrelerini değiştirmek veya veri setinin özelliklerini değiştirmek gibi farklı yollarla oluşturulabilir. Topluluktaki en iyi aday çözümlerin seçimi, topluluk kitaplığındaki gereksiz çözümlerin genel tahmin doğruluğunu azaltacağı gerçeğinden dolayı topluluk öğrenme algoritmalarının genel performansında çok önemli bir role sahiptir. Kümeleme çözümlerinin doğruluğu ve çeşitliliği arasında, biri artarken diğeri azaldığından, bir denge vardır. Ek olarak, kümeleme topluluğundaki optimal küme sayısı, kümeleme topluluğunun kararını etkilemektedir. En iyi topluluk kümeleme adaylarını bulmak için topluluktan gereksiz küme çözümlerini ortadan kaldırmak, son yıllarda araştırmalarda öne çıkan bir konu olarak ortaya çıkmıştır [4–9]. Bununla

birlikte, alt kümenin optimal boyutunu belirleme sorunu henüz literatürde yeterince incelenmemiştir. Bu faktörler nedeniyle, bu tezde önerilen optimizasyon modeli özgündür. Böyle ideal bir alt küme boyutunu seçmek ve aynı anda hem doğru hem de çeşitli sınıflandırıcıları seçmek için topluluk alt kümesini bulma problemi, bir optimizasyon çerçevesi ile modellenmiştir.

Optimizasyon, bir dizi kısıtlamaya tabi olarak, olası tüm çözümler arasından bir soruna en iyi çözümü bulma sürecidir. Problem matematiksel olarak, minimize edilecek veya maksimize edilecek bir amaç fonksiyonundan ve uygun çözümleri sınırlayan bir dizi kısıtlamadan oluşan bir optimizasyon modeli olarak temsil edilebilir. Optimizasyon, optimizasyon probleminin değişkenleri üzerinde kısıtlamaların yokluğunda veya varlığında oluşan, kısıtsız optimizasyon ve kısıtlı optimizasyon olmak üzere iki gruba ayrılmaktadır. Kısıtsız optimizasyonda, optimizasyon probleminin değişkenleri üzerinde herhangi bir kısıtlama yoktur. Optimize edilecek amaç fonksiyonu, n'nin değişken sayısı olduğu n boyutlu bir uzay üzerinde tanımlanır. Hedef, amaç fonksiyonunu minimize eden veya maksimize eden değişkenlerin değerlerini bulmaktır. Buna karşılık, kısıtlı optimizasyonda, optimizasyon probleminin değişkenleri üzerinde bir veya daha fazla kısıtlama vardır. Burada hedef, amaç fonksiyonunu optimize ederken kısıtlamaları sağlayan değişkenlerin değerlerini bulmaktır. Kısıtlar, denklemler veya eşitsizlikler olarak temsil edilebilir ve uygun çözümler, kısıtlamalar tarafından tanımlanan n-boyutlu uzayın bir alt kümesi içinde yer alır. Optimizasyon problemleri, amaç fonksiyonunun ve kısıtlamaların doğasına bağlı olarak lineer veya lineer olmayan olarak nitelendirilir. Lineer optimizasyon problemleri, lineer amaç fonksiyonlarını ve lineer kısıtlamaları içerirken, lineer olmayan optimizasyon problemleri, lineer olmayan amaç fonksiyonlarını ve/veya lineer olmayan kısıtlamaları içerir.

Global çözümü bulmak, tüm optimizasyon problemlerinin temel görevidir. Optimizasyon teorisinden bilindiği gibi, amaç fonksiyonu konveks bir fonksiyon ise, optimal çözümün global olduğu garanti edilebilmektedir. Başka bir deyişle, önerilen optimizasyon modeli, kitaplık içinde en uygun koşulları sağlayan kardinalitenin bir alt kümesini seçmektedir. Makine öğrenimi algoritmalarından sınıflandırma ya da kümeleme yapacak optimizasyon modeline ait amaç fonksiyonunun konveks olmadığı durumlarda, global çözüme yakınsamak için izlenebilecek yollardan ikisinde, amaç fonksiyonu, iki konveks fonksiyonun farkı ve ya ikinci dereceden konik programlama şekilde modellenmektir.

Bu tezin amacı, kümeleme topluluğu içerisinden doğruluk ve çeşitlilik ödünleşimini optimize ederek en iyi alt topluluğu belirleyen modelin tanımlanmasıdır. En iyi alt topluluğu belirleyecek model optimizasyon tekniklerinden konveks fark

programlama ve ikinci dereceden konik programlama kullanılarak iki farklı yaklaşım ile tanımlanmıştır. Bu tezdeki yeni topluluk kümeleme algoritmaları, verilerden bağımsız olarak genelleştirilmiş bir model olarak çözülmüş ve literatürdeki ilgili kümeleme algoritmalarıyla başarı oranlarını karşılaştırmak için farklı alanlardan gerçek veri kümelerine uygulanmıştır.

Bu tez, kümeleme problemine, doğruluk ile çeşitliliği dengelerken topluluk içerisinden alt toplulukta kullanılacak optimal sayıda kümeleme çözümünü otomatik olarak belirleyen bir optimizasyon modeli aracılığıyla yaklaşmaktadır. Yeni geliştirilen topluluk kümeleme algoritmaları, çeşitli veri kümelerinde literatürdeki diğer algoritmalarla karşılaştırıldığında sıklıkla daha iyi sonuçlar elde edilmiştir.

1.1 Litaratür İncelemesi

Optimizasyon problemlerinin basit yaklaşık çözümleri, daha düşük hesaplama karmaşıklığı ve daha hızlı sonuçlar elde etmesi nedeniyle, özellikle de kesin çözümlerin hesaplama açısından yoğun olduğu veya pratik olmadığı büyük ölçekli veya zamana duyarlı senaryolarda sıklıkla tercih edilmektedir. Özetle, teorik açıdan kesin çözümler ideal olsa da, gerçek dünya uygulamaları için yaklaşık çözümler genellikle daha pratik ve yeterlidir. Seyreklik, yüksek boyutlu verileri daha küçük bir boyutla temsil etmeyi ve yeniden yapılandırmayı mümkün kılar. Seyrek optimizasyon, çözümdeki bağımsız değişkenlerin ($\mathbf{x} \in \mathbb{R}^n$) çoğunun sıfır olduğu durumları ifade eder ve genellikle l_0 normunu, $||\mathbf{x}||_0$, içerir. Bu norm, çözümdeki sıfır olmayan elemanların sayısını hesaplamaktadır. Matrisler için, seyreklik, matristeki çoğu ögenin sıfır olması anlamına gelir. Makine öğreniminde yıllardır seyrek optimizasyon problemleri kullanılmaktadır ve sinyal işleme [10], görüntü işleme [11], kanser tedavisi için radyoterapi, genelleştirilmiş özdeğer problemleri [12] ve topluluk seçimi gibi birçok çalışmada benimsenmiştir.

Topluluk yöntemleri, birçok makine öğrenimi problemine son teknoloji çözüm olarak kabul edilmektedir. Topluluk kümeleme, oldukça ilgi gören popüler bir denetimsiz öğrenme tekniğidir. Bu teknik, daha iyi kümeleme üretmek için birkaç temel kümeleme algoritmasının sonuçlarını birleştirmeyi amaçlamaktadır. Son yıllarda, bu bakış açısını geliştirmek için çeşitli teknikler kullanılması topluluk kümelemeyi mümkün kılmaktan öte gelişmesini sağlamıştır. Topluluk kümelemenin amacı, tüm bireysel kümeleme algoritmalarından gelen bölümleme verilerini birleştirerek son kümeleri gerçek çözüme yakınsatmaktadır [13–15]. Topluluk kümeleme, bireysel kümeleme algoritmalarının dezavantajlarını azaltabilmekte ve onları daha büyük veri kümeleri için daha uygun hale getirebilmektedir.

Geleneksel topluluk kümeleme yöntemleri, ortak bir karara varmak için tüm kümeleme çözümlerini kullanmayı amaçlamaktadır. Bununla birlikte, son araştırmalar topluluk içindeki çeşitli ve doğru kümeleme çözümlerinin topluluk öğrenme performansını iyileştirdiğini göstermiştir. Bu nedenle, literatürde topluluk seçim prosedürleri geliştirilmiş ve topluluk kitaplığından en doğru ve çeşitli alt kümenin seçilmesinde etkili olmuştur [4, 5, 8, 16–18].

Topluluk seçilirken dikkate alınması gereken doğruluk ve çeşitlilik olmak üzere iki temel faktör bulunmaktadır. Kümeleme çözümlerinin doğruluğu ve çeşitliliği arasında, biri artarken diğeri azalacağından, bir ödünleşim bulunmaktadır. Bunun yanında, kümeleme topluluğundaki optimal kümeleme modeli sayısı, kümeleme topluluğunun kararını etkilemektedir. En iyi topluluk kümeleme adaylarını bulmak için topluluktan gereksiz küme çözümlerini ortadan kaldırmak, son araştırmalarda öne çıkan bir konu olarak ortaya çıkmıştır [4–9].

Topluluk budamasının, doğruluğu korurken alt topluluğun kardinalitesini azalttığı bildirilmiştir. Topluluk budama yöntemleri, grubun karmaşıklığını azaltmak için topluluk seçim yöntemleri olarak da düşünülebilir. Çalışma [19]'de yazarlar, "birçok-her-şeyden-daha-iyi-olabilir" teoremini kanıtlar kanıtlamaz, küçük ama doğru bir topluluk elde etmek mümkün hale gelmiştir. Bu yaklaşımla beraber yalnızca bir model alt kümesi tahmin adımında dahil edilmektedir. Çalışma [20, 21]'deki yöntemler, modellerin yalnızca bir alt kümesini dinamik olarak sorgulayarak orijinal toplulukla aynı çözümü üretmişlerdir. Bu teknikler, en azından orijinal topluluk kadar iyi performans gösteren topluluk üyelerinin bir alt kümesini aramaktadırlar. Alt küme oluşturulduktan sonra, modellerin nihai mutabakatı (konsensüs) için sadece seçilmiş olanlar tutulmaktadır. Ne yazık ki, güçlendirilmiş bir topluluk kadar iyi performans gösteren küçük boyutlu bir topluluk aramanın NP-zor bir problem olduğu gösterilmiştir [22]. Bu yöntemlerin ana fikri, tüm oyların sayılmadığı durumlarda bile çoğunluk oylamasında kazanan sınıfın belirlenebilmesidir. Çalışma [20], uygun şekilde seçilirse, grubun yalnızca bir alt kümesi kullanıldığında bile tahmin performansının maksimum düzeyde elde edilebileceğini göstermiştir. Yıllar geçtikçe, çok sayıda topluluk budama yöntemi geliştirilmiştir. Sıralama ve aramaya dayalı yöntemler, bir topluluk altkümesi seçmek için en popüler iki yaklaşımdır. Sıralama yöntemleri, belirli bir ölçüye göre bireysel üyeleri sıralamakta ve bir eşiğe göre en üst sıradaki temel modelleri seçmektedir [23–26]. Arama tabanlı yöntemler ise, sınıflandırıcının uzayında buluşsal (heuristic) bir arama yürütmekte ve sınıflandırıcının alt kümelerinin ortak performansını değerlendirmektedir [3, 27, 28].

Optimizasyon tabanlı budamada, topluluk budama problemi bir optimizasyon

problemi olarak tanımlanmaktadır ve amaç, doğruluk ve çeşitliliği maksimize eden en iyi alt-topluluğu keşfetmek için optimizasyon problemini çözmektir. Literatürde yarı tanımlı programlama (YTP) topluluk budama algortimalarında kullanılmış yöntemlerden biridir [3]. Çalışma [29]'da, çalışma [3]'deki modele benzer şekilde elemanlarının aday modellerin hataları ile tanımlandığı bir matrisle amaç fonksiyonunun toplam hatayı en aza indirdiği ve çeşitliliği en üst düzeye çıkardığı kısıtlı bir optimizasyon modeli kullanarak bir topluluk seçim modeli oluşturulmuştur. Burada sıfır norm, student t-dağılımının negatif log olasılığı olacak şekilde yaklaştırılmıştır. Modeli çözmek için disiplinli dışbükey-içbükey programlama kullanmışlardır. Çalışma [30] ise, konveks fark algoritması (KFA) kullanarak oluşturulan topluluk budaması modelinde sıfır norm yaklaşımlarından $\|x\|_0 = \sum_{i=1}^{n}(1 - e^{-\alpha|x_i|}), \alpha > 0$ kullanılarak panelize edilmiştir.

Çalışma [3]'deki sayısal deneyler, YTP tabanlı budama algoritmasının literatürdeki diğer buluşsal yöntemlerden daha iyi performans gösterdiğini göstermiştir fakat hesaplama maliyeti oldukça yüksektir [31]. Öte yandan, sayısal hesaplamalar, İkinci Dereceden Koni Programlamayı (İDKP) çözmenin hesaplama maliyetinin, YTP'nin hesaplama maliyetinden çok daha düşük ve lineer Programlamanın (DP) maliyetine yakın olduğunu göstermektedir [32]. Ayrıca Kim ve Kojima [33], İDKP'nin YTP'den çok daha az CPU (Merkezi İşlem Birimi) zamanı kullandığını belirtmektedir.

Tüm bu nedenlerden dolayı bu tezde, tahmin aşaması için topluluğun en iyi adaylarının seçileceği şekilde doğruluk ve çeşitlilik arasındaki dengeyi aynı anda optimize eden kümeleme için iki konveks fonksiyonun farkı ve ikinci dereceden konik optimizasyon modelleri önerilmiştir. Önerilen modeller gerçek dünya verileri kullanılarak test edilmiş ve tahmin doğrulukları literatürdeki topluluk kümeleme yöntemleri ile karşılaştırılmıştır. Topluluk seçimi problemi için önerilen optimizasyona dayalı bu modeller, makine öğrenimindeki topluluk kümesi seçimi literatürüne yeni bir katkıdır. Ayrıca, diğer budama algoritmalarından farklı olarak, optimizasyon modelinin çözülmesiyle alt topluluk otomatik olarak elde edilmiştir.

1.2 Tezin Amacı

Bu tezde kümeleme problemi optimizasyon teorisi kullanılarak topluluk öğrenimi ile modellenmiştir. Toplulukta optimum doğruluk ve çeşitliliği sağlayan alt topluluğu oluşturan kümeleme çözümleri bir optimizasyon modeli tarafından belirlenmiştir. Kümeleme literatüründeki sıkça kullanılan k-means algoritması temel kümeleme yöntemi olarak seçilmiştir. Bu tezde geliştirilen yeni topluluk küme seçimi modelleri ve önerilen algoritmanın performansı, farklı alanlardan gerçek veri kümeleri

üzerinde literatürde ilgili kümeleme algoritmaları ile karşılaştırılmıştır. Önerilen tezde alt topluluğun tanımlanması iki farklı yöntem ile modellenmiştir. Bunlardan birincisi konveks fark programlama ve ikincisi ikinci dereceden konik programlama algoritmalarıdır. Bu tezde optimizasyon modelleri ile tanımlanan topluluk seçimi yaklaşımlarının, literatürdeki mevcut topluluk seçim algoritmalarından sıklıkla daha iyi performans gösterdiği gösterilmiştir. KF ve İDKP kullanarak topluluk budamanın bir avantajı, budama boyutuna ihtiyaç duymadan alt kümeyi otomatik olarak bulmaktır, bu da yöntemi literatürde tartışılan diğer birçok yöntemden ayırmaktadır.

1.3 Tezin Katkısı

Makine öğrenimi ve topluluk öğreniminin budaması ile ilgili literatürdeki birincil eksiklik, optimum hiper parametreyi otomatik olarak bulma zorluğundan ileri gelen sınırlamadır. Topluluk öğrenmedeki diğer bir sorun ise, çoğunlukla tahmin doğruluğu ve çeşitliliğinin birlikte değil tek başına değerlendirilmesidir. Literatürde sadece bazı çalışmalar, bir optimizasyon modeli tarafından, ama otomatik bir seçim kullanmadan, sıralı arama tabanlı yöntemlerle her ikisini de dikkate almaktadır [5]. Bu tezde, konveks fark (KFA) ve ikinci dereceden koni programlama (İDKP) algortimaları ile topluluğun budandığı yeni optimizasyon modelleri önerilmiştir ve engeller aşağıdaki katkılarla aşılmaktadır:

- Doğruluk ve çeşitlilik arasındaki dengenin, alt topluluğu otomatik olarak elde ederken, aynı anda optimize edilmesi,
- Konveks olmayan dönüşüme ihtiyaç duyulmadan İDKP kullanılarak maksimum kesme problemi aracılığıyla küme seçimi.

Bu tezde, Bölüm 1'de literatür taraması, tezin amacı ve katkısı açıklanmıştır. Bölüm 2'de ise matematiksel altyapıyı oluşturacak temel bilgilendirme yapılmıştır. Bölüm 3'de makine öğrenmesi, topluluk öğrenmesi ve modelleri karşılaştırdığımız yöntemlerin açıklamaları yapılmıştır. Bölüm 4, önerilen metotlar ve nümerik deneylere ayrılmıştır. Son olarak, Bölüm 5, sonuç ve önerileri oluşturmaktadır.

2
Matematiksel Altyapı

Bu bölümde matematiksel altyapı için gerekli temel bilgiler verilecektir.

2.1 Optimizasyon Teorisi

Optimizasyonun temel fikri, bir dizi kısıtlamaya tabi olarak ve ya olmayarak amaç fonksiyonunu minimize ya da maksimize eden değişkenlerin bulunmasıdır. Amaç fonksiyonu tipik olarak bir çözümün ne kadar "iyi" olduğunu ölçen matematiksel bir ifadedir, kısıtlamalar ise herhangi bir uygun (feasible) çözüm tarafından karşılanması gereken koşullardır. Yapılması gereken kısıtlamalara tabi amaç fonksiyonunu maksimize eden veya minimize eden çözüm olan optimal çözümü bulmaktır. Optimum çözüm tipik olarak, amaç fonksiyonunun global maksimumu veya minimumu olmak veya bir dizi optimallik koşulunu sağlamak gibi belirli özelliklerle karakterize edilmektedir.

Optimizasyon algoritmaları, optimum çözümü verimli ve etkili bir şekilde aramak için kullanılmaktadır. Bu algoritmalar tipik olarak uygun çözüm kümesini araştıran ve her adımda amaç fonksiyon değerini iyileştirmeye çalışan yinelemeli prosedürleri içermektedir. Basit gradyan iniş yöntemlerinden lineer programlamaya, lineer olmayan programlamadan, dinamik programlamaya ve diğer tekniklere dayalı daha karmaşık yöntemlere kadar değişen birçok farklı optimizasyon algoritması türü vardır.

Bu tezdeki problem, kısıtlı optimizisayon kavramlarına dayandığından bu bölümün devamında kısıtlı optimizasyon hakkında ayrıntılı bilgiler sunulacaktır. Optimizasyon problemlerinin çözüm yollarına geçmeden önce, konuyla ilgili temel tanımlar ve teoremler açıklanacaktır.

x bilinmeyenler veya parametreler olarak da adlandırılan değişkenlerin vektörünü, x'e bağlı f fonksiyonu ise maksimize veya minimize edilmek istenen amaç fonksiyonunu, ve c_i, bilinmeyen x vektörünün sağlaması gereken belirli denklemleri ve eşitsizlikleri tanımlayan kısıt fonksiyonlarını ifade etmek üzere bir optimizasyon problemi aşağıdaki şekilde ifade edilebilir:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{ö.k.} \quad & c_i(x) = 0\,, i \in \epsilon \\ & c_j(x) \geq 0\,, j \in I \end{aligned} \tag{2.1}$$

Burada I ve ϵ, sırasıyla eşitlik ve eşitsizlik kısıtlamaları için indeks kümelerdir.

Tanım 2.1. Eğer $\forall x$ için $f(x^*) \leq f(x)$ ise, o zaman x^* noktasına f fonksiyonunun *global minimum*dur denir. Eğer $\forall x, x \neq x^*$ için $f(x^*) < f(x)$ ise, o zaman x^* noktasına f fonksiyonunun *kesin global minimum* noktası denir.

Tanım 2.2. Bir $\mathcal{N}(x^*)$ komşuluğu

$$\mathcal{N}(x^*) = \{x \in \mathbb{R}^n \quad \text{ö.k.} \quad \|x - x^*\| < \delta\}$$

olacak şekilde x^* merkezli ve δ yarıçaplı bir *açık top* ifade etmektedir [34].

Tanım 2.3. Eğer $\forall x \in \mathcal{N}$ için $f(x^*) \leq f(x)$ olacak şekilde x^*'in bir $\mathcal{N}$ komşuluğu var ise x^* *lokal minimum*dur denir. Eğer $\forall x, x \neq x^*$ için $f(x^*) < f(x)$ ise, o zaman x^* noktasına f fonksiyonunun *kesin lokal minimum* noktası denir.

Tanım 2.4. Bir f fonksiyonun *gradyan*ı $\nabla f(x)$

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

olacak şekilde kısmi türevlerden oluşan bir vektördür.

Tanım 2.5. Bir f fonksiyonun *Hesse-Hessian* matrisi

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix} \tag{2.2}$$

olacak şekilde ikinci kısmi türevlerden oluşan $n \times n$ bir matrisdir.

Teorem 2.1 (Clairaut Teoremi- İkinci Mertebeden Karışık Türevlerin Eşitliği)**.** *Eğer f iki kere sürekli türevlenebilen bir fonksiyon ($f \in C^2$) ise*

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}, i, j = 1, 2, \ldots, n, \quad i \neq j$$

denklemi sağlanır.

Sonuç 2.1. *Hessian matrisi simetrik bir matristir. Diğer bir deyişle,*

$$(\nabla^2 f(x))^T = (\nabla^2 f(x)),$$

dir.

Tanım 2.6. Varsayalım ki f vektör değerli fonksiyonu, her bir f_i reel-değerli bir fonksiyon olmak üzere,

$$f(x) = f(x_1, x_2, \ldots, x_n) = \begin{pmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_m(x_1, x_2, \ldots, x_n) \end{pmatrix}$$

olacak şekilde ifade edilsin. O halde, ∇f matrisinin elemanları $(\nabla f(x))_{i,j} \equiv \frac{\partial f_i(x)}{\partial x_i}$ olacaktır.
f fonksiyonunun x noktasındaki *Jacobian*'ı $\nabla f(x)^T$ şeklinde tanımlanır.

Teorem 2.2 (Taylor Teoremi)**.** *Varsayalım ki $f : \mathbb{R}^n \to \mathbb{R}$ sürekli türevlenebilir (C^1) ve $p \in \mathbb{R}^n$ olsun. O halde, herhangi $p \in (0,1)$ için*

$$f(x+p) = f(x) + \nabla f(x+\xi p)^T p$$

dir. Dahası, eğer f iki kere türevlenebilir bir fonksiyon ise herhangi $\xi \in (0,1)$ için

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+\xi p) p \, d\xi$$

ve

$$f(x+p) = f(x) + \nabla f^T(x) p + \frac{1}{2} p^T \nabla^2 f(x+\xi p) p$$

olacaktır.

Teorem 2.3 (Birinci-dereceden Gerek Koşulları)**.** *Eğer x^* bir lokal minimum ve f, x^*'in bir komşuluğunda sürekli türevlenebilir ise $\nabla f(x^*) = 0$ olur.*

İspat. Varsayalım ki $\nabla f(x^*) \neq 0$ olsun. $p = -\nabla f(x^*)$ olacak şekilde p vektörü tanımlansın ve $p^T \nabla f(x^*) = \|\nabla f(x^*)\|^2$ olduğu göz önüne alınsın. ∇f, x^* yakınında sürekli olduğundan $\forall\ \xi \in [0, T]$ için

$$p^T \nabla f(x+\xi p) < 0$$

olacak şekilde bir $T > 0$ skaleri vardır. Her $\bar{\xi} \in (0, T]$ için Taylor teoreminden ötürü herhangi bir $\xi \in (0, \bar{\xi})$ için

$$f(x^* + \bar{\xi}p) = f(x^*) + \bar{\xi}p^T \nabla f(x^* + \xi p)$$

olmalıdır. Bu yüzden, her $\bar{\xi} \in (0, T]$ için $f(x^* + \bar{\xi}p) < f(x^*)$ olacaktır. Bu durumda, f azalırken x^*'dan uzaklaşan bir yön bulmuş oluruz ki x^* lokal minimum olamaz. Bu ise bir çelişkidir.

■

Tanım 2.7. Eğer $\nabla f(x^*) = 0$ ise x^* bir *durağan nokta*dır denir.

Sonuç 2.2. *Teorem 2.5'e göre her lokal minimum bir durağan nokta olmak zorundadır.*

Tanım 2.8. [35] Bir $n \times n$ boyutlu A simetrik matrisinin *pozitif tanımlı* oması için gerek koşul $x = (x_1, x_2, \ldots, x_n)^T$ olmak üzere, $\forall x \neq 0$ için $x^T Ax > 0$ olmasıdır. Diğer bir deyişle, eğer A matrisinin tüm öz değerleri pozitif ise A matrisi pozitif tanımlıdır.

Tanım 2.9. [35] Bir A simetrik matrisi eğer

- $\forall x$ için $x^T Ax \geq 0$ (diğer bir deyişle, A matrisinin öz değerlerinin tümü negatif değil ise) ise *A pozitif yarı tanımlı*dır,
- $\forall x \neq 0$ için $x^T Ax < 0$ (diğer bir deyişle, A matrisinin öz değerlerinin tümü negatif ise) ise *A negatif tanımlı*dır,
- $\forall x \neq 0$ için $x^T Ax < 0$ (diğer bir deyişle, A matrisinin öz değerlerinin tümü pozitif değil ise) ise *A negatif yarı tanımlı*dır,
- $x^T Ax$ hem pozitif hem negatif değerler alabiliyor (diğer bir deyişle, A matrisinin içinde en az bir negatif ve en az bir pozitif öz değer mevcut) ise *A tanımsız*dır.

Teorem 2.4. *İkinci dereceden türevlenebilen $f : \mathbb{R}^n \to \mathbb{R}$ fonksiyonunun durağan noktası x^* ve $\nabla^2 f(x)$, $f(x)$ fonksiyonunun Hessian matrisi olsun. Bu koşullar altında aşağıdaki ifadeler doğrudur:*

- *$\nabla^2 f(x)$ pozitif yarı tanımlı ise, x^*, $f(x)$ için global minimumdur.*
- *$\nabla^2 f(x)$ pozitif tanımlı ise, x^*, $f(x)$ için kesin global minimumdur.*
- *$\nabla^2 f(x)$ negatif yarı tanımlı ise, x^*, $f(x)$ için global maksimumdur.*
- *$\nabla^2 f(x)$ negatif tanımlı ise, x^*, $f(x)$ için kesin global maksimumdur.*

- $\nabla^2 f(x)$ *tanımsız ise,* x^**,* $f(x)$*'in büküm(eyer) noktasıdır.*

Teorem 2.5 (İkinci-dereceden Gerek Koşulları)**.** *Eğer* f*,* x^* *bir lokal minimumu ve* $\nabla^2 f(x)$*,* x^**'ın açık bir komşuluğunda sürekli ise* $\nabla f(x^*) = 0$ *ve* $\nabla^2 f(x^*)$ *pozitif yarı tanımlı olur.*

Teorem 2.6 (İkinci-dereceden Yeter Koşulları)**.** *Varsayalım ki* $\nabla^2 f(x)$*,* x^**'ın açık bir komşuluğunda sürekli olsun. Eğer*

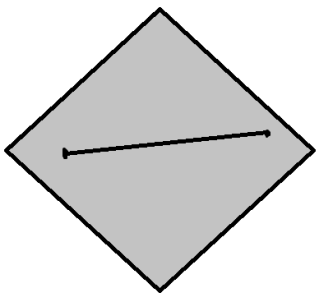
- $\nabla f(x^*) = 0$
- $\nabla^2 f(x^*)$ *pozitif yarı tanımlı*

ise, o zaman, x^**,* f*'in bir kesin yerel (strict local) minimumudur.*

Konvekslik varsayımı ile fonksiyonların Hessian matrislerinin yapısını inceleme zorluğundan kaçınılmış olur.

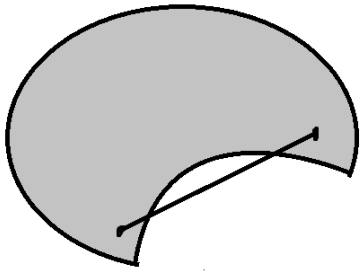
Tanım 2.10. Eğer C kümesi içindeki herhangi iki farklı nokta arasındaki doğru yine C kümesi içinde kalıyor ise, yani her $x_1, x_2 \in C$ ve $\theta \in \mathbb{R}$ için $\theta x_1 + (1-\theta)x_2 \in C$ oluyor ise, $C \subseteq \mathbb{R}^n$ kümesine bir *afin* küme denir.

Tanım 2.11. Eğer C kümesi içindeki herhangi iki farklı nokta arasındaki doğru parçası yine C kümesi içinde kalıyor ise, yani her $x_1, x_2 \in C$ ve $0 \leq \theta \leq 1$ olan her θ için $\theta x_1 + (1-\theta)x_2 \in C$ oluyor ise, C'ye bir *konveks* küme denir.



Şekil 2.1 Konveks küme

Kabaca, kümedeki her nokta, aralarındaki engelsiz bir düz yol boyunca diğer noktalar tarafından görülebiliyorsa, burada engellenmemiş küme içinde uzanıyor demektir, küme konvekstir. Her afin kümesi aynı zamanda konvekstir, çünkü içindeki herhangi iki farklı nokta arasındaki tüm doğruyu ve dolayısıyla noktalar arasındaki doğru parçasını da içerir. Şekil 2.1 ve şekil 2.2, $\mathbb{R}$'deki bazı basit konveks ve konveks olmayan kümeleri göstermektedir.



Şekil 2.2 Konveks olmayan küme

Tanım 2.12. Tüm $0 \leq \theta \leq 1$ ve $x, y \in C$ için

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$$

ise f fonksiyonu C kümesi üzerinde konvekstir denir. Fonksiyonun kesin konveks olması için eşitsizliğin "$<$" olması gereklidir.

Teorem 2.7. *Eğer f konveks bir fonksiyon ise her x^* yerel minimum noktası bir global minimumdur. Ayrıca, eğer f türevlenebilir ise her x^* durağan noktası bir global minimumdur [34].*

Lemma 2.1. *$f : \mathbb{R}^n \to \mathbb{R}$ olmak üzere, $\forall x, y \in C$ için*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

ise, f fonksiyonu konvekstir denir.

Tanım 2.13. $\mathcal{K} \subset \mathbb{R}^n$ olsun. Eğer tüm $x \in \mathcal{K}$ ve $\alpha > 0$ için $\alpha x \in \mathcal{K}$ oluyorsa, $\mathcal{K}$ (lineer) konidir denir.

Tanım 2.14. Eğer $\mathcal{K}$ konisi için tüm $\alpha, \beta > 0$ ve tüm $x, y \in \mathcal{K}$ için $\alpha x + \beta y \in \mathcal{K}$ oluyor ise $\mathcal{K}$ konveks koni olarak adlandırılır. O halde, bir $\mathcal{K}$ kümesi konveks ve koni ise konveks koni olur.

2.1.1 Kısıtlı Optimizasyon

Kısıtlı optimizasyon, bir dizi kısıtlamayı karşılarken bir probleme mümkün olan en iyi çözümü bulmak için kullanılan matematiksel ve hesaplamalı bir yaklaşımdır. Matematik, bilgisayar bilimi, mühendislik, ekonomi ve yöneylem araştırması dahil olmak üzere çeşitli akademik disiplinlerin ve pratik uygulamaların temel taşıdır. Bu yaklaşım, optimize edilecek bir amaç fonksiyonunun ve optimizasyon süreci sırasında uyulması gereken bir dizi kısıtlamanın tanımlanmasını içermektedir. Burada hedef, tüm kısıtlamaları karşılarken amaç fonksiyonunu maksimuma çıkaran veya minimuma indiren karar değişkenlerinin değerlerini bulmaktır.

Aşağıdaki optimizasyon problemi ele alınsın:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{ö.k.} \quad & h_i(x) = 0\,, i = 1,2,\ldots,m \\ & \ell_j(x) \leq 0\,, j = 1,2,\ldots,t. \end{aligned} \tag{2.3}$$

Denklem (2.3) ile tanımlanan optimizasyon probleminde $x \in \mathbb{R}^n$ karar değişkenini göstermektedir. Optimize edilecek (maksimum veya minimum) olan problemin amaç fonksiyonu $f : \mathbb{R}^n \to \mathbb{R}$ ile tanımlanmaktadır. $h_i(x)$ ve $\ell_j(x)$, sırasıyla, eşitlik kısıt fonksiyonunu ve eşitsizlik kısıt fonksiyonunu ifade etmektedir. Kısıtları sağlayan x vektörleri arasından amaç fonksiyonunu en küçük yapacak x^* vektörü bulunduğunda bu değere *optimum değer* adı verilmektedir. Kısıtları sağlayan herhangi bir noktaya *uygun nokta (feasible point)* denmektedir. Amaç fonksiyonunun tanımlı olduğu ve tüm kısıtların sağlandığı kümeye *uygun bölge (feasible region)* denir ve

$$\omega = \{x | h_i(x) = 0, i = \{1,2,\ldots,m\}\,; \ell_j(x) \leq 0, j = \{1,2,\ldots,t\}\} \tag{2.4}$$

ile ifade edilmektedir. Kısıtlı optimizasyon teorisinin gelişimine büyük ölçüde ışık tutan bir temel kavram, *aktif kısıt* kavramıdır. Eğer $\ell_j(x) = 0$ ise bu durumda $\ell_j(x) \leq 0$ eşitsizlik kısıtı x^* uygun noktasında **aktiftir** ve $\ell_j(x) < 0$ ise x^* olası (feasible) noktasında **aktif değildir** denir. O halde, optimizasyon problemlerindeki $h_i(x) = 0$ eşitlik kısıtlarının her biri uygun noktada her zaman aktiftir denebilir. Bir x^* olası (feasible) noktasında aktif olan kısıtlar, x^*'ın komşuluklarında ki uygulanabilirlik alanını kısıtlarken, diğer aktif olmayan kısıtların x^*'ın komşuluklarında hiçbir etkisi yoktur [36]. Bir eşitsizlik kısıtı *gevşek değişken* kavramı kullanılarak eşitlik kısıtına aşağıdaki şekilde dönüştürülebilir:

$$\ell_j(x) \geq 0 \iff \ell_j(x) + \xi = 0, \xi \geq 0. \tag{2.5}$$

Tanım 2.15. $x_i \in \mathbb{R}$, $\alpha \in i \in \mathbb{R}$ olmak üzere eğer

$$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n = 0 \tag{2.6}$$

eşitliğinde tüm $i = 1,2,\ldots,n$ için $\alpha_i = 0$ oluyor ise $x_1, x_2, \ldots, x_n$ vektörleri *lineer bağımsız*dır denir. Eğer 2.6 eşitliğinde $i \in 1,2,\ldots,n$ için en az bir tane $\alpha_i \neq 0$ ise $x_1, x_2, \ldots, x_n$ vektörleri *lineer bağımlı*dır denir.

Tanım 2.16. Lineer Bağımsızlık Kısıt Niteliği (LICQ) Denklem (2.3) ile verilen optimizasyon probleminde $x \in \omega$ ve $\mathscr{A}$ aktif kısıtların indis kümesi ele alındığında

eğer $\nabla h_i(x)$, $i \in \mathscr{A}$ lineer bağımsız ise *lineer bağımsızlık kısıt niteliği (LICQ)* sağlanır.

Tanım 2.17. Eğer bir x^* noktasında lineer bağımsızlık kısıt niteliği sağlanıyorsa x^* noktasına *düzenli nokta* denir.

Tanım 2.18. Eğer $\nabla h_1(x^*), \nabla h_2(x^*), \ldots, \nabla h_m(x^*)$ gradyan vektörleri lineer bağımsız ise $h_i(x^*)$ kısıtını sağlayan x^* noktasına *durağan nokta* denir [36].

Teorem 2.8. *S yüzeyinde $h_i(x) = 0$ ile tanımlı x^* noktasında teğet düzlemi,*

$$P = \{y : \nabla h_i(x^*)y = 0\} \tag{2.7}$$

ile tanımlanır [36].

Lemma 2.2. *Varsayalım ki x^*, $h_i(x) = 0$ eşitlik kısıtının bir durağan noktası ve bu eşitlik kısıtlarının ait olduğu $f(x)$ optmizasyon probleminin lokal extremum (minimum veya maksimum) noktası olsun. O halde,*

$$\nabla h_i(x^*)y = 0 \tag{2.8}$$

eşitliğini sağlayan her $y \in \mathbb{R}^n$

$$\nabla f(x^*)y = 0 \tag{2.9}$$

eşitliğini de sağlamalıdır [36].

2.1.2 Lagrange Teorisi

Kısıtlı optimizasyon teorisinde önemli bir yer tutan Lagrange teorisi, kısıtlar ve çözümler arasındaki boşluğu zarif bir şekilde kapatan matematiksel bir çerçeve olarak ortaya çıkmaktadır. Yöntemin temelinde, kısıt fonksiyonlarının amaç fonksiyonuna dâhil edilmesi ve bu sayede birinci mertebeden türev bilgisine dayanılarak çözüme gidilmesi yatmaktadır. Lagrange çarpanları (her kısıta atanan skaler değerler), kısıtlı optimizasyon probleminin bir kısıtsız optimizasyon problemine dönüştürülmesine olanak tanımaktadır.

Denklem (2.3) ile verilen optimizasyon problemi için Lagrange fonksiyonu

$$\mathscr{L}(x,u,v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{t} v_i \ell_i(x) \tag{2.10}$$

olarak tanımlanır. Burada, u_i eşitlik kısıtının ve v_i eşitsizlik kısıtının Lagrange çarpanları olarak adlandırılır. Optimizasyon teorisine göre, her problem, özgün (primal) ve dual olmak üzere, iki yüzü olan bir madalyon gibi düşünülebilir. Her bir

maksimizasyon probleminin bir minimizasyon karşılığı bulunurken, her minimizasyon probleminin de bir maksimizasyon eşleniği mevcuttur. Özgün ve dual problemler arasındaki ilişkilerin simetrikliğe dayanması gerekmektedir. Nitekim dual problemin duali yine başlangıçtaki özgün problemi işaret etmektedir. İlk ele alınan problem özgün (primal) olarak adlandırılırken, buna denk düşen modele dual model denir. Özgün problemi çözmek, belirli şartlar altında dual problemi çözmekle aynı sonuca yol açacaktır. Esas problemin optimal değeri, dual problemin optimal değerine eşit ya da ondan büyük olmak zorundadır. Eğer bir kesin eşitsizlik mevcutsa, bu durumda bir dual aralık (duality gap) oluştuğu kabul edilmektedir.

O halde, Lagrange dual fonksiyonu

$$g(u,v) = \min_{x\in\mathbb{R}^n} \quad \mathscr{L}(x,u,v) \tag{2.11}$$

ile ifade edilir. Dolayısıyla, (2.11)'a karşılık gelen dual problem aşağıdaki şekilde ifade edilir:

$$\begin{aligned} \max_{u\in\mathbb{R}^m, v\in\mathbb{R}^t} \quad & g(u,v) \\ \text{ö.k.} \quad & u \geq 0. \end{aligned} \tag{2.12}$$

Teorem 2.9 (Karush-Kuhn-Tucker Koşulları (KKT))**.** *Varsayalım ki x^*, denklem (2.3) optimizasyon probleminin lokal minimumu ve verilen kısıtlar üzerinde düzenli bir nokta ve, f ve h_i sürekli türevlenebilir fonksiyonlar olsun. O halde, aşağıdaki koşulları sağlayan, $u_i \in \mathbb{R}^m$ ve $v_i \in \mathbb{R}^t$ Lagrange çarpan vektörleri mevcuttur:*

- $\nabla\mathscr{L}(x^*,u,v) = 0$
- $v_j\ell_j(x^*) = 0, j = 1,2,\ldots,t$
- $v_j \geq 0, j = 1,2,\ldots,t$

Teorem 2.10 (İkinci Mertebeden Gerek Şart)**.** *Varsayalım ki f,h,ℓ ikinci dereceden sürekli türevlenebilir fonksiyonlar ve x^* bir düzenli nokta olsun. $u_i \in \mathbb{R}^m$ ve $v_i \in \mathbb{R}^t$ Lagrange çarpan vektörleri ile KKT koşulları sağlansın. Eğer bir x^* noktası için Lagrange fonksiyonunun Hessian matrisi, kısıtların etkin olduğu alanda pozitif yarı-tanımlı ise, x^* bir lokal minimumdur. Yani,*

$$w^T\nabla^2\mathscr{L}(x^*,u_i,v_i)w \geq 0, w \in \mathscr{T}(x^*) \tag{2.13}$$

dir. Burada $\mathscr{T}(x^*)$, x^* noktasındaki kısıtlı alanın teğet uzayıdır ve w bu uzaydaki herhangi bir vektördür.

Teorem 2.11 (İkinci Mertebeden Yeter Şart)**.** *Varsayalım ki f, h, ℓ ikinci dereceden sürekli türevlenebilir fonksiyonlar ve x^* bir düzenli nokta olsun. $u_i \in \mathbb{R}^m$ ve $v_i \in \mathbb{R}^t$ Lagrange çarpan vektörleri ile KKT koşulları sağlansın. Eğer bir x^* noktası için Lagrange fonksiyonunun Hessian matrisi, kısıtların etkin olduğu alanda pozitif yarı-tanımlı ise, x^* bir kesin lokal minimumdur. Yani,*

$$w^T \nabla^2 \mathscr{L}(x^*, u_i, v_i) w > 0, \forall w \neq 0, w \in \mathscr{T}(x^*) \tag{2.14}$$

dir. Burada $\mathscr{T}(x^*)$, x^* noktasındaki kısıtlı alanın teğet uzayıdır ve w bu uzaydaki herhangi bir vektördür.

2.2 Konveks Optimizasyon

Konveks optimizasyon, konveks fonksiyonların konveks kümeler üzerindeki minimumlarını bulmayı inceleyen matematiksel optimizasyonun bir alt dalıdır. Konveksliğin önemi, lokal minimumların aynı zamanda global minimumlar olması özelliğinde yatmaktadır, bu da optimizasyon sürecini basitleştirmektedir. Konveks optimizasyonun yararları disiplin sınırlarını aşmaktadır. Ekonomide, piyasa davranışını betimleyen ve kaynakların optimal dağıtımını tasarlayan modellerde merkezi bir role sahiptir. Kontrol teorisi, stabilite ve optimallik emreden sistem tasarımı için konveks optimizasyonun sağlamlığından (robustness) yararlanmaktadır. Hızla gelişen veri bilimi ve makine öğrenimi alanı, optimizasyon algoritmalarının yüksek boyutlu veri uzayları ile uğarşmak için, konveks optimizasyonun vazgeçilmez olduğunun başka bir kanıtıdır.

Bir konveks optimizasyon problemi, amaç fonksiyonunun konveks olduğu, uygun (feasible) kümenin konveks olduğu ve varsa tüm eşitlik kısıt fonksiyonlarının afin ve eşitsizlik kısıt fonksiyonlarının konveks olduğu bir problem olarak ortaya çıkmaktadır. Bir konveks optimizasyon problemi, $h_i(x) = a_i^T(x) - b_i$ olmak üzere,

$$\begin{aligned} \min \quad & f_0(x) \\ \text{ö.k.} \quad & f_i(x) \leq 0\,, i = 1, 2, \ldots, t, \\ & h_i(x) = 0\,, i = 1, 2, \ldots, m, \end{aligned} \tag{2.15}$$

şeklinde ifade edilebilir [37].

Konveks optimizasyon problemleri, birçok mühendislik ve bilim alanında karşılaşılan problemleri çözmek için kullanılmaktadır. Aşağıda bazı yaygın konveks optimizasyon problem türleri ve bunların tipik uygulamaları daha ayrıntılı bir şekilde ele alınmıştır.

2.2.1 Lineer Programlama (LP) Problemleri

Amaç fonksiyonunun belirli eşitlik ve eşitsizlik kısıtları altında minimum ya da maksimum değerinin arandığı kısıtlı optimizasyon problemlerinde, amaç ve kısıt fonksiyonlarının her biri lineer ise optimizasyon problemi lineer programlama problemi olarak tanımlanmaktadır. $x, c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ ve $A \in \mathbb{R}^{mxn}$ olmak üzere

$$\begin{aligned} \min \quad & c^T x \\ \text{ö.k.} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \tag{2.16}$$

lineer programlama probleminin uygun çözüm kümesi $S = \{x : Ax = b, x \geq 0\}$ konvekstir. S uygun çözüm kümesi iki boyutlu uzayda sonlu sayıda doğrularla sınırlanan konveks bir düzlemsel bölge, üç boyutlu uzayda sonlu sayıda düzlemlerle sınırlanan konveks bir bölge ve $n > 3$, n-boyutlu uzayda hiperdüzlemlerle sınırlanan polihedral konveks bir küme oluşturmaktadır.

2.2.2 Kuadratik Programlama (KP) Problemleri

Denklem (2.15) optimizasyon problemine, eğer, amaç fonksiyonu kuadratik (konveks) ve kısıt fonksiyonları afin ise, kuadratik programlama (KP) adı verilir. $Q \in S_+^n$ bir matris, A, mxn reel matris, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ ve $f(x) = \frac{1}{2}\langle x, Qx \rangle$ olmak üzere

$$\begin{aligned} \min \quad & f(x) \\ \text{ö.k.} \quad & Ax \leq b, \\ & x \geq 0, \end{aligned} \tag{2.17}$$

ile verilen problem kuadratik programlama problemi olarak adlandırılır. Q pozitif tanımlı bir matris olduğundan f kesin konveks ve uygun çözüm kümesi $S = \{x : Ax \leq b, x \geq 0\}$ konveks bir küme olacağından $f(x)$ fonksiyonun yalnızca bir tane x^* minimum noktası vardır. Eğer amaç fonksiyonu ile birlikte eşitsizlik kısıtları da kuadratik (konveks) olursa problem kuadratik kısıtlı kuadratik program olarak adlandırılır. Q matrisinin sıfır olduğu durumda problem, lineer programlama problemine dönüşecektir.

2.2.3 Yarı Tanımlı Programlama (YTP) Problemleri

K, pozitif yarı tanımlı kxk matrislerinin konisi olan S_+^k olduğunda, denklem (2.17) konik programlama problemine yarı tanımlı programlama (YTP) adı verilir ve

$G, F_1, F_2, \ldots, F_n \in S^k$, $A \in \mathbb{R}^{pxn}$ olmak üzere

$$\begin{aligned} \min \quad & c^T x \\ \text{ö.k.} \quad & x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \preceq 0, \\ & Ax = b, \end{aligned} \tag{2.18}$$

şeklinde ifade edilir. Burada tüm $G, F_1, F_2, \ldots, F_n$ matrisleri köşegen matris olduğunda problem lineer programlama problemine dönüşecektir.

2.2.4 Konveks Fark Programlama (KFP)

Bu bölümde, literatürde var olan ve analizimize yardımcı olacak matematiksel altyapılar açıklanacaktır.

Tanım 2.19. Γ, $\mathbb{R}^n$'de bir konveks küme olsun. $g, h : \Gamma \to \mathbb{R}$ olmak üzere

$$f : \Gamma \to \mathbb{R}, \quad f(x) = g(x) - h(x)$$

şeklinde ifade edilebilen gerçel değerli f fonksiyonu Γ üzerinde bir Konveks Fark (KF) foksiyonu olarak adlandırılır. KF programları, $f_i = g_i - h_i$ $(i = 1, \ldots, m)$ olacak şekilde,

$$\min\{f_0(x) : x \in \Gamma, f_i(x) \leq 0 \ (i = 1, \ldots, m)\}$$

biçiminde ifade edilmektedir [38].

Varsayalım ki $\eta(x)$ alt yarı sürekli ve konveks fonksiyon olmak üzere $\eta : \mathbb{R}^n \to \mathbb{R}$ ve $\eta^*(y) = \sup\{x^T y - \eta(x) : x \in \mathbb{R}^n\}$, $\eta(x)$'nın eşlenik fonksiyonu olsun. $\mathbb{X}$ ve $\mathbb{Y}$, $\mathbb{R}^n$'nin kapalı altkümeleri olacak şekilde KF program ve duali arasındaki bağlantı $\alpha = \inf\{g(x) - h(x) : x \in \mathbb{X}\} = \inf\{h^*(y) - g^*(y) : y \in \mathbb{Y}\}$ ile görülebilir.

Burada, α'nın sonlu olması, $dom\, g = \{x \in \mathbb{R}^n : g(x) < \infty\}$ olmak üzere $dom\, g \subset dom\, h$ and $dom\, h^* \subset dom\, g^*$ anlamına gelmektedir ; yani diğer bir deyişle, g sonlu olduğunda f'de sonludur. Bir fonksyion, eğer epigrafı $\mathbb{R}^{n+1}$'de sonlu sayıda kapalı yarı uzayın kesişimi ise, $\mathbb{R}^n$ üzerinde çokyüzlü (polyhedral) olarak adlandırılır [39]. Afin fonksiyonları ve gösterge (indicator) fonksiyonları çokyüzlü konveks fonksiyonlara örnek olarak düşünülebilir. KF programlamada $g(x)$ veya/ve $h(x)$ çokyüzlü ise, KF algoritmasının gerekli optimalite koşulları ve yakınsaması sağlanır.

Varsayalım ki $\partial\eta(x^0) = \{y \in \mathbb{R}^n : \eta(x) - \eta(x^0) \geq (x - x^0)^T y, \ \forall x \in \mathbb{R}\}$ olsun ve x^0' da η'nın bir alt-gradyanı tanımlansın. Burada, $\partial\eta(\cdot)$ bir kapalı konveks kümedir. Eğer $x^0 \in \text{int}(dom\,(\eta))$ ise, o zaman $\partial\eta(x^0) \neq \emptyset$.

Tanım 2.20. Genelleştirilmiş Karush-Kuhn-Tucker (KKK) Koşulu:

(i) Eğer x^* (primal) KF programın bir optimal çözümü ise, $\partial h(x^*) \subset \partial g(x^*)$ dir.

(ii) Eğer y^* dual KF programın bir optimal çözümü ise, $\partial g^*(y^*) \subset \partial h^*(y^*)$ dir.

$\mathscr{P}$ ve $\mathscr{D}$, sırasıyla, primal ve dual KF programının çözüm kümeleri olmak üzere

$$\bigcup\{\partial g^*(y) : y \in D\} \subset P \subset dom\, g, \bigcup\{\partial h(x) : x \in P\} \subset D \subset dom\, h^* \tag{2.19}$$

dir. Yeterli bir lokal optimalite koşulu için, x^*, $U(x)$ komşuluğunda

$$\partial h(x^*) \cap \partial g(x^*) \neq 0,\ \forall x \in U(x) \cap dom\, g \tag{2.20}$$

olacak şekilde $g - h$'nin lokal minimize noktası olduğu belirtilmelidir.

Aşağıdaki regülarize edilmiş konveks olmayan problem ele alınsın:

$$\underset{x \in \mathscr{C}}{\text{minimize}}\ f(x) + \rho \|x\|_0. \tag{2.21}$$

Burada, ρ, $\|x\|_0$ ceza(penalty) teriminin pozitif regülarize sabitidir, $\mathscr{C}$, $\mathbb{R}^n$'de bir konveks kümedir ve f ise $\mathbb{R}^n$ üzerinde bir KF fonksiyonudur. $\|x\|_0$ adım fonksiyonu olarak tanımlanabilecek kardinalite ölçüsünü ifade ettiğinden $\theta > 0$ yaklaşımın sıklığını kontrol eden parametre olmak üzere $z(x;\theta)$ sürekli fonksiyonu ile yaklaşık olarak hesaplanabilir. Bu durumda denklem (2.21) optmizasyon probleminin yeni formu:

$$\underset{x \in \mathscr{C}}{\text{minimize}}\ f(x) + \rho \sum_{i=1}^{n} z(x_i;\theta) \tag{2.22}$$

olacaktır.

Varsayım 1. [40] Varsayalım ki x değişkeni ve θ paremetresinin bir fonksiyon ailesi $z : \mathbb{R} \to \mathbb{R}$ aşağıdaki özellikleri sağlasın:

(a) Her $x \in \mathbb{R}$ için $\lim_{\theta \to \infty} z(x;\theta) = r(x)$.

(b) Herhangi $\theta > 0$ ve her $x \in \mathbb{R}$ için $z(x;\theta) = z(|x|;\theta)$, ve x değişkeni için $z(x;\theta)$, $[0, \infty)$ aralığında artar.

(c) Herhangi $\theta > 0$ için, $\phi(x;\theta)$ ve $\omega(x;\theta)$, $\mathbb{R}$ üzerinde sonlu konveks fonksiyonlar ve $z(x;\theta) = \phi(x;\theta) - \omega(x;\theta)$ olmak üzere $z \in \mathbb{R}$ bir KF fonksiyonudur.

(d) $\partial z(x;\theta) = \{p - h : p \in \partial\phi(x;\theta), h \in \partial\omega(x;\theta)\}$ olmak üzere her $x \in \mathbb{R}$, $\ell \in \partial z(x;\theta)$ için $x\ell \geq 0$.

(e) Herhangi $a \leq b$ ve $0 \notin [a,b]$ için, $\lim_{\theta\to\infty} \sup\{|y| : y \in \partial z(x;\theta), x \in [a,b]\} = 0$.

Bu durumda, varsayım (b) kullanılarak $\Omega = \{(x,y) : (x,y) \in \mathscr{C}, |x_i| \leq y_i (i = 1,2,\ldots,n)\}$ olacak şekilde denklem (2.22) optimizasyon probleminin başka bir denk formu aşağıdaki şekilde ifade edilir:

$$\underset{(x,y)\in\Omega}{\text{minimize}}\ f(x) + \rho \sum_{i=1}^{n} z(y_i;\theta). \tag{2.23}$$

Lemma 2.3. *[40] Bir $x^* \in \mathscr{C}$ noktası denklem (2.22) probleminin bir global (lokal) çözümüdür ancak ve ancak $(x^*, |x^*|)$, denklem (2.23) optimizasyon probleminin bir global (lokal) çözümüdür. Dahası, eğer (x^*, y^*), denklem (2.23) optimizasyon probleminin global bir çözümü ise o zaman, x^*, denklem (2.22) optimizasyon probleminin de bir global çözümüdür.*

Teorem 2.12. *[40] $\mathscr{M}$ ve $\mathscr{M}_\theta$, sırasıyla, denklem (2.22) ve (2.23) optimizasyon problemlerinin çözüm kümeleri olsun.*

(a) Varsayalım ki (θ_k), $\theta_k \to \infty$ olacak şekilde negatif olmayan sayıların bir dizisi olsun ve (x^k), herhangi k için $x^k \in \mathscr{M}_{\theta_k}$ olacak şekilde bir dizi olsun. Eğer $x^k \to x^*$ ise, o zaman $x^* \in \mathscr{M}$.

(b) Eğer C kompakt ise, o zaman herhangi $\epsilon > 0$ için bir $\theta(\epsilon) > 0$ mevcuttur öyle ki her $\theta \geq \theta(\epsilon)$ için $\mathscr{M}_\theta \subset \mathscr{M} + B(0,\epsilon)$ sağlanır.

(c) Eğer her $\theta > 0$ için $\mathscr{M}_\theta \cap \mathscr{K} \neq \emptyset$ olan bir K sonlu kümesi mevcut ise, o zaman her $\theta \geq \theta_0$ için $\mathscr{M}_\theta \cap \mathscr{K} \subset \mathscr{M}$ olacak $\theta_0 \geq 0$ mevcuttur..

Sonuç 2.3. *[40] Varsayalım ki z'nin $[0,\infty)$ üzerinde konkav, $\mathscr{C}$'nin en az bir köşeye sahip polihedral konveks bir küme ve f $\mathscr{C}$ üzerinde alttan sınırlı ve konkav olsun. O zaman, denklem (2.23)'de tanımlanan Ω kümesi de, en az bir köşeye sahip polihedral konveks bir kümedir. ϑ, Ω'nın köşe kümesi olsun ve $\overline{\mathscr{M}_\theta} = \{x : \exists y \in \mathbb{R}^n$ öyle ki $(x,y) \in \vartheta\}$ olarak tanımlansın. Bu durumda, tüm $\theta > 0$ için, $\emptyset \neq \overline{\mathscr{M}_\theta}$ olur ve bir $\theta_0 > 0$ vardır ki tüm $\theta \geq \theta_0$ için, $\overline{\mathscr{M}_\theta} < \mathscr{M}$ eşitsizliği sağlanır.*

f'in konkav, C çokyüzlü konveks kümede alttan sınırlı olması ve z'nin üstel bir yaklaşım olması durumunda orijinal problemin çözümü ile yaklaşık problemin çözümü

arasındaki tutarlılık [41] ve [42] tarafından çalışılmıştır fakat [40] çalışmasında Teorem 2.12 ile verilen genel sonuçların yanı sıra, Sonuç 2.3 ile daha güçlü sonuçlar elde etmektedir.

Bizim buradaki motivasyonumuz, orijinal çözüm ile yaklaşık çözüm arasındaki tutarlılığı bulmaktır; bu nedenle, bu problemlerin lokal çözümlerinin özelliklerini tanımlamamız gerekmektedir.

Lemma 2.4. *[40] Aşağıdaki özellikler doğrudur.*

(a) $x^* \in C$*, denklem (2.21) optimizasyon probleminin bir lokal optimumudur ancak ve ancak* x^**,* $\mathscr{C}^* = \{x \in C : x_i = 0,\ her\ i \neq \text{supp}(x^*)\}$ *olacak şekilde*

$$\underset{x \in \mathscr{C}^*}{\text{minimize}}\ f(x), \tag{2.24}$$

probleminin lokal optimumudur. Burada, $\text{supp}(x)$*,* x *desteğini ifade etmektedir.*

(b) Eğer $x^* \in C$*, denklem (2.21) optimizasyon probleminin bir lokal optimumu ise, o zaman bazı* $\overline{x^*} \in \partial f(x^*)$ *için*

$$\langle \overline{x^*}, x - x^* \rangle \geq 0 \quad \forall x \in \mathscr{C}^* \tag{2.25}$$

olur.

Problem (2.22) aşağıdaki şekilde ifade edilsin:

$$\underset{x}{\text{minimize}}\ G(x) - H(x). \tag{2.26}$$

O zaman, bir $x^* \in \mathscr{C}$ noktası için, koşul (2.20)

$$0 \in \partial G(x^*) - \partial H(x^*), \tag{2.27}$$

şeklinde yazılabilir ki bu ifade bazı $\overline{x^*} \in \partial f(x^*)$ ve $\overline{y_i^*} \in \lambda \partial_{r_\theta} f(x_i^*)$, $\forall\, (i = 1, \ldots, n)$ için

$$\langle \overline{x^*}, x - x^* \rangle + \langle \overline{y^*}, x - x^* \rangle \geq 0, \quad \forall x \in \mathscr{C}, \tag{2.28}$$

denklemine eşdeğerdir.

Böylece, lokal optimalliğin tutarlılık sonuçlarını vermeye hazır duruma gelinir.

Teorem 2.13. *[40]* $\mathfrak{T}$ *ve* $\mathfrak{T}_\theta$*, sırasıyla, (2.25) ve (2.28) koşullarını sağlayan* $x \in \mathscr{C}$*'in kümeleri olsun. Bu durumda aşağıdaki sonuçlar elde edilir:*

(a) Varsayalım ki $\theta_k \to \infty$ olacak şekilde (θ_k) negatif olmayan sayıların bir dizisi olsun ve herhangi k, $x^k \in \mathfrak{T}_\theta$ için (x^k) bir dizi olsun. Eğer $x^k \to x^*$ ise, o zaman $x^* \in \mathfrak{T}$.

(b) Eğer $\mathscr{C}$ kompakt ise, o zaman herhangi $\epsilon > 0$ için $\theta(\epsilon) > 0$, sayısı mevcuttur öyle ki her $\theta \geq \theta(\epsilon)$ için, $\mathfrak{T}_\theta \subset \mathfrak{T} + B(0, \epsilon)$ sağlanmaktadır.

(c) Her $\theta > 0$ için $\mathfrak{T}_\theta \cap \mathfrak{T} \neq \emptyset$ karşılanacak şekilde bir sonlu $\mathscr{K}$ kümesi mevcut ise, o zaman her $\theta \geq \theta_0$ için $\mathfrak{T}_\theta \cap \mathscr{K} \subset \mathfrak{T}$'nın sağlandığı bir $\theta_0 \geq 0$ sayısı mevcuttur.

2.2.5 İkinci Dereceden Konik Programlama (İDKP)

Bu bölümde, İDKP için temel tanımlar ve formülasyonlar tanıtılacaktır.

İDKP problemleri, bir afin lineer monifoldun ve ikinci dereceden (Lorentz) konilerin kartezyen çarpımının kesişimi üzerinde bir lineer fonksiyonu en aza indirmeyi içeren konveks optimizasyon problemleridir [43]. Lineer programlar, konveks kuadratik programlar ve ikinci dereceden kısıtlı konveks kuadratik programların tümü ve belirtilen kategorilere girmeyen diğer birçok program İDKP problemleri olarak tanımlanabilmektedir. Ayrıca, bir afin küme ile pozitif yarı-belirli matrislerin konisinin kesişimine ilişkin optimizasyon problemi olan YTP, özel bir durum olarak İDKP'yı içermektedir. Sonuç olarak İDKP, LP ile KP ve YTP arasında yer almaktadır. LP, KP ve YTP problemleri gibi İDKP problemleri, polinom zamanında İç Nokta Metodları (İNM'ler) kullanılarak çözülebilmektedir. İNM'ler, İDKP problemlerini çözmek için yineleme başına LP ve KP problemlerinden daha fazla ancak karşılaştırılabilir boyut ve karmaşıklıktaki YTP'larden daha az hesaplama çabası gerektirmektedir [44]. Aşağıdaki İDKP ele alınsın:

$$\begin{aligned} &\text{minimize} && f^T x \\ &\text{subject to} && \|A_i x + b_i\| \leq c_i^T x + d_i,\ i = 1, \ldots, N. \end{aligned} \tag{2.29}$$

Burada, $x \in \mathbb{R}^n$ değişken ve $f \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{(n_i-1)\times n}$, $b_i \in \mathbb{R}^{n_i-1}$, $c_i \in \mathbb{R}^n$, ve $d_i \in \mathbb{R}$ parametreleri ifade etmektedir. Kısıtta ortaya çıkan norm, standart Öklid normuna, yani, $\|z\| = (z^T z)^{1/2}$ karşılık gelir. $\|A_i x + b_i\| \leq c_i x + d_i$ kısıtı n_i boyutlu bir İDK kısıtı olarak adlandırılır çünkü m boyutlu standart veya birim İDK (Lorentz konisi) $\mathscr{K}_m = \left\{ \begin{bmatrix} u \\ t \end{bmatrix} \;\middle|\; u \in \mathbb{R}^{m_i-1}, t \in \mathbb{R}, \|u\| \leq t \right\}$ şeklinde tanımlanır. $m = 1$ için, birim ikinci-dereceden koni $\mathscr{K}_1 = \left\{ t \;\middle|\; t \in \mathbb{R}, 0 \leq t \right\}$ olarak tanımlanmaktadır. İkinci dereceden koninin bir kısıtını sağlayan noktalar kümesi, bir afin fonksiyona (mapping) göre birim ikinci dereceden koninin ters görüntüsüdür:

$$\|A_i x + b_i\| \leq c_i^T x + d_i \iff \begin{bmatrix} A_i \\ c_i{}^T \end{bmatrix} x + \begin{bmatrix} b_i \\ d_i \end{bmatrix} \in \mathscr{K}_{n_i}$$

ve konvekstir. Böylece, amaç fonksiyonu konveks olduğundan ve kısıtlar bir konveks küme tanımladığından, denklem (2.29) bir konveks programlama problemidir [43].

$\|u\| \leq t \iff \begin{bmatrix} tI & u \\ u^T & t \end{bmatrix} \succeq 0$, olduğundan dolayı ikinci dereceden koni, pozitif yarı-kesin matrislerden oluşan bir koniye gömülebilir, yani, ikinci dereceden bir koni kısıtı, lineer bir matris eşitsizliğine eşdeğerdir ($\succeq$, matris eşitsizliğini belirtir). Bu özellik kullanılarak, denklem (2.29) aşağıdaki gibi bir YTP olarak ifade edilebilir:

$$\begin{aligned} &\text{minimize} && f^T x \\ &\text{subject to} && \begin{bmatrix} (c_i^T x + d_i) I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{bmatrix} \succeq 0 \; i = 1, \ldots, N. \end{aligned} \tag{2.30}$$

İDKP'yi doğrudan çözen İNM'ları, probleme uygulanan YTP yöntemlerinden önemli ölçüde daha iyi en kötü-durum karmaşıklığına sahiptir: dualite boşluğunu kendisinin sabit bir kesrine indirmek için gereken yineleme sayısı İDKP algoritması ve YTP yaklaşımı için, sırasıyla, yukarıdan $O(\sqrt{n})$ ve $O(\sum_i n_i)$ ile sınırlanmıştır [45]. Ayrıca, pratikte en önemlisi, her yinelemenin önemli ölçüde daha hızlı olmasıdır: İDKP algoritmasında yineleme başına iş miktarı $O(n^2 \sum_i n_i)$ ve YTP için $O(n^2 \sum_i n_i^2)$ şeklindedir. Bu sayılar arasındaki fark, ikinci dereceden kısıtların n_i boyutu büyük olduğunda oldukça dikkat çekicidir. Denklem (2.29)'deki notasyonları sadeleştirmek için $u_i = A_i x + b_i, t_i = c_i^T x + d_i; i = 1, \ldots, N$ olsun.

Böylece, denklem (2.29) aşağıdaki şekilde ifade edilebilir:

$$\begin{aligned} &\text{minimize} && f^T x \\ &\text{subject to} && \|u_i\| \leq t_i, \; i = 1, \ldots, N, \\ & && u_i = A_i x + b_i, t_i = c_i^T x + d_i \; i = 1, \ldots, N. \end{aligned} \tag{2.31}$$

Dual İDKP Problemi

Denklem (2.29)'in duali

$$\begin{aligned} \text{maximize} \quad & -\sum_{i=1}^{N}(b_i^T z_i + d_i w_i) \\ \text{subject to} \quad & \sum_{i=1}^{N}(A_i t + z_i + c_i w_i) = f, \\ & \|z_i\| \leq w_i \; i = 1, \ldots, N, \end{aligned} \tag{2.32}$$

ile verilir. Burada, değişkenler $\boldsymbol{z_i} \in \mathbb{R}^{n_i - 1}$ ve $\boldsymbol{w} \in \mathbb{R}^N$ vektör olarak ifade edilir. Denklem (2.32) (dual İDKP) de bir konveks programlamadır çünkü amaç fonksiyonu konkav ve kısıtlar konveksdir. Aslında, bu denklem, denklem (2.31) ile aynı forma sahiptir. Bu durumda, dual İDKP'de aynı (primal) İDKP'da (denklem (2.29)) olduğu gibi eşitlik kısıtlamalarını ortadan kaldırarak yeniden biçimlendirilebilir.

Eğer primal İDKP probleminde tüm kısıtları sağlayan bir x var ise o zaman, denklem (2.29) uygun (fesaible) olur. Eğer x, kısıtları bir kesin eşitsizlik ile sağlıyorsa o zaman kesin uygundur denir. $\boldsymbol{z}$ ve $\boldsymbol{w}$ vektörleri, denklem (2.32)'deki kısıtları sağlıyorlarsa dual uygun (dual fesasible) olarak adlandırılır ve ayrıca $\|z_i\| \leq w_i \; i = 1, \ldots, N$'yi de sağlıyorlarsa kesin dual uygun olarak adlandırılır. (Kesin) Uygun $\boldsymbol{z_i}, \boldsymbol{w}$ var olması durumunda (Dual İDKP) denklem (2.32) (kesinlikle) uygundur denir. Denklem (2.29)'nin (Primal İDKP) optimal değerini, eğer problem uygun değil (infeasible) ise $\mathfrak{p}^* = \infty$ kabulu ile, $\mathfrak{p}^*$ olarak gösterelim ve denklem (2.32)'un (Dual İDKP) optimal değerini, eğer problem uygun değil (infeasible) ise $\mathfrak{d}^* = -\infty$ kabulu ile, $\mathfrak{d}^*$ olarak kabul edelim. Dual problemle ilgili temel doğrular üç durumda verilmektedir:

1. (**Zayıf Dualite (Week Duality)**) $\mathfrak{p}^* \geq \mathfrak{d}^*$,

2. (**Güçlü Dualite (Strong Duality)**) Eğer primal ya da dual problem kesin uygun (strictly feasible) ise, o zaman $\mathfrak{p}^* = \mathfrak{d}^*$,

3. Eğer primal ya da dual problem kesin uygun (strictly feasible) ise, o zaman primal ya da dual uygun noktalar vardır öyle ki bunlar optimal değerler olarak alınır.

2.3 İç Nokta Metodu (İNM)

Lineer olmayan programlama algoritmaları genellikle arama teknikleri olarak tanımlanan yinelemeli süreçleri kullanmaktadır. Her yinelemede algoritma, belirli

bir noktadan başlayarak aranacak yönü belirlemekte, daha sonra, yeni bir nokta belirlemek için bu doğrultuda ilerlemektedir. İç nokta metotları uygun (feasible) kümenin sınırına yalnızca limitte yaklaşmaktadır. Çözüme, uygun bölgenin içinden veya dışından yaklaşabilirler ancak hiçbir zaman bu bölgenin sınırında bulunmamaktadırlar. Bu arama tekniklerinin çeşitli biçimleri mevcuttur.

Bu bölümde, $f_0, f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}^m$ konveks ve ikinci dereceden sürekli türevlenebilir fonksiyonlar ve $A \in \mathbb{R}^{pxn}$, $rankA = p < n$ olmak üzere,

$$\begin{aligned} \min \quad & f_0(x) \\ \text{ö.k.} \quad & f_i(x) \leq 0,\ i = 1, \ldots, m, \\ & Ax = b, \end{aligned} \tag{2.33}$$

konveks optimizasyon problemini çözmek için iç nokta metotlarından bahsedilmiştir. Problemin çözülebilir olduğunu varsayalım yani bir x^* optimal noktası mevcut olsun. Ayrıca problemin kesin uygun (strictly feasible) olduğunu varsayalım. Bu durumda, x^* ile beraber KKT koşullarını sağlayan dual optimal $u^* \in \mathbb{R}^m$, $v^* \in \mathbb{R}^p$ Lagrange çarpanları var olacaktır. İç nokta metotları, denklem (2.33) optimizasyon problemini (veya KKT koşullarını) Newton yöntemini eşitlik kısıtlı problemler dizisine veya KKT koşullarının değiştirilmiş versiyonları dizisine uygulayarak çözmektedir.

Primal-dual iç nokta yöntemleri, özellikle yüksek doğruluk (accuracy) gerektiğinde, doğrusaldan daha iyi yakınsama sergileyebildikleri için genellikle bariyer yönteminden daha verimlidir. Lineer, kuadratik, ikinci dereceden koni, geometrik ve yarı tanımlı programlama gibi temel bazı problem sınıfları için, özelleştirilmiş primal-dual yöntemler bariyer yönteminden daha üstün performans göstermektedir. Lineer olmayan konveks optimizasyon problemleri için primal-dual iç nokta yöntemleri hala aktif araştırma konusu olması yanında büyük bir potansiyele sahiptirler.

Primal-dual iç nokta yöntemi için verilen algoritma aşağıdaki gibi açıklanabilir. Öncelikle, arama yönü (search direction) aşağıdaki modifiye KKT koşulları tanımlansın:

$$r_t(x, u, v) = \begin{bmatrix} \nabla f_0(x) + Df(x)^T u + A^T v \\ -\text{diag}(u) f(x) - (1/t)\mathbf{1} \\ Ax - b \end{bmatrix}$$

ve $t > 0$. Burada, $f : \mathbb{R}^n \to \mathbb{R}^m$ ve f'in türev matrisi Df

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}, \quad Df(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix}$$

şeklinde olmalıdır. Eğer x, u, v, $r_t(x,u,v) = 0$ eşitliğini sağlıyorsa ve $f_i(x) < 0$ ise, o halde $x = x^*(t)$, $u = u^*(t)$ ve $v = v^*(t)$ olmalıdır. Dahası, x primal (birincil) uygun ve u, v dual uygun olmak üzere dual aralık (duality gap) m/t şeklindedir. r_t'nin ilk blok bileşeni,

$$r_{\text{dual}} = \nabla f_0(x) + Df(x)^T \lambda + A^T v,$$

dual residü (dual residual) ve son blok bileşeni ise $r_{\text{pri}} = Ax - b$, primal residü (primal residual) olarak adlandırılır. Orta blok,

$$r_{\text{cent}} = -\text{diag}(\lambda) f(x) - (1/t)\mathbf{1},$$

merkeziyet residüdür (centrality residual). (x,u,v) noktasında $f(x) \prec 0$, $u \succ 0$ olacak şekilde sabit t için $r_t(x,u,v) = 0$ doğrusal olmayan denklemleri çözmek için Newton adımı ele alınsın. Mevcut nokta ve Newton adımı, sırasıyla, $y = (x,u,v)$, $\Delta y = (\Delta x, \Delta u, \Delta v)$ şeklinde ifade edilsin.

Newton adımı, lineer denklemlerle karakterize edilir:

$$rt(y + \Delta y) \approx rt(y) + D_{rt}(y)\Delta y = 0,$$

yani,

$$\Delta y = -D_{rt}(y)^{-1} rt(y).$$

x, u ve v cinsinden ifade edilerse,

$$\begin{bmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m u_i \nabla^2 f_i(x) & Df(x)^T & A^T \\ -\text{diag}(u) Df(x) & -\text{diag}(f(x)) & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} -r_{\text{cent}} \\ -r_{\text{dual}} \\ -r_{\text{pri}} \end{bmatrix} \tag{2.34}$$

olacaktır. *Primal-dual arama yönü* $\Delta y_{\text{pd}} = (\Delta x_{\text{pd}}, \Delta u_{\text{pd}}, \Delta v_{\text{pd}})$, denklem (2.34)'ün çözümü olarak tanımlanır. Ayrıca, $A^T = b$ eşitliğini sağlıyorsa, yani primal uygunluk residüsü r_{pri} sıfırsa, o zaman, $A\Delta x_{\text{pd}} = 0$ olur, bu da Δx_{pd}, (primal) uygun bir yön

tanımlar: herhangi bir s için, $x + s\Delta x_{\text{pd}}$ $A(x + s\Delta x_{\text{pd}}) = b$ eşitliği sağlanır.

Primal-dual iç nokta yönteminde, iterasyonlar, limitte algoritma yakınsadıkça hariç olmak üzere, $x^{(k)}$, $u^{(k)}$ ve $v^{(k)}$ kesinlikle uygun (feasible) olmak zorunda değildir. Bu, algoritmanın k adımıyla ilişkilendirilen bir düallik aralığının $\eta^{(k)}$ kolayca hesaplanamayacağı anlamına gelmektedir. Bunun yerine $f(x) \prec 0$ ve $u \succeq 0$ olan herhangi bir x için yedek dual aralığı,

$$\hat{\eta}(x, \lambda) = -f(x)^T u$$

şeklinde taımlanır. Yedek aralık $\hat{\eta}$, eğer u ve v dual olarak uygunsa, yani $r_{\text{pri}} = 0$ ve $r_{\text{dual}} = 0$ ise, dual aralık olacaktır. Yedek dual aralık $\hat{\eta}$ ' ya karşılık gelen t parametresinin değeri $m/\hat{\eta}$ şeklinde olacaktır.

Algorithm 1 Primal-Dual İç Nokta Metodu

1: $f_i(x) < 0$, $i = 1, \ldots, m$, $u > 0$, $v > 1$, $\epsilon_{feas} > 0$, $\epsilon > 0$ koşullarını sağlayan x verildiğinde;
2: **repeat**
3: t belirle. Set $t := vm/\hat{\eta}$.
4: Primal-dual arama yönünü hesapla Δy_{pd}.
5: Doğru arama (line search) ve güncelleme (update). $s > 0$ için adım uzunluğunu belirle ve $y := y + s\Delta y_{\text{pd}}$ olarak belirle.
6: **until** $\|r_{\text{pri}}\|_2 \leq \epsilon_{feas}$, $\|r_{\text{dual}}\|_2 \leq \epsilon_{feas}$ ve $\hat{\eta} < \epsilon$

3
Makine Öğrenmesi

3.1 Giriş

Birinci nesil bilgisayarların 1945 yılında ortaya çıkması, bilim insanlarını makinelerin zekâ ile donatılması konusunda düşünmeye itmiştir. 1950 yılında geliştirdiği Turing testi ile Alan Mathison Turing, bilgisayarların düşünme yetilerine bir kriter getirerek modern bilgisayarların temelini atmıştır [46].

Hesaplama biliminin geniş alanında, makine öğrenmesinin ortaya çıkışı bir paradigma değişimine işaret etmektedir. Makine öğrenmesi, özünde, makinelerin verilerden öğrenme ve yorumlama yeteneği ile donatılabileceği fikrine dayanmaktadır. Yapay zekanın önemli bir dalı olarak ortaya çıkan makine öğrenmesi, detaylı talimatların her işlemi belirlediği geleneksel hesaplama yöntemlerinin aksine, makinelere tıpkı insanlar gibi deneyimlerden öğrenmeyi öğretmeye benzeyen daha organik bir yaklaşım sunmaktadır.

Makine öğrenmesindeki ilk araştırmalar, özellikle örüntü tanıma alanında yoğunlaşmıştır. En basit tek katmanlı Yapay Sinir Ağı (YSA) modeli olan “Perceptron", 1950’lerde Frank Rosenblatt tarafından önerilmiştir. İlk makine öğrenmesi kitabı 1965 yılında yayımlanmış, 1968’de Vapnik ve Chervonenkis (VC) [47] istatistiksel öğrenme teorisini geliştirmiş, VC entropi ve VC boyutu gibi temel kavramları tanıtmıştır. Bu kavramlar, 1971’de fonksiyonel uzayda büyük sayılar kanununun bulunmasını ve öğrenme işlemiyle ilişkisinin tanımlanmasını sağlamıştır.

Son yıllarda, derin öğrenme (deep learning) adı verilen bir alt dal, büyük veri setlerini kullanarak karmaşık problemleri çözmek için güçlü algoritmalar geliştirmiş ve özellikle nesne tanıma, konuşma tanıma gibi alanlarda önemli başarılar elde etmiştir. Günümüzde, yapay zekâ uygulamaları sağlık, finans, ulaşım gibi çeşitli sektörlerde yaygın olarak kullanılmaktadır [48, 49].

3.1.1 Öğrenme nedir?

Öğrenme, çok ve çeşitli tanımları olan, günümüzde hala çok yönlü araştırmaların yapıldığı bir konudur. Zaman içerisinde bilim insanları tarafından çeşitli tanımlar yapılmıştır. Örneğin, Simon "Öğrenme, bir sistemdeki, aynı görevin tekrarı veya aynı popülasyondan alınan başka bir görevin daha verimli ve daha etkili bir şekilde yapılmasını sağlayan herhangi bir değişikliktir" derken, Michalski öğrenmeyi, deneyimlenen şeyin temsillerini yapılandırmak veya değiştirmek olarak tanımlamaktadır [50].

İnsan bağlamında öğrenme, deneyim, çalışma veya öğretilme yoluyla bilgi veya becerilerin birikmesidir. Bu kavramı makinelere aktarırken "öğrenme", makinenin görev performansını aşamalı olarak geliştirme yeteneğini ifade etmektedir. Bu gelişme, insan müdahalesinin veya programlamanın bir sonucu değil, daha ziyade makinenin verileri işlemesinin ve algoritmalarını geliştirmesinin bir sonucudur. Bunu insan hayatına benzetmek gerekirse müzik aleti çalmayı öğrenmek düşünülebilir. İlk başta, tökezlenebilir, yanlış notalara basılabilir ve zamanlama konusunda zorluk yaşanabilmektir. Ancak uygulama (veriler), bir öğretmenin rehberliği (algoritma) ve geri bildirimi (model ayarlaması) ile daha çok gelişilebilir. Zamanla, daha karmaşık parçalar çalınabilir ve hatta notaları görmeden önce notaların nasıl çalınacağı tahmin edebilmektedir (tahmin doğruluğu).

Peki bir makine tam olarak nasıl "deneyimler" veya "öğrenir"? Makineler için öğrenme, insanların ders kitaplarından, derslerden veya deneyimlerden öğrenmesine benzemektedir. Makineler sayılar, görüntüler, metinler ve sesler dahil olmak üzere çeşitli veri türlerinden öğrenmektedir. Veriler ne kadar çeşitli ve zengin olursa, başarılı öğrenme şansı da o kadar artmaktadır. Makine öğreniminin temel kavramı, makinenin beynine eşdeğer olan, modeldir. Eğitim aşamasında model verilere maruz kalmakta, tahminlerde bulunmakta ve tahminlerinin doğruluğuna göre kendini ayarlamaktadır. Bu yinelemeli süreç, modelin tahminleri kabul edilebilir bir doğruluk seviyesine ulaşana veya modelin daha fazla gelişemeyeceği anlaşılana kadar devam etmektedir. Eğitimden sonra modelin öğrenmesi sabitlenmez. Doğruluğunu onaylamak için yeni, görülmemiş verilere karşı test edilir. İyi performans gösterirse dağıtıma hazır hale gelmekte aksi takdirde önceki deneyimlerini kullanarak kendini geliştirmek üzere eğitim aşamasına geri dönmektedir.

Makine öğrenimi, istatistik, olasılık, veri madenciliği, örüntü tanıma ve yapay zekâ gibi disiplinlerle sıkı bir ilişki içerisindedir. Makine öğrenimi algoritmalarının etkinliğinde önemli bir faktör, verinin ifade veya gösterim biçimidir. Özellikle, verinin doğrusal olarak ayrılabilir olması, makine öğrenimi yöntemlerinde kritik bir faktördür.

Makine öğrenmesinin günlük yaşamdaki başlıca uygulama alanları öneri sistemleri (örneğin Netflix, spotify), finansal hizmetler (örneğin kredi skoru tahminleme), sağlık sektörü (örneğin hastalık tahmini), otomotiv (örneğin sürücüsüz araçlar), üretim (örneğin hata tespiti), dijital güvenlik (örneğin dolandırıcılık tespiti) ve sesli asistanlar (örneğin siri, google asistan) olarak özetlenebilir.

Makine öğreniminin temel prensibi, geçmiş bilgiler ve gözlemlerden yararlanarak gelecekteki olaylar hakkında bilgi edinme ve bu bilgiyi kullanarak gelecekteki durumları tahmin etme yeteneğidir. Bu karar alma sürecinde en temel unsur, eldeki bilgidir. Bilgi, bir amaca yönelik olarak işlenmiş veriyi temsil eder. Veriyi bilgiye veya anlamlı hale dönüştürme süreci, veri madenciliği kavramıyla ilişkilidir ve bu bağlamda veri analizi olarak adlandırılır. Veri madenciliği, veri setlerinde gizli olan kuralları, ilişkileri veya örüntüleri keşfetmeye yönelik bir disiplindir.

Veri Madenciliği (VM), genellikle bağımsız değişken ($y \in \mathbb{R}$) sayısının çok yüksek olduğu veri setlerinde sıkça kullanılan bir alan olarak öne çıkmaktadır. Makine öğrenimi, istatistik ve veri madenciliği arasında sıkı bir bağ vardır. Makine öğrenimi yöntemleri, veri madenciliği algoritmalarının temelini oluşturur.

Bu bağlamda, bilgi çıkarımı ve öğrenme süreçleri, istatistiksel yöntemlerle desteklenmiş veri madenciliği teknikleri aracılığıyla gerçekleştirilir. Sonuç olarak, makine öğrenimi, istatistik ve veri madenciliği alanları arasında kesişen bir noktada bulunarak bilgi çıkarımında önemli bir rol oynar.

3.2 Makine Öğrenmesi Çeşitleri

Makine öğrenmesi, geçmiş deneyimleri kullanarak öğrenme süreçlerini yönlendirmeye çalışan bir alan olarak karşımıza çıkmaktadır. Öğrenim yöntemlerine göre farklılaşan makine öğrenmesi, denetimli öğrenme (supervise learning), denetimsiz öğrenme (unsupervised learning), yarı denetimli öğrenme (semi-supervise learning) ve takviyeli öğrenme (reinforcement learning) olmak üzere çeşitli alt gruplara ayrılır.

Bu tezde denetimsiz öğrenme problemlerinden kümeleme problemi kullanıldığından bu bölümün alt başlıkları olarak yalnızca denetimli ve denetimsiz öğrenme üzerine odaklanılacaktır.

3.3 Denetimli Öğrenme

En yaygın teknik olan denetimli öğrenmede algoritmalar etiketli (labeled) veriler kullanılarak eğitilmektedir. Veri kümesindeki her örnek doğru çıktıyla

eşleştirilmektedir. Algoritma, eğitim verileri üzerinde yinelemeli olarak tahminler yapmakta ve kendisini doğru çıktılar yönünde ayarlamakta, böylece doğru tahminler üretmenin en iyi yolunu öğrenmektedir. Denetimli öğrenmenin en yaygın tekniği olan algoritmalar, etiketli (*labeled*) veriler kullanarak eğitilir. Bu durumda, veri kümesindeki her örnek, doğru çıktıyla ilişkilendirilmiştir. Algoritma, eğitim verileri üzerinde tekrarlayan tahminler yapar ve kendisini doğru çıktılar doğrultusunda ayarlar. Bu süreç, en iyi tahminleri üretme yeteneğini öğrenme amacı taşır.

Matematiksel olarak etiketleme kavramı, genellikle (X, Y) çifti ile ifade edilir, burada X girdi özelliklerini, Y ise bu girdilere karşılık gelen etiketleri temsil eder. Denetimli öğrenme algoritmaları, bu (X, Y) çiftleri üzerinden modelini eğitir ve doğru tahminler yapmak için bu ilişkiyi öğrenir.

Denetimli öğrenme sürecinde, bir eğitmen sisteme dış müdahalede bulunur. Bu sürecin temel hedefi, girdi (input) ve çıktı (output) değerleri arasındaki ilişkiyi kavramaktır. Böylece, yeni girdi değerleri için çıktı değerleri tahmin edilebilir. Girilen değer ile istenen değer arasındaki fark, önceden belirlenen hata değerinden küçük olduğunda eğitim devam eder. Sistem, girdiler için istenen doğruluğa ulaştığında eğitim tamamlanır ve süreç sona erer. Sınıflandırma algoritmaları, regresyon, yapay sinir ağları, destek vektör makineleri denetimli öğrenme örnekleri olarak verilebilir.

3.3.1 Sınıflandırma (Clasification)

Denetimli öğrenmenin bir alt kümesi olarak sınıflandırma, büyük ölçüde her girişin açıkça etiketlendiği veri kümelerine dayanmaktadır. Bu veri kümelerindeki her veri noktası, belirli bir sınıf veya kategorinin örneği olarak hizmet etmektedir. Sınıflandırmanın temel amacı, yalnızca bu etiketleri bilinen veriler üzerinde kopyalamak değil, aynı zamanda yeni, görünmeyen veri noktalarını doğru bir şekilde etiketlemek için bu bilinen veri kümesinden genelleme yapmaktır. İyi bir sınıflandırıcıyı vasat bir sınıflandırıcıdan ayıranda bu genellemedir. Aslında bu, hassas bir dengeleme eylemidir; Eğitim verilerine çok fazla uyum sağlayan bir model (aşırı uyum) yeni veriler üzerinde düşük performans gösterebilirken, çok genelleştirilmiş bir model (yetersiz uyum) doğru sınıflandırma için gereken hassasiyetten yoksun olabilmektedir. Sınıflandırma sanatı ve bilimi, doğru dengeyi sağlayan modeller oluşturmakta, verilerin karmaşıklıklarında gezinerek doğasında var olan kategorizasyonları ortaya çıkarmakta yatmaktadır.

Sınıflandırma problemlerinde, eğitim kümesi (training set) ve sınıf etiketleri (class label) önceden bilinmektedir. Kullanılan algoritmadan bağımsız olarak, sınıflandırma modelinin oluşturulmasında kullanılan örüntülerin boyutu, veri kümesini temsil

eder. Veri kümesinin bir alt kümesi, eğitim kümesi olarak adlandırılır, bu kümenin veri kümesinden çıkartılması ile test kümesi elde edilir. Eğitim kümesi $V = \{(x_1, y_1), \ldots, (x_{n_m}, y_m)\}$ olmak üzere, burada girdi (input) $(x_1, x_2, \ldots, x_n)$, $x_i \in \mathbb{R}^n$ ve çıktı (output) ya da etiket $(y_1, y_2, \ldots, y_n)$, $y_i \in \{-1, +1\}$ olarak adlandırılır.

Sınıflandırma problemlerinde yaygın olan yöntemler, k-en yakın komşu (k-nearest neighbor), destek vektör makineleri (DVM-support vector machines), karar ağaçları (decision trees), yapay sinir ağları (artificial neural network) olarak ifade edilebilir [51].

3.4 Denetimsiz Öğrenme

Denetimli öğrenmeden farklı olarak denetimsiz öğrenme, etiketlenmemiş (unlabeled) verilerle ilgilenmektedir. Burada amaç doğru tahminler üretmek değil, verilerdeki temel yapıları veya kalıpları tespit etmektir. Denetimsiz öğrenme bağlamında, sistemin öğrenmesine rehberlik eden bir eğitmen bulunmaz. Sisteme sadece girdi değerleri ($(x_1, x_2, \ldots, x_n)$, $x_i \in \mathbb{R}^n$) sağlanır ve öğrenmesi beklenir, ancak doğru çıktılarla ilgili herhangi bir bilgi verilmez. Temel amaç, girdi değerlerinin öznitelikleri üzerinden öğrenme gerçekleştirmektir. Denetimsiz öğrenme, kümeleme (clustering) ve ilişkilendirme (association) problemlerine uygulanmaktadır.

3.4.1 Kümeleme (Clustering)

Başlangıçta, kümeleme basit bir gruplama görevi gibi görünse de daha derine inildiğinde verilerin anlattığı gizli hikayeleri ortaya çıkarmak için bir arayışa dönüşmektedir. Kümeleme, denetimsiz bir doğada çalışmakta ve önceden tanımlanmış etiketlerin rehberliği olmadan, içsel benzerliklere dayalı olarak veri içinde doğuştan gelen gruplamaları aramaktadır.

Kalabalık bir şehir meydanında çok sayıda insanı gözlemlediğimizi düşünürsek, bazıları aileler, bazıları arkadaş grupları ve diğerleri belki de iş arkadaşları olabilir. Hiçbir işaret bu grupları etiketlemese de, etkileşimler, kıyafetler veya yaş gibi ince ipuçlarına dayanarak kimin hangi gruba ait olduğu tahmin edilebilmektedir. Kümeleme ise, veri bilimi bağlamında benzer şekilde çalışmaktadır. Veri noktalarını gözlemlemekte ve belirli özelliklere dayanarak hangi veri noktalarının ortak hikayeleri paylaştığını belirlemektedir.

Kümeleme, bir veri kümesini alt kümelere (sub-clusters) ayıran denetimsiz bir öğrenme tekniğidir. Temel amaç, birbirlerine diğer alt kümelerdekilerden daha çok benzeyen veri noktalarını gruplamaktır.

Tanım 3.1. (Yakınlık) Bir X veri matrisinde i. ve j. satırlar arasındaki yakınlık değeri $d(i,j)$ ile ifade edilmektedir. Eğer tüm (i,j) değerleri için

i. $d(i,i) = 0$,

ii. $d(i,j) = d(j,i)$,

iii. $d(i,j) \geq 0$,

iv. $d(i,j) = 0 \iff i = j$,

v. $d(i,k) \leq d(i,j) + d(j,k)$, $(i,j = 1,2,\ldots,n)$, $(k = 1,2,\ldots,p)$,

sağlanıyorsa, $d(i,j)$ *uzaklık fonksiyonu* olarak adlandırılır.

"Benzerlik" kriteri, alana özgü metriklere veya genel mesafe ölçütlerine bağlıdır. Kümelemede benzerlik (similarity) ve farklılık (dissimilarity) sayısal değeri, uzaklık ölçüsü (distance measure) olarak adlandırılmaktadır. Uzaklıkların hesaplanmasında en çok kullanılan uzaklık ölçüleri aşağıdaki gibi sıralanabilir [52]:

1. Minkowski Uzaklığı
$$d(i,j) = \left(\sum_{k=1}^{p} |x_{ik} - y_{jk}|^r \right)^{\frac{1}{r}}, \tag{3.1}$$

2. Manhattan City Blok Uzaklığı ($r = 1$ *durumu*)
$$d(i,j) = \sum_{k=1}^{p} |x_{ik} - y_{jk}|, \tag{3.2}$$

3. Öklit (Euclidean) Uzaklığı ($r = 2$ *durumu*)
$$d(i,j) = \sqrt{\sum_{k=1}^{p} (x_{ik} - y_{jk})^2}, \tag{3.3}$$

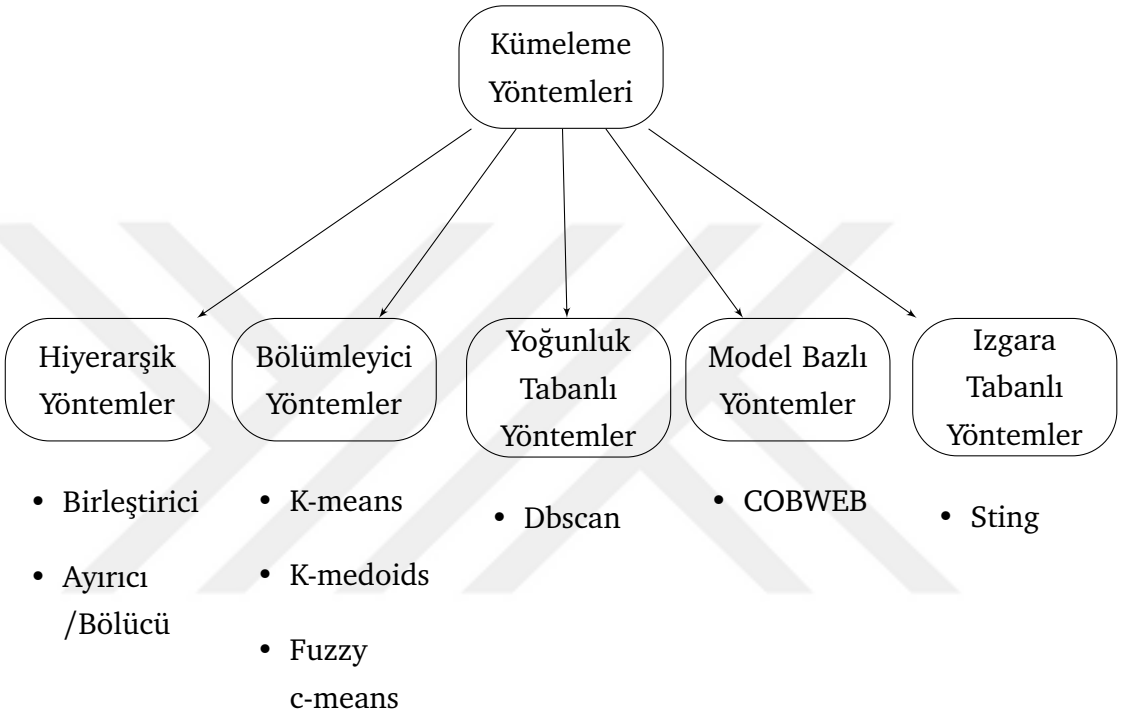
   ($\| \cdot \|$ ile de gösterilmektedir).

4. Supremum Uzaklığı
$$d(i,j) = \max_{1 \leq k \leq d} |x_{ik} - y_{jk}|. \tag{3.4}$$

En sık kullanılan ölçüm yöntemlerinden biri olan öklid uzunluğu, gerçek bir üçgenin hipotenüs uzunluğudur. Öklid ve manhattan uzaklık fonksiyonları, genellikle

benzerliklerin belirlenmesinde kullanılmaktadır. Uzaklık fonksiyonunun büyük bir değer alması, benzerliğin düşük olduğunu, küçük bir değer alması ise benzerliğin yüksek olduğunu göstermektedir. Bu uzaklık ölçüleri, değişkenlerin ölçü birimlerine göre farklılık göstermektedir. Bir ölçü biriminde iki birey arasındaki uzaklık, ölçü birimi değiştirildiğinde en uzak iken en yakın hale gelebilmekte; bu nedenle, uzaklık hesaplanmadan önce değişkenlerin standartlaştırılması gerekmektedir [53].

Kümelemede temel yöntemler Şekil 3.1'deki gibi ifade edilebilir:



Şekil 3.1 Kümeleme Yöntem ve Algoritmaları

Bu yöntemler, benzer özelliklere sahip veri noktalarını tanımlamak ve gruplamak için kullanılmaktadır. Her bir küme, içindeki veri noktalarının birbirine benzemesi ve diğer küme elemanlarından farklı olması temel prensibiyle oluşturulmaktadır.

- Hiyerarşik kümeleme yöntemleri (Hierarchical Clustering), birbirine benzer veri noktalarını birleştirerek veya tam tersine ayırarak kümeler oluşturur. Bu yöntem, veri gruplamalarını bir ağaç olarak görselleştirerek analistin veri içindeki hiyerarşik yapıları anlamasını sağlar. Bu sayede, veri setindeki hiyerarşik ilişkiler görsel olarak anlaşılır hale gelir, analistin veri içindeki örüntüleri keşfetmesine yardımcı olur. *Birleştirici yöntemler (agglomerative)*, benzer özelliklere sahip küçük kümeleri birleştirirken, *ayırıcı/bölücü yöntemler (divisive)* , tüm veri setini içeren büyük bir kümeden başlayarak bu kümelere ayrılır. Tüme varım (bottom up), yani birleştirici yaklaşımda, başlangıçta tüm

nesneler birbirinden ayrıdır. Başlangıç noktası olarak, eldeki verinin her bir öğesi ayrı bir küme olarak ele alınır. Ardından, benzer özelliklere sahip kümeler birleştirilerek tek bir küme elde etme amaçlanır. Tümden gelim (top-down) yaklaşımda ise ayrıştırıcı bir strateji izlenir. Bu yaklaşımda, başlangıçta tek bir küme bulunur. Her aşamada, uzaklık veya benzerlik matrisine göre nesneler ana kümeden ayrılarak farklı alt kümeler oluşturulur. Süreç tamamlandığında, her veri bir küme haline gelir.

AGNES (AGglomerative Nesting) algoritması, veri kümesini başlangıçta her örneği ayrı bir küme olarak kabul ederek başlamaktadır. Ardından, her adımda kümeler arasındaki uzaklıklara bakarak en yakın iki kümeyi birleştirmektedir. Bu aşağıdan yukarı doğru bir birleştirme yapısını izler. Öte yandan, DIANA (DIvisive ANAlysis) algoritması, başlangıçta tüm örnekleri aynı kümede kabul etmekte ve her adımda birbirine en uzak iki kümeyi belirlemeye çalışmaktadır. Bu algoritma yukarıdan aşağı doğru bir bölünme yapısını benimsemektedir.

- Bölümleyici yöntemler, veriyi belirli bir sayıda küme merkezi oluşturarak bu merkezlere en yakın şekilde bölen algoritmaları ifade etmektedir. Bu yöntemler, veri setini homojen gruplara ayırmak için kullanılmaktadır. Örneğin, en yaygın kullanılan denetimsiz öğrenme algoritmalarından biri olan *k-ortalama*, belirlenen küme sayısına göre veri setini kümelere ayırmayı hedefler. Merkez tabanlı bir teknik olan k-ortalama kümeleme (k-means Clustering), verileri "k" sayıda kümeye ayırmaktadır. Bu yöntemde, belirli bir sayıda küme merkezi seçilir, ardından veri noktaları bu merkezlere olan uzaklıklarına göre gruplandırılır. Her veri noktası en yakın olduğu küme merkezine atanır ve bu işlem iteratif olarak devam eder. Sonuç olarak, benzer özelliklere sahip veri noktaları aynı kümede toplanır. *k-medoids*, k-ortalamaya benzemektedir, ancak küme merkezlerini veri noktalarından seçer, bu da daha dirençli ve gerçekçi kümeler üretmektedir. *Fuzzy c-ortalama* ise her veri noktasının belirli bir ağırlık değeriyle her kümeye ait olma olasılığını belirler, bu nedenle belirgin sınırlar yerine bulanık kümeler üretmektedir. Bu ağırlık değeri, 0 (hiç ait olmama) ile 1 (tamamen aitlik) arasında değerler alabilir. Matematiksel olarak, bir bulanık küme içindeki bir nesne, herhangi bir küme için 0 ile 1 arasında değerler alan bir ağırlık değeriyle ilişkilendirilir. Bu yöntemde, bir nesnenin tüm kümelerdeki ağırlık değerlerinin toplamı 1 olmalıdır.

- Yoğunluk tabanlı yöntemler, veri noktalarının yoğunluklarına odaklanarak kümeleri tanımlamaktadır. Yoğunluk tabanlı bir küme, düşük yoğunluğa sahip bir bölge ile çevrili ve yüksek nesne yoğunluğuna sahip bir alanı içeren bir bölgedir. Bu şekilde tanımlanma genellikle kümeleme işlemi sırasında kümelerin düzensiz veya birbirleriyle geçmiş olduğu durumlar ve aynı zamanda gürültü

ve aykırı değerlerin bulunduğu durumlar için kullanılır. Özellikle *Gürültülü Uygulamaların Yoğunluk Tabanlı Mekansal Kümelenmesi DBSCAN (Density-Based Spatial Clustering of Applications with Noise)*, yoğunluğa dayalı bir yaklaşım olmaktadır. Birbirine yakın bir şekilde paketlenmiş veri noktalarını bir araya gruplandırmakta ve düşük yoğunluklu bölgelerde tek başına bulunan noktaları aykırı değer olarak işaretlemektedir.

- Model tabanlı yöntemler, veriyi belirli bir model veya yapay zeka algoritması üzerinden analiz ederek kümeleme yapmaktadır. Model bazlı bir yöntem olan *COBWEB*, veriyi bir ağaç yapısı içinde kümeler ve alt kümeler olarak modelleyen bir yapay zeka yaklaşımıdır.

- Izgara tabanlı yöntemler, veriyi düzenli bir ızgara üzerine yerleştirerek kümeleme yapmaktadır. *Sting*, veriyi düzenli bir ızgara üzerine yerleştirerek kümeleme yapan bir yöntemdir. Özellikle coğrafi veriler için etkili bir seçenektir.

Kümeleme alanında, belirlenen kümelerin bütünlüğünü ve uygunluğunu değerlendirmek büyük önem teşkil etmektedir. Kümelemenin doğasında olan denetimsiz metodolojisi göz önüne alındığında, güvenilir ölçümlere sahip olmak hayati önem taşımaktadır. Siluet Katsayısı, Dunn Endeksi ve Davies-Bouldin Endeksi gibi araçlar, kümelenme kalitesini ölçmek, iç uyum ve kümeler arası ayrılma gibi yönlere bakmak için sıklıkla kullanılır. Kümelemenin pratik uygulamaları çeşitli endüstrileri kapsamaktadır. Ticari sektörde, işletmelerin müşterileri satın alma alışkanlıklarına, tercihlerine veya demografik ayrıntılarına göre kategorilere ayırmasına yardımcı olarak daha kişiselleştirilmiş bir pazarlama yaklaşımına olanak tanımaktadır. Biyoloji bilimlerinde ise flora ve faunanın farklı özelliklere göre sınıflandırılmasını kolaylaştırarak potansiyel evrimsel bağlara ışık tutmaktadır. Geniş veri havuzları için kümeleme, belgelerin temalara göre sistematik bir şekilde düzenlenmesini sağlayarak içerik erişimini kolaylaştırmaktadır. Ek olarak, dijital görüntüleme alanında, görüntüleri ayırt edilebilir bölümlere ayırarak görüntü analizini ve optimizasyonunu desteklemektedir.

3.5 Topluluk Öğrenmesi

3.5.1 Giriş

Toplu sistemler olarak da bilinen çoklu sınıflandırıcı sistemler, son birkaç on yılda hesaplamalı zeka ve makine öğrenimi topluluklarında popülerlik kazanmıştır [54–57]. Bu popülerliğin nedeni, topluluk sistemlerinin çok çeşitli problem alanlarında ve gerçek dünya uygulamalarında inanılmaz derecede etkili ve çok yönlü olduğu

kanıtlanmış olmasından ileri gelmektedir. Yöneylem araştırması, programlama ve zaman yönetimi, şehir planlama, tedarik zinciri yönetimi, optimizasyon ve mühendislik [58–63] gibi çeşitli sorunları çözmek için kullanılmakta, matematiksel modeller ve optimizasyon tekniklerini bilinçli kararlar vermek için kullanmakta ve topluluk sistemlerinden de yararlanabilmektedir. Topluluk öğrenimi, bir grup öğreniciyle birlikte tahminde bulunur ve bu tahmin sonuçlarını birleştirir. Birkaç öğrenici ile işbirliği nedeniyle daha güvenilir olduğu ampirik olarak kanıtlanmıştır. Toplulukların tek modellere göre avantajı, daha fazla robust ve doğruluk sağlamak olarak rapor edilmiştir [64]. Topluluk öğrenme yöntemlerini kullanarak, bağımsız sınıflandırıcıların gruplandırılması daha doğru sonuçlar vermektedir. Torbalama (Bagging) ve artırma (Boosting), literatürde en iyi bilinen yaklaşımlardır.

Torbalama (Bagging)

Torbalama, Breiman tarafından 1996 yılında geliştirilmiştir [65]. En eski ve en basit ancak etkili olan bir topluluk tabanlı algoritmadır. Bu yöntemde, belirli bir t örneği için, yer değiştirme ile rastgele örneklemeler yapılır. Bu işlem k kez tekrarlandığında, bazı örneklerin birden fazla kez görünebileceği ve diğerlerinin hiç görülmeyebileceği t örnekli bir veri seti elde edilir. Torbalama, aşırı öğrenmeyi önlemekte ve özellikle karar ağacı tekniklerinde kullanılan regresyon ve sınıflandırma modellerinde kullanılmaktadır.

Artırma (Boosting)

Artırma, bir öğrenme sisteminin performansını artırmak için kullanılan bir sınıflandırma yaklaşımıdır. Ancak regresyon için de kullanılabilmektedir. Artırma, her biri rastgele tahminden biraz daha iyi performans gösteren zayıf sınıflandırıcılardan oluşan bir topluluktan daha düşük eğitim (training) hatası elde edebilen, güçlü bir sınıflandırıcıya dayalı yinelemeli bir yaklaşımdır [66]. Artırma algoritmasının temel prensibi eğitim kümesindeki ağırlık kümelerini veya dağılımı korumaktır. Algoritma, bir temel öğreniciyi eğiterek başlamaktadır ve ardından temel öğrenicinin sonuçlarına dayalı olarak eğitim örneklerinin dağılımını değiştirerek hatalı bir şekilde sınıflandırılan öğrenicinin yanlış sınıflandırılmış örneklere odaklanmasını sağlamaktadır. İlk temel öğrenenin eğitimi sonrasında, değiştirilmiş eğitim örnekleri ikinci temel öğrenenin eğitimi için kullanılmakta ve sonuçlar, eğitim örneklerinin dağılımını yeniden değiştirmek için kullanılmaktadır. Bu süreç, temel öğrenenlerin sayısı önceden belirlenmiş bir değeri geçene kadar tekrarlanır ve bu noktada ağırlıklandırılır ve birleştirilir. Uyarlanabilir artırma (Adaptive boosting) en sık kullanılan artırma yöntemidir [66].

3.5.2 Topluluk Budaması (Ensemble Pruning)

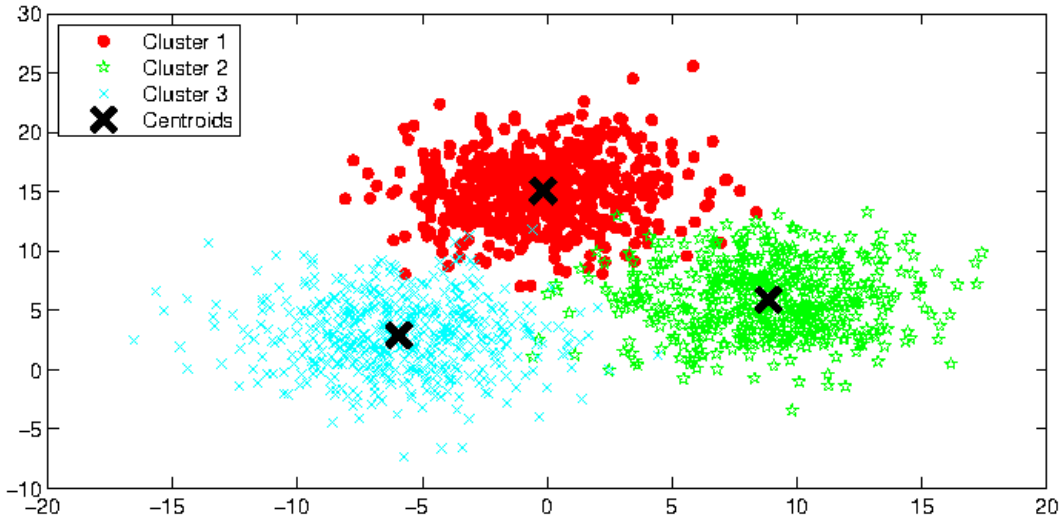
Topluluk budama, bir toplulukta eğitilmiş bireysel modellerin tümünü birleştirmek yerine, topluluğu oluşturan alt modellerin bir alt kümesini seçmeye çalışan bir yöntemdir [67]. Ayrıca, topluluk budama yöntemleri bir topluluğun karmaşıklığını azaltmak için kullanılmaktadır. Makine öğrenimi ve veri madenciliği, topluluk budama algoritmalarından yoğun bir şekilde yararlanmaktadır. Daha önce belirtildiği gibi, artırma ve torbalama, temsili topluluk yaklaşımlarının iyi bilinen örnekleridir. Bununla birlikte, dikkate değer katkılarına rağmen, torbalama gibi mevcut teknikler genellikle çok büyük topluluklar oluşturmaktadır ve bu da yüksek bellek maliyetlerine, hesaplama harcamalarına ve etkinlikte ara sıra düşüşlere neden olmaktadır [67, 68].

Topluluk budama, topluluğun alt küme uzayında en iyi çözümü aramak olarak değerlendirilebilir. Ancak, çapraz doğrulama ile kapsamlı bir arama kullanarak en iyi alt grubu belirlemek zaman ve çaba gerektirmektedir. Bu nedenle, sıralı birleştirme (ordered aggregation) [69–71], genetik algoritmalar aracılığıyla optimizasyon tabanlı modeller [19], YTP [3], ikinci dereceden programlama [72] gibi birkaç yaklaşık arama stratejisi ve lineer programlama [73], optimuma yakın alt toplulukları bulmak için önerilmiştir. [74]'a göre, topluluk budama yaklaşımı genel olarak sıralama tabanlı, kümeleme tabanlı, optimizasyon tabanlı ve diğerleri olmak üzere dört türe ayrılabilmektedir.

Sıralama tabanlı yaklaşımlar, topluluktaki sınıflandırıcıları bazı değerlendirme ölçütlerine göre bir kez sıralamaya ve nihai topluluk için en üst sıradaki sınıflandırıcıları seçmeye çalışmaktadır. Çok sayıda çalışmada, kappa budama [75], oryantasyon budama [69] ve denetimsiz marj tabanlı sıralama topluluk budaması [76] gibi birçok sıralamaya dayalı grup budama algoritması bulunmaktadır.

Kümelemeye dayalı budama yöntemleri, nihai bir topluluk oluşturmak için birkaç temsili bireysel sınıflandırıcıyı belirlemeye çalışmaktadır. Bu yaklaşımlar genel olarak iki aşamada çalışmaktadırlar. İlk aşamada, düşük tahmin çeşitliliğine sahip bireysel sınıflandırıcıları, tahminlerine dayalı olarak bir dizi kümede birleştirmek için bir kümeleme tekniği kullanılmaktadır. Bu arayış için birçok kümeleme tekniği kullanılmaktadır [77–79]. Popüler kümeleme algoritmalarından biri, k-ortalama, Şekil 3.2'de örneklendirilmiş merkez tabanlı bir kümeleme yöntemidir [78]. Kümelere ait alt topluluklar, ikinci aşamada nihai bir topluluk oluşturmak için birleştirilmektedir. Bu aşamaya ulaşmak için çeşitli teknikler önerilmiştir. [80] çalışmasında, her kümede diğer kümelerden en fazla uzaklığa sahip sınıflandırıcıyı seçmektedir. [81] çalışmasında ise, topluluk üyelerinin bir alt kümesi, alt grubun doğruluğu azalmaya başlayana kadar her bir küme içindeki bireysel sınıflandırıcıları yinelemeli olarak eleyerek seçilmektedir. Son olarak, çalışma [82]'de, her bir kümenin ağırlık merkezi

nihai topluluğa dahil edilmiştir.



Şekil 3.2 Rastgele Verilerle K-Ortalama Algoritması Örneği.

Optimizasyona dayalı yöntemler, topluluk budamasını, genelleştirme performansını optimize eden (örneğin, ayrı bir doğrulama setindeki doğruluk) orijinal grubun alt kümesini keşfetmek için bir optimizasyon problemi olarak tasvir edilir. Sezgisel (Heuristik) optimizasyon, matematiksel programlama ve olasılıksal yöntemler gibi çeşitli optimizasyon stratejileri geliştirilmiştir. Genetik Algoritma tabanlı Seçici Topluluk (GASEN) [19], bir topluluğun N boyutlu bir ağırlık vektörü ile ilişkilendirildiği sezgisel optimizasyon yönteminin bir örneğidir. GASEN, ağırlık vektörü popülasyonunda küçük ağırlıklara sahip bireysel sınıflandırıcıları filtrelemek için bir genetik algoritma kullanır. [19]'deki çalışmaya ek olarak, toplu budamaya tepe tırmanma yöntemi [24, 28, 83, 84] gibi çok sayıda ek sezgisel optimizasyon algoritması uygulanmıştır. Topluluk budama, matematiksel programlama yaklaşımları kullanılarak bir matematiksel optimizasyon problemi olarak formüle edilebilir. Örneğin, [3] çalışması, topluluk budamayı ikinci dereceden bir tamsayı programlama problemi olarak kabul etmiş ve yaklaşık bir çözüm elde etmek için YTP'yi kullanmıştır, bununla beraber [72] çalışmasında, topluluk budama ikinci dereceden bir programlama (KP) olarak düzenlenmiştir. Budama modelini çözmek için [29, 30]'daki çalışmalarda sırasıyla disiplinli konveks-konkav programlama (DKKP) ve bir konveks fark algoritması (KFA) kullanılmıştır. [3]'daki çalışmadan motive olarak, toplu budamayı İDKP olarak modellenmiştir.

3.6 Karşılaştırılan Yöntemler

3.6.1 Ortak Kriter Yöntemi (Joint Criteria Method)

Her bir kümeleme çözümü arasındaki benzerlikleri ölçen Normalleştirilmiş Karşılıklı Bilgi (NKB) fonksiyonu [85] aşağıdaki şekilde verilmektedir:

$$I(X;Y) = \sum_{y \epsilon Y} \sum_{x \epsilon X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \tag{3.5}$$

$$NMI(X,Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}. \tag{3.6}$$

Burada, $I(X,Y)$, X ve Y kümeleri arasındaki karşılıklı bilgi fonksiyonu, $p(x)$ ve $p(y)$ marjinal dağılım fonksiyonları ve $H(X)$, $H(Y)$ ise X ve Y kümelerine karşılık gelen entropi fonksiyonlarıdır. C_i, $i-inci$ kümeleme çözümünü, $NMI(C,C_i)$ ise denklem (3.6)'de ifade edilen fonksiyonu temsil etmek üzere, $E = \{C_1, C_2, \ldots, C_n\}$ farklı kümleme çözümlerinin kütüphanesi olsun [86]. Normalleştirilmiş Karşılıklı Bilgi Toplamı (NKBT)

$$SNMI(C,E) = \sum_{i=1}^{n} NMI(C,C_i), \tag{3.7}$$

olarak tanımlanır. Burada, $SNMI(C,E)$ doğruluğu (accuracy) ölçen fonksiyondur [5].

K topluluk boyutuna sahip bir öğrenme modeli kümesi için maksimize edilen bir amaç fonksiyonu olan "Toplanmış Amaç Fonksiyonu (TAF)" olarak bilinen denklem (3.8) aşağıda verilmiştir. İlk terim doğruluk (accuracy) değerlendirmesini, ikinci terim ise çeşitlilik değerlendirmesini yapmaktadır ve α parametresi her bir terimin ağırlığını düzenlemektedir [5].

Ortak kriter yöntemi, kalite ve çeşitlilik terimlerini aşağıda verilen aynı amaç fonksiyonunda birleştirmektedir:

$$\alpha \sum_{i=1,\ldots,K} SNMI(C_i,L) + (1-\alpha) \sum_{i \neq j} (1 - NMI(C_i,C_j)). \tag{3.8}$$

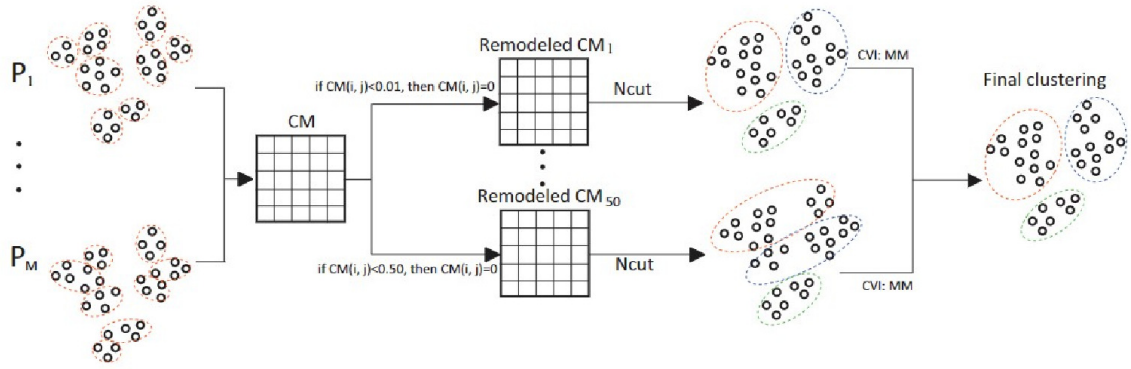
Burada, $SNMI(C_i,L)$, L kümeleme çözümlerinin geniş bir kütüphanesi olmak üzere C_i kümeleme çözümünün kalitesini hesaplamaktadır. Çeşitlilik (diversity) ise $(1 - NMI(C_i,C_j))$ ile ölçülmektedir.

3.6.2 Kümele ve Seç Yöntemi (Cluster and Select Method)

Fern ve Lin [5] tarafından geliştirilmiş olan *Kümele ve Seç* yöntemi , topluluk kütüphanesindeki her bir kümeleme çözümünü bir varlık (entity) olarak ele almakta ve birbirleriyle ilişkilerini analiz etmektedir. Bu yöntemde, gereksiz tekrarları azaltmak adına, C_1 ve C_2 benzer kümeleme çözümleri olup C_1'in toplulukta seçilmediği durumlarda, C_2'de, yüksek kalitede olduğunda bile, seçilmemektedir. Benzerliklerine göre gruplandırılan kümeleme çözümlerinin her bir grubundan yalnızca bir kümeleme çözümü seçilmektedir. Kümeleme çözümü kütüphanesi k gruplara ayrılmakta ve her bir grup, belirli bir k boyutlu topluluk için ilgili çözümlerin bir kümesinden oluşmaktadır. Kümeleme çözümleri k gruplarına ayrıldıktan sonra, her bir grup içindeki en yüksek kaliteli kümeleme çözümü, $SNMI$ ile gösterilen bir kalite ölçüsü kullanılarak topluluk alt kümesi elde etmek için seçilmektedir. NKB ikili matrisi, kümeleme çözümlerini k gruba ayırmak için spektral kümelemeye tabi tutulmaktadır [87].

3.6.3 Kanıt Biriktirme Yöntemi (The Evidence Accumulation Method)

Orijinal veri uzayından bir birliktelik matrisine kernel dönüşüm olan *Kanıt Biriktirme Yöntemi* (KBY), kümeleme yöntemlerindeki baz bölümlerden veri toplamaktadır. Ancak, bu dönüşüm küme yapısı bilgilerinin kaybına neden olabileceğinden, literatürde önerilen bazı yöntemler, verileri geri almak ve topluluk sürecine yeniden eklemeye çalışmaktadır. [88] çalışmasında, bazı kanıtların birleşim (co-association) matrisinden çıkarılmasının daha doğru kümeleme sonuçlarına yol açabileceği ilginç bir olgu tanıtılmıştır. İlk akla gelen sezgisel açıklama, orijinal birleşim matrisindeki bazı kanıtların gürültü (noise) olabileceğidir, bu da son kümelemeyi olumsuz olarak etkileyecektir. Ancak, bu tür kanıtları tespit etmek zordur, matrisden çıkarmak ise daha da zordur. Bu sorunu ele almak için, Zhong ve arkadaşları [88], negatif kanıtların sadece baz bölümlerinde bazen ortaya çıkmasından ötürü, düşük oluşum sıklıklarına sahip çoklu düzey kanıtlarını çıkarmaktadırlar. Daha sonra, normalleştirilmiş kesim kullanarak çoklu kümeleme sonuçları elde etmektedirler.



Şekil 3.3 KBY'ye genel bakış [88].

Şekil 3.3 CM'nin birlikte ilişkilendirme matrisini ve MM'nin yeni tanımlanmış dahili geçerlilik indeksini temsil ettiği bu yaklaşıma genel bir bakış sunmaktadır. Her grup içinde en yüksek kaliteye sahip kümeleme çözümü, bir kalite metriği olan NKB kullanılarak topluluk alt kümesini belirlemek için seçilir.

3.6.4 PrunedOPT

Aşağıda verilen, doğruluk ve çeşitlilik arasındaki dengeyi optimize eden model, konveks bir küme üzerinde KF fonksiyonlarının minimizasyonudur:

$$\min_{x} \quad \{\tau\|x\|_2^2 - [x^T(\widetilde{G} + \tau I)x - \rho\|x\|_0]\}, \tag{3.9}$$

burada $\|.\|_0$ sıfır normuna ve $\|.\|_2$ Öklidyen normuna karşılık gelmektedir [29].

$$\|x\|_0 = \lim_{\epsilon \to 0} \sum_{i=1}^{n} \frac{\log(1+\mid x_i \mid/\epsilon)}{\log(1+1/\epsilon)}$$

olarak yaklaştırılır ve sonra, $\epsilon > 0$ seçilerek limit ihmal edilir. Bir düzenleme sabiti ile cezalandırılarak, model topluluk boyutu k'dan bağımsız hale getirilir; ancak en iyi k kapsamlı bir arama yoluyla seçilirse, algoritmanın performansı hız açısından basitçe kabul edilemez derecede yavaş olur. Bu nedenle, [29] çalışmasında gerçek değerlerin bazı aralıkları içinde tanımlanabilecek yeni bir parametre ρ tanıtılarak arama alanını azaltan bir formülasyon önermiştir.

4
Önerilen Modeller

4.1 Konveks Fark Programlama ile Topluluk Kümeleme Seçilimi

Bu bölümde, Konveks Fark (KF) programlama teorisi kullanılarak genel seyrek kuadratik ve konveks olmayan yaklaşımların gerekli ve yeterli optimallik koşullarını incelenmiştir. Uygulama olarak, makine öğreniminde topluluk seçim problemi ele alınarak gerekli optimallik koşullarının sağlandığı gösterilmiştir. Ayrıca, bu çalışmada analiz edilen teoriye bir uygulama olarak, topluluk seçim probleminde amaç fonksiyonunun ikinci terimi belirli bir üstel terim [41] olarak değiştirilmiştir ve sıfır norma konveks olmayan bir yaklaşım uygulanmıştır.

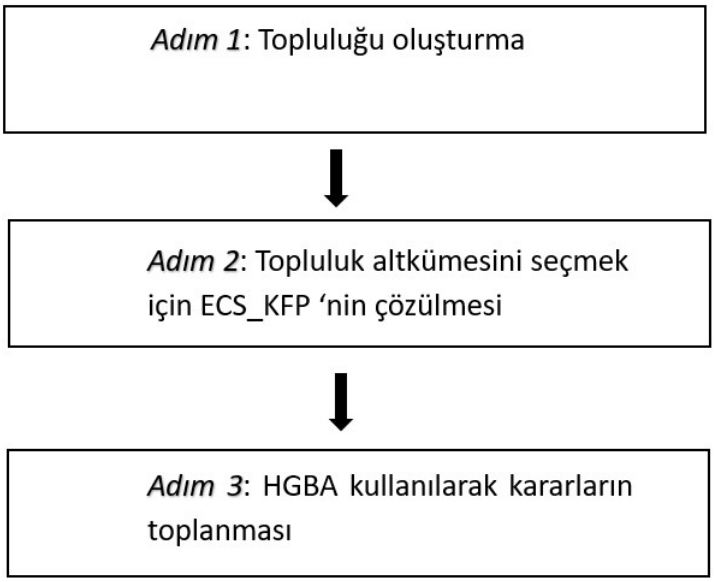
Seyreklik (Sparsity), hesaplamayı daha hızlı hale getirmek ve son derece büyük ölçekli hesaplamalara olanak sağlamak için yüksek boyutlu verileri daha küçük bir boyutla temsil etmeyi ve yeniden yapılandırmayı mümkün kılmaktadır. Kardinalite anlamına gelen l_0-norm ile amaç fonksiyonunu cezalandırmak, optimizasyon problemlerinde seyrekliği uygulamanın en yaygın yoludur. Seyrek optimizasyon problemleri yıllardır makine öğreniminde kullanılmaktadır. Sıfır-norm içeren çalışmalar, konveks optimizasyon, konveks olmayan optimizasyon ve uygun yaklaşımlarla konveks olmayan tam reformülasyon (exact reformulation) ile çözülebilmektedir. Bu kategoriler sırasıyla ele alınırsa, konveks yaklaşımlar için bilinen bir yöntem En Az Mutlak Küçültme ve Seçim Operatörü (EAMKSO) (LASSO-Least Absolute Shrinkage and Selection Operator)'dur, l_0-normu l_1-normu ile değiştirmektedir [89]. İkinci kategori, $\|x\|_0$'ı konveks olmayan sürekli bir fonksiyonla, örneğin üstel konkav bir fonksiyon [41] veya bir logaritmik fonksiyon [90] ile yaklaştırılmasını içermektedir. Sürekli konveks olmayan bir program tarafından dışbükey olmayan tam bir reformülasyon olan son kategori, çoğunlukla Destek Vektör Makinelerinde (DVM) öznitelik seçimi gibi makine öğrenimi uygulamalarında kullanılmaktadır [91]. Büyük ölçekli veri kümeleri için çözülemeyecek olması nedeniyle, [92] ve [93] çalışmalarında KF programlamayla kesin bir ceza yöntemi önerilmiştir.

KF yaklaşım problemlerinin makine öğreniminde öznitelik seçimi, topluluk seçimi, genelleştirilmiş özdeğer problemi, l_1-DVM, l_∞-DVM, ...vb. gibi birçok

uygulaması bulunmaktadır. Örneğin, [12]'de genelleştirilmiş özdeğer problemi (GÖP), seyrek KF programlama olarak formüle edilmiş ve büyütme-azaltma (majorization-minimization) yaklaşımı ile çözülmüştür. Çalışmalarında, diller arası belge alımında (cross-language document retrieval) uygulanan kanonik korelasyon analizinde seyrek GÖP probleminin özel uygulamasını değerlendirmişlerdir. [94] çalışmasında ise, en çok tercih edilen yöntemlerden biri olarak bilinen DVM çözümü için seyrek KF Algoritmaları (KFA) yöntemi kullanılmıştır.

Bu bölümde, topluluk oluşturulması, önerilen modelle topluluk budaması ve son olarak kümelerin toplanması (aggregation) açıklanacaktır (bknz. Şekil 4.1).

Adım 4.1.2'de önerilen model denklem (4.12), ECS_KFP olarak adlandırılmıştır. Topluluk kümesi seçimini oluşturmak için üç temel adım kullanılmıştır. İlk adım, bir temel kümeleme algoritması olarak k-ortalama kullanılarak topluluğun oluşturulması, ikinci adım, topluluğun bir alt kümesini elde etmek için ESC_KFP ile topluluk budaması ve son adım ise, Adım 1'de oluşturulan topluluğun kümeleme kararlarının Hiper Grafik Bölümleme Algoritması (HGBA) [86] kullanılarak toplanmasıdır (aggregate).



Şekil 4.1 Önerilen modelin akış şeması[95].

4.1.1 Adım 1 : Topluluğu Oluşturma

Üretim adımı, farklı kümeleme çözümleri elde etmek üzerine kuruludur. Çok sayıda temel küme çözümü oluşturmak için, aynı kümeleme algoritmasının çeşitli parametreleriyle oluşturulan farklı kümeleme algoritmaları veya çeşitli kümeleme algoritmaları, çeşitli nesne temsilleri, veri toplamanın farklı alt kümeleri ve nesnelerin

farklı alanlara izdüşümleri kullanılabilmektedir [96]. Topluluk kütüphanesinin üretim adımı bu çalışmada üç adımdan oluşmaktadır:

i. Rastgele Başlatma

ii. Rastgele Öznitelik Seçimi

iii. Rastgele Lineer Projeksiyon

Burada, k-ortalama, adım (i)'ye karşılık gelen rastgele parametrelerle k-ortalama algoritmasını çalıştırarak çeşitli kümeleme çözümlerinin elde edildiği bir temel model olarak kullanılmıştır. Daha sonra (ii) adımında, rastgele seçilen her öznitelik (feature) için kümeleme çözümleri topluluğun kitaplığına eklenmiştir. Son olarak, adım (iii), verilerin geometrik yapısı korunurken boyutunu düşürmek için, rastgele bir izdüşüm matrisi kullanılarak orijinal özniteliklerin daha düşük boyutlara indirgenmesi için kullanılmıştır.

4.1.2 Adım 2 : ECS_KFP ile Topluluk Budama

Doğruluk ve çeşitlilik değiş tokuşunu optimize eden,

$$\begin{aligned} \text{maks} \quad & x^T \widetilde{D} x \\ \text{ö.k.} \quad & \sum_i x_i = k, \\ & x_i \in \{0,1\}, \end{aligned} \tag{4.1}$$

modeli ele alınsın. Burada, $\widetilde{D}$, aday modeller arasındaki ilişkileri dikkate alan önceden belirlenmiş

$$D_{ij} = \begin{cases} D_{ii} = SNMI(C, C_i)\ ,\ (i = 1, ..., n) \\ D_{ij} = 1 - NMI(C_i, C_j)\ ,\ (j = 1, ..., n) \end{cases} \tag{4.2}$$

şeklinde D matrisinin normalize edilmiş halidir. Burada, D matrisinin köşegen olmayan elemanları iki çözüm arasındaki çeşitliliği (diversity) ölçerken köşegen elemanlar ise i-inci çözümün doğruluğunu (quality) ölçmektedir. D matrisinin normalleştirilme adımı, [3] çalışmasında da belirtildiği gibi, tüm

$$\widetilde{D}_{ij} = \frac{D_{ij}}{N}, \tag{4.3}$$

şeklinde gösterilebilir. Burada, N, topluluktaki kümeleme çözümlerininin sayısını temsil etmektedir ve

$$\widetilde{D}_{ij,i\neq j} = \frac{1}{2}\left(\frac{D_{ij}}{D_{ii}} + \frac{D_{ij}}{D_{jj}}\right). \tag{4.4}$$

Denklem (4.1) optimizasyon probleminin, $x^T Bx = 1$, B pozitif tanımlı matrisin eksik olduğu seyrek genelleştirilmiş özdeğer probleminin özel bir durumu olduğu açıktır. Topluluk seçimi problemini seyrekleştirmek için, denklem (4.1) optimizasyon problemine ($x^T x = 1$) kısıtı eklenir, burada B birim matrisi olarak alınır. ρ cezalandırma parametresi olmak üzere Lagrange Gevşemesini kullanarak denklem (4.1) optimizasyon problemini gevşetmek çok önemlidir. Bu nedenle, $\sum_i x_i = k$ kısıtı $\|x\|_0 = k$ olarak yazılabilir ve bu, topluluktan seçilen alt kümenin önem derecesine karşılık gelmektedir. Yani, x ikili (binary) vektörünü gerçel (real) vektöre gevşetmek için seyreklik kullanılabilir. Böylece, denklem (4.1) problemi aşağıdaki

$$\begin{aligned} \max_{x\in\mathbb{R}^n}\ & x^T\widetilde{D}x - \rho\|x\|_0 \\ & x^T x = 1\ , \end{aligned} \tag{4.5}$$

kısıtlı optimizasyon problemi olarak düzenlenebilir. $\widetilde{D}$'ın simetrik ve negatif olmayan bir matris sınıfından olduğu denklem (4.5) optimizasyon problemi ele alınsın. O halde denklem (4.5), konkav olmayan amaç fonksiyonun konveks kısıt kümesi üzerinden maksimize edilmesidir. $A := \{x : x^T x = 1\}$ NP-zordur. $x^T x = 1$ kısıtı çözüm uzayını gereğinden fazla kısıtladığından, eşitlik kısıtı $x^T x \leq 1$ olarak gevşetilebilir. Böylece denklem (4.5) optimizasyon problemi

$$\begin{aligned} \max_{x\in\mathbb{R}^n}\ & x^T\widetilde{D}x - \rho\|x\|_0 \\ & x^T x \leq 1\ , \end{aligned} \tag{4.6}$$

şekline dönüşür.

Denklem (4.6) optimizasyon problemi aşağıdaki şekilde standart forma getirilebilir:

$$\begin{aligned} \min_{x\in\mathbb{R}^n}\ & -x^T\widetilde{D}x + \rho\|x\|_0 \\ & x^T x \leq 1\ . \end{aligned} \tag{4.7}$$

$\mathbb{S}^n$ pozitif yarı tanımlı matrislerden oluşan bir küme ve $\tau \in \mathbb{R}$ olsun. O zaman, eğer $\tau \geq 0$ ise $\widetilde{D} \in \mathbb{S}^n$ olduğundan $\widetilde{D} + \tau I_n \in \mathbb{S}^n$ olduğu söylenebilir.

Eğer $\widetilde{D}$ belirsizse $\tau \geq -\lambda_{min}(\widetilde{D})$ öğesi seçilerek $\widetilde{D} + \tau I_n \in \mathbb{S}^n$ olduğundan tekrar emin

olunabilir. Böylece, eğer $\tau \geq max(0, -\lambda_{min}(\widetilde{D}))$ ise herhangi $\widetilde{D} \in \mathbb{S}^n$ için $\widetilde{D} + \tau I_n \in \mathbb{S}^n$'nin doğru olduğu söylenebilir [97]. Bu durumda, denklem (4.7) optimizasyon problemi

$$\begin{aligned} \min_x \left\{ \tau \|x\|_2^2 - \left[x^T \left(\widetilde{D} + \tau I \right) x - \rho \|x\|_0 \right] \right\} \\ x^T x \leq 1 \, , \end{aligned} \tag{4.8}$$

şeklinde ifade edilebilir. Bu çalışmada, denklem (4.8) optimizasyon probleminin ikinci terimi için Bölüm 2.2.4, Varsayım 1 tarafından sağlandığı iyi bilinen, konveks olmayan yaklaşımlardan biri olan,

$$z(x;\theta) = 1 - e^{-\theta |x|} \tag{4.9}$$

fonksiyonu seçilmiştir [41]. $z(x;\theta)$ konkav bir fonksiyondur ve

$$z(x;\theta) = \theta \mid x \mid - \left(\theta \mid x \mid - (1 - e^{-\theta |x|}) \right) \tag{4.10}$$

şeklinde bir KF fonksiyonu olarak ifade edilebilir. Böylece, denklem (4.8) optimizasyon problemi

$$\begin{aligned} \min_x \left\{ \tau \|x\|_2^2 - \left[x^T \left(\widetilde{D} + \tau I \right) x - \rho \sum_{i=1}^n \left[\theta \mid x_i \mid - \left(\theta \mid x_i \mid - (1 - e^{-\theta |x_i|}) \right) \right] \right] \right\} \\ x^T x \leq 1 \, , \end{aligned} \tag{4.11}$$

şeklinde formüle edilebilir. Burada bulunan mutlak değer fonksiyonu türevlenemediği için yeni bir y değişkeni tanımlanarak denklem (4.11) optimizasyon problemi, $\Omega = \{(x,y) : (x,y) \in C, \mid x_i \mid \leq y_i\}$ olmak üzere,

$$\begin{aligned} ECS\_KFP: \quad \min_{\Omega} \left\{ \tau \|x\|_2^2 - \left[x^T \left(\widetilde{D} + \tau I \right) x - \rho \sum_{i=1}^n \left[\theta y_i - \left(\theta y_i - (1 - e^{-\theta y_i}) \right) \right] \right] \right\} \\ x^T x \leq 1 \, , \end{aligned} \tag{4.12}$$

formuna dönüşecektir.

KFP modeli, ECS_KFP, çözülürken, çözüm vektörü ikili çözüme dönüştürüleceğinden x için bir eşik değeri seçilmiştir. Bu nedenle, Algoritma 2'deki ikili çözüm deneysel

olarak

$$x_i = \begin{cases} 1 & \text{eğer } x_i \geq \delta, \\ 0 & \text{eğer } x_i \leq \delta, \end{cases} \tag{4.13}$$

olacak şekilde $\delta = \frac{1}{10}$ eşik değeri ile belirlenmiştir. Bu seçim, topluluk altkümesi boyutunda bir azalmaya yol açacak seyrek bir çözüm sunmaktadır.

Algoritma 2, denklem 4.12 optimizasyon probleminin `Python`'ın `CVXPY` kütüphanesinde disiplinli konveks konkav programlama (`DKKP`) paket uygulaması ile çözülmüştür. `DKKP`, disiplinli konveks programlama (DKP) ve konveks-konkav programlama (KKP) fikirlerinin birleştirilmesiyle oluşturulan bir optimizasyon yöntemidir [29]. DKP, problemler oluşturulurken izlenmesi gereken bir dizi kuralı gerektirirken, KKP konveks olmayan problemleri çözmek için düzenli bir sezgisel yöntemdir. KKP, konveks olmayan terimleri, orijinal KF probleminin kısıtlı bir formu olan bir konveks üst sınır ile değiştiren başka bir tür sezgisel algoritmadır. Bu, minimizasyon probleminin, güncel noktanın etrafındaki bir üst sınır çözülerek yaklaştırıldığı büyütme-azaltma yönteminin bir örneğidir. KKP, KPA'nın bir versiyonu olarak düşünülebilir . `DKKP`, KF programlamanın basit bir standart formudur. `CVXPY`, konveks optimizasyon için özelleşmiş bir kütüphanedir. `DKKP` yöntemi öncelikle problemin `DKKP` kurallarını karşıladığını doğrular ve sonra konveks programı, her fonksiyonu grafik uygulamasıyla değiştirerek bir koni problemine dönüştürür. Ardından, rastgele bir genel konik çözücü ile konik problemi çözer.

Algorithm 2 *ECS_KFP*

Girdi : Veriseti, Temel öğreniciler, N ▷ N: topluluk boyutu
Parametreler : τ, ρ, θ ▷ δ: eşik değeri
Çıktı : NKB değerleri

1: E topluluk kütüphanesini oluştur
2: 4.4 ile verilen $\widetilde{D}$ matrisini hesapla
3: Aşağıdaki denklem (4.12) ile verilen modeli çöz:

$$\min_x \left\{ \tau \|x\|_2^2 - \left[x^T \left(\widetilde{D} + \tau I \right) x - \rho \sum_{i=1}^{n} \left[\theta y_i - \left(\theta y_i - (1 - e^{-\theta y_i}) \right) \right] \right] \right\}$$
$$x^T x \leq 1$$

4: Her bir i-inci kümeleme çözümü için $x_i \geq \delta$ olacak şekilde tüm x_i'leri bulun
5: 4'deki alt kümenin kümeleme kararlarını HGBA kullanarak birleştirin

4.1.3 Adım 3 : Toplama (Birleştirme)

Konsensüs fonksiyonu, üretim adımında oluşturulan tüm bölümlerin birleştirilerek yeni bir bölüm oluşturulduğu bir işlemdir. Literatürde, Strehl ve Gosh [86], bölümler

arasındaki yakınlığı veya kümeleme çözümlerinin birbirine olan çeşitliliğini ölçen NKB fonksiyonunu maksimize eden konsensüs fonksiyonunu önermiştir. Budama adımından sonra, yeni alt-topluluk çözümlerinin birleştirilerek nihai bir kümeleme yapılması gerekmektedir. Bu çalışmada, kümeleme problemleri için literatürde en yaygın tercih edilen yöntemlerden biri olan HGBA [86] kullanılmıştır. HGBA yaklaşımı hakkında daha geniş bilgi, MATLAB kodu ile birlikte literatürde bulunabilir [98].

4.1.4 ECS_KFP için Optimalite İncelemesi

Bu bölümde, denklem (4.12) optimizasyon probleminin Bölüm 2.2.4'de verilen gerekli varsayım ve optimalite koşullarını sağladığı gösterilmiştir. Dolayısıyla, bu yeni formülasyon,

$$f(x) = \tau \|x\|_2^2 - x^T \left(\widetilde{D} + \tau I\right) x$$

ve

$$z(y;\theta) = \left[\theta y_i - \left(\theta y_i - (1 - e^{-\theta y_i})\right)\right]$$

olmak üzere, denklem (2.23) optimizasyon probleminin bir uygulanmasına karşılık gelmektedir. Burada, $\phi(y;\theta)$ ve $\omega(y;\theta)$ $\mathbb{R}$'de sonlu konveks fonksiyonlar olacak şekilde

$$\phi(y_i;\theta) := \theta y_i$$

ve

$$\omega(y_i;\theta) := \theta y_i - (1 - e^{-\theta y_i})$$

olduğu Varsayım 1, part (c)'de verilmiştir. Böylece, $z(y;\theta)$ bir KF fonksiyonu olacaktır. $\mathscr{M}$ ve $\mathscr{M}_\theta$, sırasıyla, denklem (2.21) ve (2.22) problemlerinin çözüm kümeleri olsun. Eğer $\theta_k \to \infty$ olacak şekilde (θ_k) negatif olmayan sayıların bir dizisi ise ve (x^k), her k ve $x^k \to x^*$ için $x^k \in \mathscr{M}_{\theta_k}$ olacak şekilde bir dizi ise bu durumda $x^* \in \mathscr{M}$ olacağı Teorem 2.12'de ispatlanmıştır. $-y_i \le x_i \le y_i$ eşitsizlik kısıtı ve $x^T x \le 1$ uygun bölgesi kompakt olduğundan, herhangi bir $\epsilon > 0$ için $\mathscr{M}_\theta \subset \mathscr{M} + B(0,\epsilon)$, $\forall \theta \ge \theta(\epsilon)$ olacak şekilde bir $\theta(\epsilon) > 0$ sayısı vardır (bknz. Teorem 2.12, part (b)). Teorem 2.12'in sonucunun, konveks olmayan $z(x;\theta)$ yaklaşımının herhangi bir negatif olmayan θ parametresi için denklem (2.22) optimizasyon probleminin bir çözüme yakınsadığının garantisini verdiğine dikkat edilmesi gerekmektedir. Dahası, Teorem 2.12'in (b) kısmında belirtildiği üzere, denklem (2.22) optimizasyon probleminin herhangi bir optimal çözümü, denklem (2.21) optimizasyon probleminin optimal çözümünün bir ϵ komşuluğunda bulunmaktadır.

Konveks olmayan z yaklaşımının $[0, \infty)$ aralığında konkav olduğu, f'nin $\mathscr{C}$ üzerinde alttan sınırlı konkav olduğu açıktır ve denklem (4.12) optimizasyon probleminin uygun (feasible) bölgesi, kısmen bir polihedral özelliktedir çünkü $-y_i \leq x_i \leq y_i$ kısıtı konveks bölgenin polihedral doğasına kısmen katkıda bulunmaktadır. Bununla birlikte, $x^T x \leq 1$ disk koşulunun polihedral olmaması nedeniyle, genel olarak uygun kümenin polihedral olduğu söylenememektedir, ancak uygun kümenin sınır boyunca polihedral olduğu ve enine kesişme durumunda bile "stabil" olduğu söylenebilmektedir. Ayrıca, $x^T x \leq 1$ kısıtı, birinci dereceden değil, ikinci dereceden bir koni kesiti olduğu için temel bir kısıttır ve genellikle daha fazla karmaşıklığa yol açmamaktadır. Bu durumda, Sonuç 2.3'e göre, $\overline{\mathscr{M}_\theta} = \{x : \exists y \in \mathbb{R}^n \text{ öyle ki } (x, y) \in \vartheta\}$ kümesi var olacaktır, burada ϑ, Ω'nın bir köşe kümesidir. O zaman, $\emptyset \neq \overline{\mathscr{M}_\theta}$, $\forall \theta > 0$ ve her $\theta \geq \theta_0$ için $\overline{\mathscr{M}_\theta} < \mathscr{M}$ olduğu bir $\theta_0 > 0$ bulunmaktadır. Denklem (4.12) optimizasyon problemi ele alınsın. Buradaki terimleri, Bölüm 2.2.4, Önerme 2.4'deki gibi temsil etmek için aşağıdaki tanımlamalar yapılmaktadır:

$$G(x) := \tau \|x\|_2^2 + \rho \sum_{i=1}^{n} \theta y_i,$$

$$H(x) := \Big[x^T (\widetilde{D} + \tau I) x + \rho \sum_{i=1}^{n} (\theta y_i - (1 - e^{-\theta y_i})) \Big].$$

Bu da bizi Teorem 2.13'ye götürmektedir. Denklem (2.25) ve denklem (2.28) tarafından verilen gerekli optimallik koşulları sağlandığından, Teorem 2.13 geçerlidir.

4.1.5 Nümerik Deneyler

Bu çalışmada veri setleri UCI Machine Learning Repository'den [99] seçilmiştir. Tablo 4.1'de, sırasıyla, veri seti özellikleri, öznitelik özellikleri, örnek sayısı, öznitelik sayısı ve küme sayısı bilgileri verilmiştir.

Tablo 4.1 Veri seti bilgileri

Veri	Veri Seti Özellikleri	Öznitelik Özellikleri	Örnek Sayısı	Öznitelik Sayısı	Küme Sayısı
Wine	Çok Değişkenli	Tamsayı, Gerçek	178	13	3
Synthetic Control	Zaman-Serisi	Gerçek	600	N/A	6
Segmen-tation	Çok Değişkenli	Gerçek	2310	19	7
Ecoli	Çok Değişkenli	Gerçek	336	8	8
Glass	Çok Değişkenli	Gerçek	214	10	6
Zoo	Çok Değişkenli	Kategorik, Tamsayı, Gerçek	101	17	7

Topluluk kütüphanesinin oluşturulması üç adımdan oluşmaktadır. Bu üç adımdan her birinde, elli kümeleme çözümü oluşturulmuştur. Böylece, topluluk kütüphanesinde 150 farklı kümeleme çözümü elde edilmiştir. Bu 150 kümeleme çözümü arasından, alt-topluluk için en iyi adaylar Algoritma 2 kullanılarak seçilmiştir.

Algoritma 2'in performans sonuçları, ele alınan problemin denetimsiz makine öğrenme problemi olması nedeniyle NKB kullanılarak değerlendirilmektedir, çünkü kümeler içindeki karşılıklı bilginin değeri, kümeleme kararının kalitesini yansıtmaktadır. Diğer bir deyişle, NKB değerleri ne kadar 1'e yakınsa, kümeleme sonuçlarının kalitesi o kadar iyi olacaktır. Sonuçlar, çalışma [5]'deki açgözlü (greedy) tabanlı yaklaşımlarla karşılaştırılmıştır, bunlar Ortak Kriter (Joint Criteria) ve Kümele ve Seç (Cluster and Select) yöntemleri olarak adlandırılmaktadır. Ortak Kriter Yöntemi, kalite ve çeşitlilik terimlerini birleştirerek ortak kriter fonksiyonunu oluştururken, Kümele ve Seç Yöntemi, topluluktaki her kümeleme çözümünü bir model olarak dikkate almakta ve bu kümeleme çözümlerinin ilişkisini analiz etmektedir [5]. Ayrıca sonuçlar, benzer bir çalışma olan [29] çalışması ile kıyaslanmıştır. Bu çalışmada, sıfır-normu öğrenci t dağılımının negatif log olasılığı(negative-log likelihood of a Student t-distribution) ile yaklaştırılmıştır. ECS_KFP modeli için ρ hiper-parametresi, en iyi NKB sonuçlarını veren $\rho = [10^{-1}, 10^{-2}, 10^{-3}, 1, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}, 10^{6}]$ değerleri arasından seçilmiştir.

Tablo 4.2 NKB değerleri

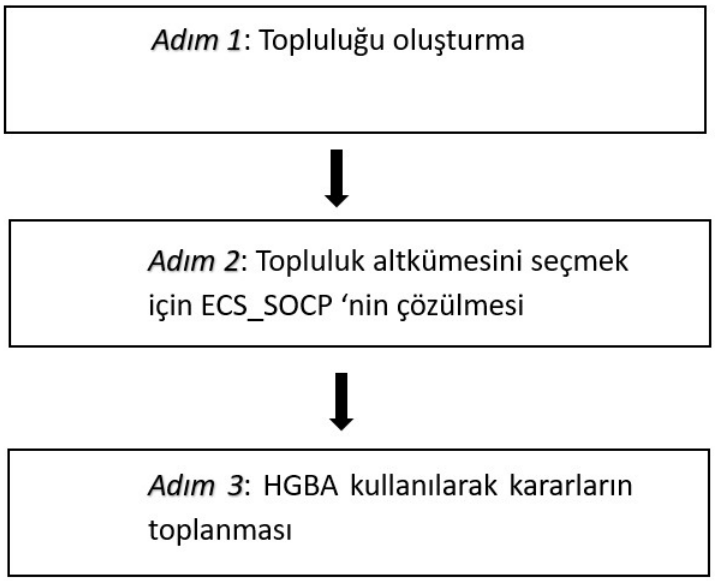
Veri	Joint Criteria	Cluster &Select	PrunedOpt	ECS_KFP	En iyi Boyut
Wine	0,716708	0,566190	0.763000	**0,781696**	52
Synthetic Control	0,698854	0,761608	0.777000	**0,832222**	55
Segmen-tation	0,396007	0,197260	**0.484000**	0,461336	55
Ecoli	**0,634155**	0,494483	-	0,578387	55
Glass	0,668443	0,660236	**0.757000**	0,714068	55
Zoo	0,713078	0,665306	**0.777000**	0,735905	54
Mean	0,625611	0,514162	0.593000	**0,650739**	**54**

Tablo 4.2'nin son sütunu topluluk seçiminin kardinalitesini göstermektedir. Diğer bir deyişle, seyrek çözümün kardinalitesine atıfta bulunmaktadır. Daha iyi sonuçlar kalın yazı tipi ile ifade edilmiştir. Neredeyse her veri kümesi için konveks konkav programlamanın, diğer greedy tabanlı yöntemlere göre NKB değerleri açısından daha iyi performans gösterdiği açıktır. Sıfır normuna yaklaşım foksiyonu farklı olan PrunedOpt ile ise benzer performans göstermiştir. "Wine", "Synthetic Control" veri setleri için ECS_KFP performansı domine ederken "Segmantation", "Glass" ve "Zoo" veri setlerinde PrunedOpt ve "Ecoli" veri setinde Ortak Kriter metodu daha başarılı bir performans göstermiş olduğu görülmektedir. ECS_KFP ve PrunedOpt arasındaki yaklaşım fonskiyonu farkı veri setlerine bağlı seçim ile belirlenebilir.

4.2 İkinci Derecen Konik Programlama ile Topluluk Kümeleme Seçilimi

Bu bölümde, Şekil 4.2'de verilen akış şemasında gösterildiği gibi, topluluğun oluşturulması, önerilen modelle topluluk budaması ve son olarak kümelerin toplanması (aggregation) açıklanacaktır.

Adım 4.2.2 önerilen denklem (4.34) optimizasyon problemi, ESC_SOCP olarak adlandırılmıştır. Bölüm (4.1)' da anlatıldığı gibi topluluk kümesi seçimini oluşturmak için üç temel adım kullanılmıştır.



Şekil 4.2 Önerilen modelin akış şeması [95].

Kısaca, ilk adım, bir temel kümeleme algoritması olarak k-ortalama kullanılarak topluluğun oluşturulmasını içermektedir. İkinci adım, topluluğun bir alt kümesini elde etmek için ESC_SOCP'yi çözerek oluşturulmaktadır. Son olarak, son adım ise, Adım 4.2.1'de oluşturulan topluluğun kümeleme kararları Hiper Grafik Bölümleme Algoritması (HGBA) [86] kullanarak toplanmaktadır (aggregate).

4.2.1 Adım 1 : Topluluğu Oluşturma

Adım (4.1.1)'de açıklandığı üzere üretim adımı, farklı kümeleme çözümleri elde etmeye yönelik bir süreç içermektedir. Aynı kümeleme algoritmasının çeşitli parametreleriyle oluşturulan farklı kümeleme algoritmaları veya farklı kümeleme algoritmaları, çeşitli nesne temsilleri, veri toplamanın farklı alt kümeleri ve nesnelerin farklı alanlara izdüşümleri kullanılmasıyla bir dizi temel küme çözümü oluşturulur [96]. Topluluk kütüphanesinin üretim adımı bu çalışmada aşağıdaki üç adımdan oluşmaktadır:

i. Rastgele Başlatma

ii. Rastgele Öznitelik Seçimi

iii. Rastgele Lineer Projeksiyon

Bu süreçte, k-ortalama algoritması, adım (i)'ye karşılık gelen rastgele seçilen parametre değerleriyle çalıştırılarak çeşitli kümeleme çözümleri elde etmek için bir

temel öğrenici olarak kullanılmıştır. Daha sonra (adım (ii)), her biri rastgele seçilen bir öznitelik (feature) için elde edilen kümeleme çözümleri topluluğun kitaplığına eklenmiştir. Son olarak, adım (iii), verilerin geometrik yapısını koruyarak boyutunu azaltmak amacıyla, orijinal özniteliklerin daha düşük boyutlara yansıtılması için rastgele bir izdüşüm matrisi kullanılmıştır.

4.2.2 Adım 2 : ESC_SOCP ile Topluluk Budama

Adım (4.1.2)'de açıklanan aday modeller arasındaki ilişkileri dikkate alan önceden belirlenmiş D matrisi ele alınsın:

$$D_{ij} = \begin{cases} D_{ii} = SNMI(C, C_i)\ ,\ (i = 1, ..., n) \\ D_{ij} = 1 - NMI(C_i, C_j)\ ,\ (j = 1, ..., n) \end{cases} \tag{4.14}$$

Burada, D matrisinin köşegen olmayan elemanları iki çözüm arasındaki çeşitliliği (diversity) ölçerken köşegen elemanlar ise i-inci çözümün doğruluğunu (quality) ölçmektedir. D matrisi, [3] çalışmasında da belirtildiği gibi, tüm elemanları aynı düzeye taşımak adına

$$\widetilde{D}_{ij} = \frac{D_{ij}}{N}, \tag{4.15}$$

şeklinde normalleştirilebilir. Burada, N, topluluktaki kümeleme çözümlerininin sayısını temsil etmektedir ve

$$\widetilde{D}_{ij, i \neq j} = \frac{1}{2}\left(\frac{D_{ij}}{D_{ii}} + \frac{D_{ij}}{D_{jj}}\right). \tag{4.16}$$

Doğruluk ve çeşitlilik değiş tokuşunu optimize eden,

$$\begin{aligned} \text{maximize} \quad & x^T \widetilde{D} x \\ \text{subject to} \quad & \sum_i x_i = k, \\ & x_i \in \{0, 1\} \end{aligned} \tag{4.17}$$

modeli ele alınsın. Bu ikinci dereceden tam sayılı programlama problemi, NP-zor olan standart bir 0-1 optimizasyon problemidir. Gösterilmiştir ki bu formülasyon k boyutu ile Maksimum Kesim (MK-k) problemine oldukça yakındır [3]. Bu method, bir grafiğin köşelerini alt grupların her birinde bir uç noktaya sahip olan kenarların ağırlıklarının toplamı maksimum olacak şekilde iki alt gruba ayırır. MK-k problemi aşağıdaki şekilde formüle edilebilir:

$$\begin{aligned} \max_{y} \quad & \frac{1}{2}\sum_{i<j} w_{ij}(1-y_i y_j) \\ \text{s. t.} \quad & \sum_{i} y_i = N_v - 2k, \\ & y_i \in \{-1,1\}. \end{aligned} \tag{4.18}$$

Burada, N_v grafikteki köşelerin sayısını, $w_{i,j}$ ise i ve j arasındaki köşe ağırlığını temsil etmektedir. İDKP rahatlatmasını yapabilme olanağını sağlamak için grup budama formülasyonunun MK-k çerçevesine uyacak şekilde dönüştürülebilmesi, budama problemi için iyi bir çözüm elde etme fırsatı sunacaktır.

Yukarıdaki MK-k modeli denklem (4.18), W kenar ağırlık matrisini ($w_{i,i} = 0$) temsil edecek şekilde aşağıdaki gibi ifade edilebilir:

$$\begin{aligned} \min_{y} \quad & y^T W y \\ \text{s. t.} \quad & \sum_{i} y_i = N_v - 2k, \\ & y_i \in \{-1,1\}. \end{aligned} \tag{4.19}$$

Denklem (4.17) ile denklem (4.19) optimizasyon problemleri arasındaki temel fark $x_i \in \{0,1\}$ ile $y_i \in \{-1,1\}$ olduğundan, değişken dönüşümü yapılması gerekmektedir. $v_i = \{-1,1\}$ olmak üzere

$$x_i = \frac{v_i + 1}{2} \tag{4.20}$$

olarak alınırsa denklem (4.17) optimizasyon problemindeki yeni amaç fonksiyonu e'nin tüm birlerin kolon vektörü olmasıyla beraber

$$\frac{1}{4}(v+e)^T D(v+e) \tag{4.21}$$

formunu alır. $\sum_{i=1}^{n} x_i = k$ kardinalite kısıtı, I birim matris olmak üzere

$$x^T I x = k \tag{4.22}$$

şeklinde kuadratik form halinde yazılabilir. Değişken dönüşümünden sonra ise,

$$(v+e)^T I(v+e) = 4k \tag{4.23}$$

elde edilir.
Hem amaç fonksiyonu hem de kardinalite kısıtı dönüşümü sağlandığından, problemi tekrar kuadratik forma sokmak için, $v = (v_1, v_2, v_3, \cdots, v_n)$, $v_0 = 1$ olmak üzere $v = (v_0, v_1, v_2, v_3, \cdots, v_n)$ rahatlatılması yapılabilir. Daha sonra, oluşturulan yeni H matrisi

$$H_{(n+1)\times(n+1)} = \begin{pmatrix} e^T \widetilde{D} e & e^T \widetilde{D} \\ \widetilde{D} e & \widetilde{D} \end{pmatrix} \tag{4.24}$$

ile, amaç fonksiyonu $v^T H v$ formuna ulaşır. Benzer şekilde G matrisi

$$G_{(n+1)\times(n+1)} = \begin{pmatrix} n & e^T \\ e & I \end{pmatrix} \tag{4.25}$$

oluşturularak, kardinalite kısıtı da $v^T G v = 4k$ formunu alır. Böylece, $-H = \widetilde{H}$ olmak üzere problem aşağıdaki şekilde ifade edilebilir:

$$\begin{aligned} \underset{v}{\text{maximize}} \quad & v^T \widetilde{H} v \\ \text{subject to} \quad & v^T G v = 4k, \\ & v_0 = 1, \\ & v_i \in \{-1, 1\},\ \forall i \neq 0. \end{aligned} \tag{4.26}$$

Denklem 4.26 optimizasyon problemi aşağıdaki konveks olmayan kuadratik formda ifade edilebilir:

$$\begin{aligned} \text{minimize} \quad & \theta \\ \text{subject to} \quad & v^T \widetilde{H} v \leq \theta, \\ & v^T G v = 4k, \\ & v_0 = 1, \\ & v_i \in \{-1, 1\},\ \forall i \neq 0. \end{aligned} \tag{4.27}$$

Dikkat edilsin ki $v_0 = 1$ kısıtı, problem değişmeden $v_i \in \{-1, 1\}$ formuna rahatlatılabilir çünkü $-v$'nin geri kalan kısıtlamalar için uygun (feasible) olması ancak ve ancak v uygun (feasible) olduğunda mümkündür. Böylece, denklem (4.27) optimizasyon problemi

$$\begin{aligned} \text{minimize} \quad & \theta \\ \text{subject to} \quad & v^T \widetilde{H} v \leq \theta, \\ & v^T G v = 4k, \\ & v_i \in \{-1, 1\},\ \forall i \neq 0. \end{aligned} \tag{4.28}$$

formunu alacaktır. Aşağıdaki, (v, V) tarafından sağlanan eşitsizlik kısıtı ele alınsın

$$v^T \widetilde{H} v \leq \theta \tag{4.29}$$

ve $\widetilde{H}$ spektral ayrışımı

$$\widetilde{H} = \sum_{j=1}^{n+1} \lambda_j q_j q_j^T \tag{4.30}$$

kullanılsın. Genelliği kaybetmeden,

$$\lambda_1 \geq \cdots \lambda_l \geq 0 > \lambda_{l+1} \geq \cdots \geq \lambda_{n+1},$$

olduğu varsayılsın. Burada λ_j, öz değerleri ve q_j, bu öz değerlere karşılık gelen birim öz vektörleri temsil etmektedir. $\widetilde{H}^+ = \sum_{j=1}^{l} \lambda_j q_j q_j^T$ olsun. $\widetilde{H}^+$ ve $q_j q_j^T$, $j = l+1, \ldots, n$ seçilerek, $V = v^T v$ olmak üzere

$$v^T \widetilde{H}^+ v - \widetilde{H}^+ V \leq \theta \tag{4.31}$$

$$v^T q_j q_j^T v - q_j q_j^T V \leq \theta \tag{4.32}$$

elde edilir. Denklem (4.29) ve denklem (4.31) kullanılarak aşağıdaki şekilde yeni bir eşitsizlik üretilebilir:

$$v^T \widetilde{H}^+ v + \sum_{j=l+1}^{n+1} \lambda_j q_j q_j^T \leq \theta. \tag{4.33}$$

Bu durumda, (v, V) aynı zamanda denklem (4.33)'ü (denklem (4.29) ve denklem (4.31'i sağlamak üzere) sağlamalıdır. Çalışma [32]'deki motivasyon göz önüne alındığında, denklem (4.28) optimizasyon probleminde ki MK-k modeli parametre azaltma tekniği kullanılarak aşağıdaki şekilde İDKP rahatlamasına dönüştürülebilir:

$$\begin{aligned} ECS\_SOCP: \quad & \text{minimize} \quad \theta \\ & \text{subject to} \quad v^T \widetilde{H}^+ v + \sum_{j=l+1}^{n+1} \lambda_j z_j \leq \theta, \\ & \qquad\qquad v^T G v = 4k + z_j, \ (j = 1, \cdots, n+1), \\ & \qquad\qquad z_j \leq \sqrt{n}, \ (j = l+1, \cdots, n+1). \end{aligned} \tag{4.34}$$

Burada, $z_j = q_j^T V q_j$ için sınır V_{ij}'nin ya $+1$ ya da -1 olmasından ve $||q_j|| = 1$ olmasından dolayı oluşmaktadır. İDKP modeli denklem (4.34) (ECS_SOCP)

çözülürken, çözüm vektörü ikili çözüme dönüştürüleceğinden v için bir eşik değeri seçilmiştir. Bu nedenle, Algoritma 3'deki ikili çözüm deneysel olarak

$$v_i = \begin{cases} 1 & \text{eğer } v_i \geq \delta \\ 0 & \text{eğer } v_i \leq \delta, \end{cases} \tag{4.35}$$

olacak şekilde $\delta = \frac{1}{10}$ eşik değeri ile belirlenmiştir. Bu seçim, bölüm (4.1)'de açıklandığı gibi, topluluk altkümesi boyutunda bir azalmaya yol açacak seyrek bir çözüm sunmaktadır. Topluluk boyutu veya k, çözümün kalitesini belirlemektedir. k'nin, yöntemin doğruluğunu etkileyen modelin spesifik parametresi olduğu kabul edilmelidir. Yüksek değerli k seçimi, anlamlı adayların eksikliğini gösterebilir. Bu nedenle, anlamlı aday olmama ihtimaline rağmen, iyi kalibre edilmiş bir k değeri almak önem teşkil etmektedir. İNM, yani Bariyer Yöntemi, lineer veya lineer olmayan konveks optimizasyon problemlerini çözmek için kullanılan özel bir algoritma sınıfıdır. Eşitsizlik kısıtlamalarının ihlal edilmesini önlemek için amaç fonksiyonu bir bariyer terimi ile büyütülür ve böylece bu, kısıtlanmamış optimal değerin uygun uzayda kalmasına neden olur. Bu çalışmada, Matlab fonksiyonu fmincon ve İNM denklem (4.34) optimizasyon modelini çözmek için kullanılmıştır. Kısıtlamaları tanımlamak için [98]'te geliştirilen CVX kitaplığı kullanılmıştır. CVX, kullanıcıların Matlab programlama dilini kullanarak konveks optimizasyon için modeller oluşturmasını sağlayan bir yazılım aracıdır. Temelde, MATLAB'ı bir optimizasyon modelleme diline dönüştürmektedir. Bu şekilde, MATLAB'ın standart ifade sözdizimi kullanılarak kısıtlar ve amaçların belirlenmesi mümkündür. MATLAB'ın CVX kütüphanesinde uygulanan denklem 4.34 optimizasyon probleminin algoritması, Algoritma 3 adı ile aşağıda verilmektedir:

4.2.3 Adım 3 : Toplama (Birleştirme)

Hatırlatmak gerekirse, konsensüs fonksiyonu, üretim adımında oluşturulan tüm bölümlerin birleştirilerek yeni bir bölüm oluşturulduğu bir işlemdir. İki kategoriye ayrılmakla beraber, bunlar, veri koleksiyonundaki nesnelerin ortaya çıkması ve medyan bölümleridir. İlk kategori sezgisel bir yöntem olup, ikinci kategori ise yöntemlerin tamamı, mesafelerin toplamını en aza indirmeye dayalı optimizasyon yöntemlerini içermektedir. Literatürde, Strehl ve Gosh [86], bölümler arasındaki yakınlığı veya kümeleme çözümlerinin birbirine olan çeşitliliğini ölçen NKB fonksiyonunu maksimize eden konsensüs fonksiyonunu önermiştir. Budama adımından sonra, yeni alt-topluluk çözümlerinin birleştirilerek nihai bir kümeleme yapılması gerekmektedir. Bu çalışmada, literatürde en sık önerilen kümeleme

Algorithm 3 *ECS_SOCP*

Girdi : Veriseti, Temel öğreniciler, n ▷ n: topluluk boyutu
Parametreler : k, δ ▷ k: budama boyutu
▷ δ: eşik değeri

Çıktı : NKB değerleri

1: E topluluk kütüphanesini oluştur
2: Aşağıdaki denklem (4.34) ile verilen modeli çöz:

$$\begin{aligned} \text{minimize} \quad & \theta \\ \text{subject to} \quad & v^T \widetilde{H}^+ v + \sum_{j=l+1}^{n+1} \lambda_j z_j \leq \theta, \\ & v^T G v = 4k + z_j, \ j = 1, \cdots, n+1 \\ & z_j \leq \sqrt{n}, \ j = l+1, \cdots, n+1 \end{aligned}$$

3: Her bir i-inci kümeleme çözümü için $v_i \geq \delta$ olacak şekilde tüm v_i'leri bul
4: 3'deki alt kümenin kümeleme kararlarını HGBA kullanarak birleştir

problemleri yaklaşımlardan biri olan HGBA [86] kullanılmıştır.

4.2.4 Nümerik Deneyler

Bu çalışmada veri setleri UCI Machine Learning Repository'den [99] seçilmiştir. Tablo 4.3'de, sırasıyla, veri seti özellikleri, öznitelik özellikleri, örnek sayısı, öznitelik sayısı ve küme sayısı gibi açıklamalar verilmiştir.

Adım 4.2.3'te açıklandığı gibi, topluluk kütüphanesinin oluşturulması üç adımdan oluşmaktadır. Bu üç adımdan her birinde, elli kümeleme çözümü oluşturulmuştur. Böylece, topluluk kütüphanesinde 150 farklı kümeleme çözümü elde edilmiştir. Bu 150 kümeleme çözümü arasından, alt-topluluk için en iyi adaylar Algoritma 3 kullanılarak seçilmiştir.

Tablo 4.3 Veri seti bilgileri

Veri	Veri Seti Özellikleri	Öznitelik Özellikleri	Örnek Sayısı	Öznitelik Sayısı	Küme Sayısı
Non-Verbal Tourist	Çok Değişkenli	Tamsayı, Gerçek	73	22	6
E-shop Clothing	Çok Değişkenli Dizisel	Tamsayı, Gerçek	165474	14	5
Wine	Çok Değişkenli	Tamsayı, Gerçek	178	13	3
Synthetic Control	Zaman-Serisi	Gerçek	600	N/A	6
Segmen-tation	Çok Değişkenli	Gerçek	2310	19	7
Glass	Çok Değişkenli	Gerçek	214	10	6
Bupa	Çok Değişkenli	Kategorik,Tamsayı, Gerçek	345	7	2
Frog_ genus	Çok Değişkenli	Gerçek	7195	22	8
Iris	Çok Değişkenli	Gerçek	150	4	3
Diabets	Çok Değişkenli	Zaman-Serisi	768	20	2
Wdbc	Çok Değişkenli	Gerçek	569	32	2
Wpbc	Çok Değişkenli	Gerçek	198	34	2
Iono-sphere	Çok Değişkenli	Kategorik, Tamsayı, Gerçek	351	34	2
Zoo	Çok Değişkenli	Kategorik, Tamsayı, Gerçek	101	17	7

Tablo 4.4'de, ECS_SOCP topluluk budama modelinin performansının karşılaştırıldığı ortak kriter (joint criteria), kümele ve seç (cluster and select), EAM ve PrunedOPT yöntemleri bulunmaktadır. Daha önce anlatıldığı gibi, Ortak Kriter Yöntemi, kalite (quality) ve çeşitlilik (diversity) bileşenlerini tek bir kriter fonksiyonunda birleştirmektedir. Kümele ve Seç yöntemi ise topluluktaki her kümeleme çözümünü bir varlık olarak kabul ederek aralarındaki ilişkiyi analiz etmektedir. EAM, orijinal veri uzayından bir eş-ilişkilendirme (co-assosation) matrisine çekirdek dönüşümü gerçekleştirmektedir ve temel bölümlerden veri toplamaktadır. Ardından, daha doğru kümeleme sonuçları elde etmek için eş-ilişkilendirme matrisinden bazı

kanıtları kaldırmaktadır. Son olarak, hem doğru hem de çeşitli modellere sahip topluluğun optimal alt kümesini seçen optimizasyon tabanlı model PrunedOPT, DKKP kullanılarak çözülmektedir. ECS_SOCP, hem doğruluk hem de çeşitliliği göz önünde bulundurarak yukarıdaki yaklaşımlar arasından sıyrılmaktadır. PrunedOPT yöntemi de bu dengeyi göz önünde bulundursa da, maksimum kesim problemi İDKP olarak modellenmesi tamsayı programlamasını sürekli ve konveks olarak çözmeyi mümkün kılmaktadır. Bu ise bize KF programlamaya göre bir avantaj sağlamaktadır. Denklem (3.8)'deki α değeri, ortak kriter algoritması için çalışma [5]'de olduğu gibi 0.5 olarak seçilmiştir. PrunedOPT modeli için ρ hiper-parametresi, en iyi NKB sonuçlarını veren $\rho = [10^{-1}, 10^{-2}, 10^{-3}, 1, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}, 10^{6}]$ değerleri arasından seçilmiştir. Ayrıca, en iyi ρ değeri her veri seti için farklılık göstermektedir. Önerilen yaklaşım ECS_SOCP'nin, diğer yöntemlerin ortalama değer sonuçlarına kıyasla eşdeğer veya daha iyi NMI değerleri elde ettiği gözlenmektedir. Ayrıca, PrunedOPT dışında, diğer yöntemlerin budama oranını bir hiper-parametre olarak gerektirmesi ve algoritmamızda bulunan en iyi boyut bilgisinin bu yöntemler için girdi olarak verilmesi, onların performansları için haksız bir kazanım sağladığı göz önünde bulundurulmalıdır.

Ele alınan problem denetimsiz makine öğrenimi problemi olduğu için, kümeler arasındaki karşılıklı bilgi değeri, kümeleme kararının kalitesini temsil etmektedir. Diğer bir deyişle, NKB değerleri 1'e yaklaştıkça, daha kaliteli kümeleme sonuçları elde edilmektedir. Modellerin performansları, Tablo 4.4'de NKB'nin ortalama değerlerini kullanarak daha iyi sonuç için kalın yazı tipiyle ve Tablo 4.5'de her sonuç için kare parantez içinde güven aralıkları sunulmuştur. Tablo 4.4'daki sonuçlara daha yakından bakıldığında, ECS_SOCP yöntemi, "Non-verbal tourist" ve "E-shop clothing" verileri ile "Glass" verileri için ticaret analizinde üstün bir yaklaşım olarak öne çıkmaktadır. Öte yandan, "Wpbc" verileri için Ortak Kriter yöntemi ve "Wine" ve "Segmentation" verileri için Kümele ve Seç yöntemi daha iyi sonuçlar vermektedir. "Bupa", "Frog_genus", "Iris" ve "Wdbc" verileri için EAM yöntemi daha etkilidir. "Synthetic Control" ve "Diabetes" verileri için ise üstün gelen seçenek PrunedOPT'dir. Makine öğreniminde evrensel olarak kabul edilen bir standart olmadığı ve sonuçların kullanılan veri setlerine bağlı olarak değişebileceği belirtilmelidir. Yine de, ortalama değere dayanarak, yaklaşımımızın Tablo 4.4'da belirtildiği üzere üstün sonuçlar verdiği görünmektedir.

Tablo 4.4 NKB değerleri

Veri	Joint Criteria	Cluster & Select	EAM	PrunedOpt	ECS_SOCP*	*En İyi Boyut
Non-Verbal Tourist	0,32562	0,32562	0,18408	0,32565	**0,35281**	78
E-shop Clothing	0,01487	0,01487	-	0,01622	**0,01654**	96
Wine	0,70693	**0,71415**	0,43149	0,65148	0,71077	81
Synthetic Control	0,77803	0,78284	0,79617	**0,87547**	0,81947	93
Segmen-tation	0,49268	**0,53918**	0,52128	0,46216	0,52198	90
Glass	0,67486	0,67877	0,36799	0,62821	**0,73110**	113
Bupa	0,002369	0,003378	**0,012035**	0,00095	0,002639	106
Frog_ genus	0,41006	0,44999	**0,54386**	0,44388	0,45705	106
Iris	0,69156	0,67373	**0,80422**	0,70014	0,68359	98
Diabets	0,042443	0,048117	0,000802	**0,06699**	0,042891	95
Wdbc	0,39996	0,39016	**0,41431**	0,37307	0,39476	106
Wpbc	**0,050578**	0,04007	0,020618	0,02514	0,040235	102
Iono-sphere	0,12498	**0,13268**	0,11619	0,1272	0,12432	87
Zoo	0,70669	0,72641	0,76314	**0,77688**	0,74824	97
Mean	0,387259	0,391421	0,382781	0,39096	**0,403313**	**95**

Tablo 4.5 %95 güven aralıkları

Veri	Joint Criteria	Cluster & Select	EAM	PrunedOpt	ECS_SOCP*
Non-Verbal Tourist	[0,204 , 0,381]	[0,002 , 0,226]	[0,184 , 0,184]	[0,022 , 0,4262]	[0,230 , 0,404]
E-shop Clothing	[0,009 , 0,019]	[0,008 , 0,191]	-	[0,014 , 0,017]	[0,004 , 0,021]
Wine	[0,700 , 0,713]	[0,709 , 0,719]	[0,421 , 0,441]	[0,643 , 0,659]	[0,708 , 0,713]
Synthetic Control	[0,716 , 0,839]	[0,703 , 0,832]	[0,757 , 0,835]	[0,871 , 0,879]	[0,778 , 0,860]
Segmen-tation	[0,443 , 0,541]	[0,537 , 0,550]	[0,572 , 0,583]	[0,405 , 0,518]	[0,501 , 0,541]
Glass	[0,301 , 0,552]	[0,210 , 0,466]	[0,359 , 0,376]	[0,625 , 0,630]	[0,709 , 0,753]
Bupa	[0,001 , 0,003]	[0,001 , 0,005]	[0,012 , 0,012]	[0,0008 , 0,001]	[0,0001 , 0,003]
Frog_ genus	[0,049 , 0,360]	[0,159 , 0,470]	[0,541 , 0,546]	[0,299 , 0,499]	[0,055 , 0,401]
Iris	[0,294 , 0,455]	[0,621 , 0,687]	[0,799 , 0,808]	[0,699 , 0,7007]	[0,312 , 0,477]
Diabets	[0,039 , 0,045]	[0,041 , 0,054]	[0,0008 , 0,0008]	[0,064 , 0,069]	[0,390 , 0,046]
Wdbc	[0,011 , 0,308]	[0,089 , 0,378]	[0,414 , 0,414]	[0,251 , 0,420]	[0,263 , 0,447]
Wpbc	[0,023 , 0,057]	[0,204 , 0,055]	[0,020 , 0,020]	[0,023 , 0,026]	[0,026 , 0,054]
Iono-sphere	[0,118 , 0,131]	[0,122 , 0,143]	[0,107 , 0,127]	[0,124 , 0,129]	[0,116 , 0,132]
Zoo	[0,247 , 0,741]	[0,256 , 0,760]	[0,763 , 0,763]	[0,750 , 0,803]	[0,448 , 0,750]

5
SONUÇ VE ÖNERİLER

Optimizasyon, bu tezde ele alınan denetimsiz öğrenme, topluluk seçimi ve budaması gibi karmaşık problemlerin çözümünde kilit bir rol oynamaktadır. Optimizasyon, mühendislikten finansa ve yöneylem araştırmasına kadar çeşitli alanlarda birçok uygulamaya sahip, uygulamalı matematiğin önemli bir alanıdır. Özellikle konveks optimizasyon, etkili yöntemlerinin güçlü teorik temellerle desteklenmesi sayesinde oldukça popülerdir. Konveks optimizasyon problemlerinin avantajlarına lokal optimal çözümlerin her zaman aynı zamanda global çözümler olması; ya da optimallik koşullarının yalnızca gerekli değil, aynı zamanda yeterli olmasının yanı sıra uzmanlık gerektiren polinom karmaşıklığındaki etkili yöntem kullanımı ve optimal değerin sınırlarını bulmak için kullanılabilecek zengin bir dualite teorisi örnek olarak verilebilir.

Bu tez çalışmasında, makine öğrenmesi yöntemlerinden ağırlıklı olarak denetimsiz öğrenme problemlerinden topluluk seçimi (ensemble selection) ve topluluk budamasına (ensemble pruning) odaklanılmış ve makine öğrenmesinin temelini oluşturan optimizasyon teorisi detaylı bir şekilde tanıtılmıştır. Optimizasyon teorisinin önemli teoremleri ve optimal koşulları tanıtılarak, makine öğrenmesindeki temel kavramlar ve bu kavramların matematiksel çerçevesi üzerinde odaklanılmıştır. Geleneksel modellere alternatif olarak, otomatik budama boyutunu belirleyebilen yenilikçi optimizasyon modelleri geliştirilmiştir. Bu modeller, doğruluk ve çeşitlilik arasındaki dengeyi optimize eden, kümeleme problemi için, konveks fark programlama ve ikinci dereceden konik programlama kullanılarak oluşturulan, iki yenilikçi topluluk seçim modelidir.

Topluluk küme seçimi, üç temel adımda oluşturulmaktadır. İlk adım, temel kümeleme tekniği olarak k-ortalamalar kullanılarak topluluk oluşturmaktır. İkinci adım, topluluğun bir alt kümesini elde etmek için ECS_SOCP ve ECS_KFP'yi çözmektir. Son olarak, üçüncü adımda HPGA, birinci adımda belirlenen topluluğun kümeleme sonuçlarını birleştirmek için kullanılmaktadır. ECS_KFP modelinin performansı, ortak kriter yöntemi ve kümele ve seç yöntemi ile karşılaştırılırken, ECS_SOCP

modeli performansı, ortak kriter, kümele ve seç, EAM ve PrunedOPT yöntemleri ile karşılaştırılmıştır.

ECS_KFP modelinin performansı, 6 *UCI* gerçek veri seti üzerinde neredeyse her veri setinde daha iyi NKB sonuçları elde ederek üstün performans göstermiş, modelin esnekliğini ve güvenilirliğini vurgulamıştır. ECS_SOCP modeli için ise 14 *UCI* gerçek veri seti üzerinde yapılan deneysel sonuçlar, bunların ikisi ticaret verisi olmak üzere, önerilen budama algoritmasının, optimizasyon modeli kullanılarak birleşik bir çerçevede optimal çözüme yaklaştığını ve genellikle daha iyi veya eşdeğer NKB sonuçları elde ettiğini göstermektedir. Ayrıca, ECS_KFP modeli için, [40] çalışmasından yararlanılarak, yardımcı teorem ve sonuçlar kullanılarak optimalite koşullarının sağlandığı gösterilmiştir.

Önerilen iki yöntemin kullanılan ortak veri setlerinde karşılaştırılması aşağıda görülebilir:

Tablo 5.1 NKB değerleri

Veri	**ECS_KFP**	**ECS_SOCP**
Wine	**0,781696**	0,71077
Synthetic Control	**0,832222**	0,81947
Segmen-tation	0,461336	**0,52198**
Glass	0,714068	**0,73110**
Zoo	0,735905	**0,74824**
Mean	0,705045	**0,706312**

Önerilen modeller benzer performans göstermek ile beraber ECS_SOCP'nin biraz daha iyi sonuçlar verdiği gözükmektedir. ECS_KFP ve ECS_SOCP arasındaki temel fark ECS_SOCP modelinin yüksek boyut içeren ticari (business) verilerde gösterdiği performanstan ötürüdür.

Sonuçlar şu şekilde özetlenebilir:

- Ortak kriter, kümele ve seç, EAM, PrunedOPT metotları ve kendi modellerimiz olan ECS_KFP ve ECS_SOCP gerçek veri setlerinde test edilmiştir. Genel olarak, ECS_KFP veECS_SOCP arzu edilen sonuçlara ulaşmıştır,
- ECS_KFP ve ECS_SOCP kullanılarak topluluk budamanın bir avantajı, budama boyutuna ihtiyaç duymadan otomatik olarak alt küme bulabilmesidir;

bu özellik, modellerin literatürde tartışılan birçok diğer yöntem arasından sıyrılmasını sağlamaktadır,

- Tablo 4.4'e göre, ECS_SOCP kullanımının ticaret verileri için daha iyi sonuçlar sağladığı gözlemlenmektedir; bu da yöneylem araştırması (OR-Operation Research) uygulamaları için yararlı bir yön olarak değerlendirilmelidir.

Bu çalışmanın sınırlılıkları arasında, δ eşik değerinin modele ek bir değişken gibi entegre edilmesinin gerekli olduğu bulunmaktadır. Gelecekteki çalışmalarda, NKB yerine diğer kalite ve çeşitlilik ölçümlerinin kullanılması ve önerilen yöntemlerin sınıflandırma problemine, öznitelik seçimine, derin öğrenmeye ve görüntü işlemeye uygulanması araştırılabilir.

Son olarak, bu tez, makine öğrenmesinde topluluk seçimi ve budama problemlerinin optimizasyonla çözümüne yenilikçi bir yaklaşım sunmuştur. Yöneylem araştırmasının yenilikçi kullanımı, kümelemeyi optimize edebilir ve karmaşık iş ortamlarında karar verme sürecini iyileştirebilir. Bu metodoloji, rekabet avantajı kazanmak için anahtar içgörüler ve trendler sunarak verimli veri analizi sağlamaktadır. Önerilen modeller, işletmelerin öne çıkmalarını ve uzun vadeli başarı elde etmelerini destekleyecek özelliklerdedir.

KAYNAKÇA

[1] L. Bottou, F. E. Curtis, J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM review*, vol. 60, no. 2, pp. 223–311, 2018.

[2] E. Alpaydin, *Machine learning*. Mit Press, 2021.

[3] Y. Zhang, S. Burer, W. N. Street, "Ensemble pruning via semi-definite programming," *Journal of Machine Learning Research*, vol. 7, pp. 1315–1338, 2006, ISSN: 1532-4435.

[4] J. Azimi, X. Fern, "Adaptive cluster ensemble selection," 2006, pp. 992–997.

[5] X. Z. Fern, W. Lin, "Cluster ensemble selection," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 1, pp. 128–141, 3 Nov. 2008, ISSN: 19321864. DOI: 10.1002/sam.10008.

[6] Y. Hong, S. Kwong, Y. Chang, Q. Ren, "Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm," *Pattern Recognition*, vol. 41, pp. 2742–2756, 9 Sep. 2008, ISSN: 00313203. DOI: 10.1016/j.patcog.2008.03.007.

[7] Z. Yu *et al.*, "Double selection based semi-supervised clustering ensemble for tumor clustering from gene expression profiles," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, pp. 727–740, 4 2014, ISSN: 15455963. DOI: 10.1109/TCBB.2014.2315996.

[8] Z. Yu *et al.*, "Hybrid clustering solution selection strategy," *Pattern Recognition*, vol. 47, pp. 3362–3375, 10 2014, ISSN: 00313203. DOI: 10.1016/j.patcog.2014.04.005.

[9] Z. Yu *et al.*, "Incremental semi-supervised clustering ensemble for high dimensional data clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, pp. 701–714, 3 Mar. 2016, ISSN: 10414347. DOI: 10.1109/TKDE.2015.2499200.

[10] Y.-B. Zhao, *Sparse optimization theory and methods*. CRC Press, 2018.

[11] H. Liu, M.-C. Yue, A. M.-C. So, W.-K. Ma, "A discrete first-order method for large-scale mımo detection with provable guarantees," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, IEEE, 2017, pp. 1–5.

[12] B. K. Sriperumbudur, D. A. Torres, G. R. Lanckriet, "A majorization-minimization approach to the sparse generalized eigenvalue problem," *Machine Learning*, 2011, ISSN: 08856125. DOI: 10.1007/s10994-010-5226-3.

[13] A. Chapnevis, I. Guvenc, E. Bulut, "Traffic shifting based resource optimization in aggregated ıot communication," IEEE, Nov. 2020, pp. 233–243, ISBN: 978-1-7281-7158-6. DOI: 10.1109/LCN48667.2020.9314781.

[14] A. Ramtin, V. Hakami, M. Dehghan, *A perturbation-proof self-stabilizing algorithm for constructing virtual backbones in wireless ad-hoc networks*, 2014. DOI: 10.1007/978-3-319-10903-9_6.

[15] A. Ramtin, V. Hakami, M. Dehghan, "A self-stabilizing clustering algorithm with fault-containment feature for wireless sensor networks," IEEE, Sep. 2014, pp. 735–739, ISBN: 978-1-4799-5359-2. DOI: 10.1109/ISTEL.2014.7000799.

[16] R. Caruana, A. Niculescu-Mizil, G. Crew, A. Ksikes, "Ensemble selection from libraries of models," 2004, p. 18. DOI: 10.1145/1015330.1015432.

[17] J. Jia, X. Xiao, B. Liu, L. Jiao, "Bagging-based spectral clustering ensemble selection," *Pattern Recognition Letters*, vol. 32, pp. 1456–1467, 10 Jul. 2011, ISSN: 01678655. DOI: 10.1016/j.patrec.2011.04.008.

[18] D. G. Ferrari, L. N. D. Castro, "Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods," *Information Sciences*, vol. 301, pp. 181–194, 2015. DOI: 10.1016/j.ins.2014.12.044.

[19] Z.-H. Zhou, J. Wu, W. Tang, "Ensembling neural networks: Many could be better than all," *Artificial Intelligence*, vol. 137, pp. 239–263, 1-2 May 2002, ISSN: 00043702. DOI: 10.1016/S0004-3702(02)00190-X.

[20] D. Hernandez-Lobato, G. Martinez-Munoz, A. Suarez, "Statistical ınstance-based pruning in ensembles of ındependent classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 364–369, 2 Feb. 2009, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.204.

[21] S.-H. Park, J. Fürnkranz, "Efficient prediction algorithms for binary decomposition techniques," *Data Mining and Knowledge Discovery*, vol. 24, pp. 40–77, 1 Jan. 2012, ISSN: 1384-5810. DOI: 10.1007/s10618-011-0219-9.

[22] C. Tamon, J. Xiang, *On the boosting pruning problem*, 2000. DOI: 10.1007/3-540-45164-1_41.

[23] Q. Dai, "A competitive ensemble pruning approach based on cross-validation technique," *Knowledge-Based Systems*, vol. 37, pp. 394–414, Jan. 2013, ISSN: 09507051. DOI: 10.1016/j.knosys.2012.08.024.

[24] I. Partalas, G. Tsoumakas, I. Vlahavas, "An ensemble uncertainty aware measure for directed hill climbing ensemble pruning," *Machine Learning*, vol. 81, pp. 257–282, 3 Dec. 2010, ISSN: 0885-6125. DOI: 10.1007/s10994-010-5172-0.

[25] L. Rokach, "Collective-agreement-based pruning of ensembles," *Computational Statistics & Data Analysis*, vol. 53, pp. 1015–1026, 4 Feb. 2009, ISSN: 01679473. DOI: 10.1016/j.csda.2008.12.001.

[26] M. Wang, H. Zhang, "Search for the smallest random forest," *Statistics and Its Interface*, vol. 2, pp. 381–388, 3 2009, ISSN: 19387989. DOI: 10.4310/SII.2009.v2.n3.a11.

[27] X. Jiang, C.-a. Wu, H. Guo, "Forest pruning based on branch ımportance," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1–11, 2017, ISSN: 1687-5265. DOI: 10.1155/2017/3162571.

[28] C. Qian, Y. Yu, Z.-H. Zhou, "Pareto ensemble pruning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, pp. 2935–2941, 1 Feb. 2015, ISSN: 2374-3468. DOI: 10.1609/aaai.v29i1.9579.

[29] S. Özöğür-Akyüz, B. Ç. Otar, P. K. Atas, "Ensemble cluster pruning via convex-concave programming," *Computational Intelligence*, vol. 36, pp. 297–319, 1 Feb. 2020, ISSN: 14678640. DOI: 10.1111/coin.12267.

[30] M. A. Ali, D. Ucuncu, P. K. Atas, S. Ozogur-Akyuz, "Classification of motor ımagery task by using novel ensemble pruning approach," *IEEE Transactions on Fuzzy Systems*, vol. 28, pp. 85–91, 1 Jan. 2020, ISSN: 19410034. DOI: 10.1109/TFUZZ.2019.2900859.

[31] C. Helmberg, F. Rendl, "A spectral bundle method for semidefinite programming," *SIAM Journal on Optimization*, vol. 10, pp. 673–696, 3 2000, ISSN: 10526234. DOI: 10.1137/S1052623497328987.

[32] M. Muramatsu, T. Suzuki, "A new second-order cone programmıng relaxatıon for max-cut problems," *Journal of the Operations Research Society of Japan*, vol. 46, pp. 164–177, 2 2003, ISSN: 0453-4514. DOI: 10.15807/jorsj.46.164.

[33] S. Kim, M. Kojima, "Second order cone programming relaxation of nonconvex quadratic optimization problems," *Optimization Methods and Software*, vol. 15, pp. 201–224, 3-4 Jan. 2001, ISSN: 1055-6788. DOI: 10.1080/10556780108805819.

[34] S. Wright, J. Nocedal, *et al.*, "Numerical optimization," *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.

[35] S. G. Nash, A. Sofer, *Linear and nonlinear programming*. McGraw-Hill Science, Engineering & Mathematics, 1996.

[36] D. Luenberger, Y. Ye, *Linear and nonlinear programming, 2008*.

[37] S. P. Boyd, L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[38] R. Horst, N. V. Thoai, "Dc programming: Overview," *Journal of Optimization Theory and Applications*, vol. 103, no. 1, pp. 1–43, 1999.

[39] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970, vol. 28.

[40] H. A. Le Thi, T. P. Dinh, H. M. Le, X. T. Vo, "Dc approximation approaches for sparse optimization," *European Journal of Operational Research*, vol. 244, no. 1, pp. 26–46, 2015.

[41] P. S. Bradley, O. L. Mangasarian, J. B. Rosen, "Parsimonious least norm approximation," *Computational Optimization and Applications*, 1998.

[42] F. Rinaldi, F. Schoen, M. Sciandrone, "Concave programming for minimizing the zero-norm over polyhedral sets," *Computational Optimization and Applications*, vol. 46, no. 3, pp. 467–486, Jul. 2010.

[43] M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, pp. 193–228, 1-3 Nov. 1998, ISSN: 00243795. DOI: 10.1016/S0024-3795(98)10032-0.

[44] F. Alizadeh, D. Goldfarb, "Second-order cone programming," *Mathematical Programming*, vol. 95, pp. 3–51, 1 Jan. 2003, ISSN: 0025-5610. DOI: 10.1007/s10107-002-0339-5.

[45] Y. Nesterov, A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, Jan. 1994, ISBN: 978-0-89871-319-0. DOI: 10.1137/1.9781611970791.

[46] V. V. Nabiyev, "Artificial ıntelligence: Problems, methods, algorithms," *Seckin Pub. Co., Ankara*, 2005.

[47] V. Vapnik, "On the uniform convergence of relative frequencies of events to their probabilities," in *Doklady Akademii Nauk USSR*, vol. 181, 1968, pp. 781–787.

[48] Y. Bengio, I. Goodfellow, A. Courville, *Deep learning*. MIT press Cambridge, MA, USA, 2017, vol. 1.

[49] J. D. Kelleher, *Deep learning*. MIT press, 2019.

[50] S. Kocabas, "A review of learning," *The Knowledge Engineering Review*, vol. 6, no. 3, pp. 195–222, 1991. DOI: 10.1017/S0269888900005804.

[51] E. Alpaydın, *Introduction to machine learning second edition*, 2011.

[52] H. Tatlıdil, *Uygulamalı çok değişkenli istatistiksel analiz*. Engin Yayınları, 1992.

[53] M. Aldenderfer, R. Blashfield, "Cluster analysis. a sage university paper," *Newbury Park*, 1984.

[54] M. G. Hayden, *The ensemble system*. Cornell University, 1998.

[55] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and systems magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[56] F. Dell'Orletta, "Ensemble system for part-of-speech tagging," *Proceedings of EVALITA*, vol. 9, pp. 1–8, 2009.

[57] G. S. Romine *et al.*, "Representing forecast error in a convection-permitting ensemble system," *Monthly Weather Review*, vol. 142, no. 12, pp. 4519–4541, 2014.

[58] L. Torjai, F. Kruzslicz, "Mixed integer programming formulations for the biomass truck scheduling problem," *Central European Journal of Operations Research*, vol. 24, pp. 731–745, 2016.

[59] H. Dawid *et al.*, "Management science in the era of smart consumer products: Challenges and research perspectives," *Central European Journal of Operations Research*, vol. 25, pp. 203–230, 2017.

[60] C. G. Csehi, M. Farkas, "Truck routing and scheduling," *Central European Journal of Operations Research*, vol. 25, no. 4, pp. 791–807, 2017.

[61] Ş. Gür, M. Pınarbaşı, H. M. Alakaş, T. Eren, "Operating room scheduling with surgical team: A new approach with constraint programming and goal programming," *Central European Journal of Operations Research*, pp. 1–25, 2022.

[62] A. Esteso, M. Alemany, A. Ortiz, S. Liu, "Optimization model to support sustainable crop planning for reducing unfairness among farmers," *Central European Journal of Operations Research*, vol. 30, no. 3, pp. 1101–1127, 2022.

[63] O. F. C. Heine, C. Thraves, "On the optimization of pit stop strategies via dynamic programming," *Central European Journal of Operations Research*, vol. 31, no. 1, pp. 239–268, 2023.

[64] N. García-Pedrajas, C. Hervás-Martínez, D. Ortiz-Boyer, "Cooperative coevolution of artificial neural network ensembles for pattern classification," *IEEE Transactions on Evolutionary Computation*, vol. 9, pp. 271–302, 3 Jun. 2005, ISSN: 1089778X. DOI: 10.1109/TEVC.2005.844158.

[65] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 2 1996. DOI: 10.1007/BF00058655.

[66] R. E. Schapire, "A brief ıntroduction to boosting," San Francisco: Morgan Kaufmann Publishers Inc., Jul. 1999, pp. 1401–1406.

[67] Z.-H. Zhou, *Ensemble Methods Foundations and Algorithms*, R. Herbrich, T. Graepel, Eds. Chapman and Hall, 2012, ISBN: 9781439830031.

[68] L. Xu, B. Li, E. Chen, "Ensemble pruning via constrained eigen-optimization," IEEE, Dec. 2012, pp. 715–724, ISBN: 978-1-4673-4649-8. DOI: 10.1109/ICDM.2012.97.

[69] G. Martínez-Muñoz, A. Suárez, "Aggregation ordering in bagging," M. Hamza, Ed., International conference on artificial intelligence and applications, 2004, pp. 258–263.

[70] G. Martinez-Munoz, D. Hernandez-Lobato, A. Suarez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 245–259, 2 Feb. 2009, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.78.

[71] Z. Lu, X. Wu, X. Zhu, J. Bongard, "Ensemble pruning via individual contribution ordering," New York, New York, USA: ACM Press, 2010, p. 871, ISBN: 9781450300551. DOI: 10.1145/1835804.1835914.

[72] N. Li, Z.-H. Zhou, *Selective ensemble under regularization framework*, J. A. Benediktsson, J. Kittler, F. Roli, Eds., 2009. DOI: 10.1007/978-3-642-02326-2_30.

[73] L. Zhang, W.-D. Zhou, "Sparse ensembles using weighted combination methods based on linear programming," *Pattern Recognition*, vol. 44, pp. 97–106, 1 Jan. 2011, ISSN: 00313203. DOI: 10.1016/j.patcog.2010.07.021.

[74] G. Tsoumakas, I. Partalas, I. Vlahavas, *An ensemble pruning primer*, 2009. DOI: 10.1007/978-3-642-03999-7_1.

[75] D. Margineantu, T. Dietterich, "Pruning adaptive boosting," San Francisco: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 211–218.

[76] L. Guo, S. Boukir, "Margin-based ordered aggregation for ensemble pruning," *Pattern Recognition Letters*, vol. 34, pp. 603–609, 6 Apr. 2013, ISSN: 01678655. DOI: 10.1016/j.patrec.2013.01.003.

[77] M. Daszykowski, B. Walczak, "Density-based clustering methods," *Comprehensive Chemometrics*, vol. 2, pp. 635–654, 2010, ISSN: 09758887. DOI: 10.1016/B978-044452701-1.00067-3.

[78] E. W. Forgy, "Cluster analysis of multivariate data: Efficiency versus interpretability of classifications," *Biometrics*, vol. 21, pp. 768–769, 3 1965.

[79] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, pp. 47–60, 6 1996. DOI: 10.1109/79.543975.

[80] G. Giacinto, F. Roli, G. Fumera, "Design of effective multiple classifier systems by clustering of classifiers," vol. 2, IEEE Comput. Soc, 2000, pp. 160–163, ISBN: 0-7695-0750-6. DOI: 10.1109/ICPR.2000.906039.

[81] A. Lazarevic, Z. Obradovic, "Effective pruning of neural network classifier ensembles," vol. 2, 2001, pp. 796–801. DOI: 10.1109/ijcnn.2001.939461.

[82] B. Bakker, T. Heskes, "Clustering ensembles of neural network models," *Neural Networks*, vol. 16, pp. 261–269, 2 Mar. 2003, ISSN: 08936080. DOI: 10.1016/S0893-6080(02)00187-9.

[83] R. E. Banfield, L. O. Hall, K. W. Bowyer, W. P. Kegelmeyer, "Ensemble diversity measures and their application to thinning," *Information Fusion*, vol. 6, pp. 49–62, 1 Mar. 2005, ISSN: 15662535. DOI: 10.1016/j.inffus.2004.04.005.

[84] I. Partalas, G. Tsoumakas, I. Vlahavas, "Focused ensemble selection: A diversity-based method for greedy ensemble selection," vol. 178, Patras, Greece: IOS Press, Jun. 2008, pp. 117–121. DOI: 10.3233/978-1-58603-891-5-117.

[85] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, 2nd Edition, D. L. Schilling, Ed. New York: WILEY, 1991, ISBN: 0-471-20061-1.

[86] A. Strehl, J. Ghosh, "Cluster ensembles – a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002, ISSN: 1532-4435. DOI: 10.1162/153244303321897735.

[87] A. Ng, M. Jordan, Y. Weiss, *On spectral clustering: Analysis and an algorithm*, 2001.

[88] C. Zhong, L. Hu, X. Yue, T. Luo, Q. Fu, H. Xu, "Ensemble clustering based on evidence extracted from the co-association matrix," *Pattern Recognition*, vol. 92, pp. 93–106, Aug. 2019, ISSN: 00313203. DOI: 10.1016/j.patcog.2019.03.020.

[89] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[90] J. Weston, A. Elisseeff, B. Scholkopf, M. Tipping, "Use of the Zero-Norm with Linear Models and Kernel Methods," *Journal of Machine Learning Research*, vol. 3, pp. 1439–1461, 2003, ISSN: 15324435.

[91] O. Mangasarian, "Machine learning via polyhedral concave minimization," in *Applied Mathematics and Parallel Computing*, Springer, 1996, pp. 175–188.

[92] T. Pham Dinh, H. A. Le Thi, "Recent advances in dc programming and dca," in *Transactions on Computational Intelligence XIII*, N.-T. Nguyen, H. A. Le-Thi, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1–37, ISBN: 978-3-642-54455-2.

[93] M. Thiao, T. P. Dinh, H. A. Le Thi, "Dc programming approach for a class of nonconvex programs involving l 0 norm," in *Modelling, Computation and Optimization in Information Systems and Management Sciences*, Springer, 2008, pp. 348–357.

[94] H. A. Le Thi, T. Pham Dinh, H. M. Le, X. T. Vo, "DC approximation approaches for sparse optimization," in *European Journal of Operational Research*, vol. 244, 2015, pp. 26–46. DOI: 10.1016/j.ejor.2014.11.031. arXiv: 1407.0286.

[95] D. Üçüncü, S. Akyüz, E. Gül, "A novel auto-pruned ensemble clustering via socp," *Central European Journal of Operations Research*, pp. 1–23, 2023.

[96] S. Sarumathi, N. Shanthi, M. Sharmila, "A comparative analysis of different categorical data clustering ensemble methods in data mining.," *International Journal of Computer Applications*, vol. 81, pp. 46–55, 4 2013, ISSN: 0975-8887.

[97] A. Muñoz, I. Diego, "From Indefinite to Positive Semi-Definite Matrices," *Structural, Syntactic, and Statistical Pattern Recognition*, vol. 4109, pp. 764–772, 2006, ISSN: 03029743.

[98] M. Hein, S. Setzer, L. Jost, S. S. Rangapuram, "The total variation on hypergraphs - learning on hypergraphs revisited," Dec. 2013.

[99] D. Dua, C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

TEZDEN ÜRETİLMİŞ YAYINLAR

Makale

1. Üçüncü, D., Akyüz, S., & Gül, E. (2023). A novel auto-pruned ensemble clustering via SOCP. Central European Journal of Operations Research, 1-23. https://doi.org/10.1007/s10100-023-00887-9

Konferans Bildirisi

1. Üçüncü, D., Akyüz, S., Gül, E., & Wilhelm-Weber, G. (2019). Optimality conditions for sparse quadratic optimization problem. In EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization (pp. 766-777). Springer International Publishing.

2. ÜÇÜNCÜ, Duygu, D., Akyüz, S., Gül, E., & Wilhelm-Weber, G. (2018). Seyrek Kuadratik Optimizasyon Problemi için Optimalite Koşul İncelemesi. In: 38. Yöneylem Araştırması ve Endüstri Mühendisliği Ulusal Kongresi.