Çok Yüzlü Konik Sınıflandırıcılarda Gürbüz Koni Tepe Noktası Tahmini

Golara Ghorban Dordinejad

YÜKSEK LİSANS TEZİ

Elektrik ve Elektronik Mühendisliği

Aralık 2016

Robust Estimation of Cone Vertex in Polyhedral Conic Classifiers

Golara Ghorban Dordinejad

MASTER OF SCIENCE THESIS

Electrical Electronics Engineering

December 2016

Çok Yüzlü Konik Sınıflandırıcılarda Gürbüz Koni Tepe Noktası Tahmini

Golara Ghorban Dordinejad

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik ve Elektronik Mühendisliği
Kontrol ve Kumanda
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Hakan Çevikalp

Aralık 2016

ONAY

Elektrik ve Elektronik Mühendisliği Anabilim Dalı YÜKSEK LİSANS öğrencisi Golara Ghorban Dordinejad'ın YÜKSEK LİSANS tezi olarak hazırladığı "**Çok Yüzlü Konik Sınıflandırıcılarda Gürbüz Koni Tepe Noktası Tahmini**" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Danışman : Doç. Dr. Hakan Çevikalp

Yüksek Lisans Tez Savunma Jürisi:

Üye : Doç. Dr. Hakan Çevikalp

Üye : Doç. Dr. Gürkan Öztürk

Üye : Yrd. Doç. Dr. Hasan Serhan Yavuz

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

Prof.Dr. Hürriyet ERŞAHAN
Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Hakan Çevikalp danışmanlığında hazırlamış olduğum "**Çok Yüzlü Konik Sınıflandırıcılarda Gürbüz Koni Tepe Noktası Tahmini**" başlıklı tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 26/12/2016

Golara Ghorban Dordinejad

ÖZET

Sınıflandırma günümüzde örüntü tanıma, yapay zeka, istatistik, kavramsal psikoloji, görüntü analizi, tıp ve robotik gibi bir çok bilim, endüstri, ticari ve askeri alanlarda kullanılan önemli tekniklerinden biridir. Son yıllarda sınıflandırma problemlerini çözmek için bir çok yöntem önerilmektedir. Bu problemler için çeşitli matematiksel yaklaşımlar kullanılmaktadır. Literatürde, Bayes Sınıflandırma, En Yakın Komşu, Karar Ağaçları, Destek Vektör Makineleri ve Yapay Sinir Ağları en çok kullanılan yöntemlerden bazılarıdır.

Bu tez çalışmasında, çok yüzlü konik fonksiyonları temel alınarak daha geliştirilmiş yöntem elde edilmiştir. Çok yüzlü konik sınıflandırıcı ile daha gürbüz sonuçlar elde etme amacıyla koni tepe noktası tahmini yapılmıştır. Literatürde kullanılan çok yüzlü konik sınıflandırıcılarda koni tepe noktası pozitif verilerin ortalaması olarak alınmaktadır. Bu tez çalışmasında geliştirilen koni tepe noktası tahmin yöntemi için bir çok optimizasyon işlemi ile koni tepe noktası tahmin algoritması oluşturulmuştur. Çok yüzlü konik sınıflandırıcılar, genişletilmiş çok yüzlü konik sınıflandırıcılar ve Destek Vektör Makineleri gibi sınıflandırıcılar ile geliştirilen koni tepe noktası tahmin yöntemi karşılaştırılmıştır. Bu karşılaştırma için *UCI Machine Learning Repository*'deki çeşitli gerçek veri tabanları kullanılmıştır. Ayrıca sentetik bir veri kümesi oluşturulmuştur. Bu veri kümesi için pozitif verilerin ortalamasını kullanan çok yüzlü konik sınıflandırıcı yöntemi ile koni tepe noktası tahmin algoritması karşılaştırılmıştır. Gerçek ve sentetik veri kümeleri üzerinde yapılan deney sonuçları sunulan yaklaşımın veri sınıflandırma problemlerinin çözümünde etkili sonuçlar verdiğini göstermektedir.

Anahtar Kelimeler: Koni tepe noktası tahmini, Çok yüzlü konik sınıflandırıcılar, Genişletilmiş çok yüzlü konik sınıflandırıcılar, Sınıflandırma

SUMMARY

Nowadays, classification is one of the most important techniques used in science, industry, commercial and military fields such as pattern recognition, artificial intelligence, statistics, cognitive psychology, image analysis, medicine and robotics . In recent years, many methods are recommended for solving classification problems. Various mathematical approaches are being used for these problems. In literature, the most used methods are Bayesian Classification, Nearest Neighbor, Decision Trees, Support Vector Machines and Artificial Neural Networks.

In this thesis study, a more improved method is obtained based on polyhedral conic functions. For obtaining more robust polyhedral conic classifiers, cone vertex is estimated by using optimization techniques. In literature, cone vertex of polyhedral conic classifiers is taken as the mean of positive data. In this study, many optimization procedures are used in order to determine cone vertex optimally. The proposed algorithm has been compared with some classifiers such as classic polyhedral conic classifiers, extended polyhedral conic classifiers and Support Vector Machines. For comparison, these specified classifier methods have been applied to various real databases in *UCI Machine Learning Repository*. Also, a synthetic data set was created in order to compare the cone vertex estimation algorithm with the method of determining the polyhedral conic structure by using average of positive data. Experimental results on real and synthetic data sets show that the proposed approach has effective results in solving data classification problems.

Keywords: Cone vertex estimation, Polyhedral conic classifiers, Extended polyhedral conic classifiers, Classification

TEŞEKKÜR

Öncelikle, Eskişehir Osmangazi Üniversitesi Elektrik ve Elektronik Mühendisliği bölümünde Bilgisyarla Görü ve Makine Öğrenimi (MLCV) araştırma labına katılmamı sağlayan ve tez çalışmalarımda yardımcı olan danışmanım Doç.Dr. Hakan Çevikalp'e,

Her zaman arkamda duran ve akademik hayatımın en büyük danışmanı olan babam Farhad Ghorban Dordinejad'e, kadın idolüm olarak gördüğüm annem Shiva Shaeri'e, canım kardeşlerim Aylin Ghorban Dordinejad ve Sanaz Ghorban Dordinejad'a,

Yanımda desteğini her zaman ve her koşulda hissettiğim değerli eşim Hamed Ghorbanpoor'a, ayrıca bana emeği geçen eşimin annesine ve babasına,

Ve son olarak MLCV labındaki değerli arkadaşlarım, Meltem Yalçın Seyirt, Merve Elmas Erdem, Şahin Işık ve Barış Can Çam'a

Bu yüksek lisans tezi aşamasında her türlü destek ve yardımlarını benden esirgemedikleri için minnettarım ve çok teşekkür ederim.

İÇİNDEKİLER

Sayfa

ŞEKİLLER DİZİNİ

Şekil **Sayfa**

ÇİZELGELER DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

Simge veya Kısaltma	Tanım	Sayfa Numarası
CNN	Convolutional Neural Network	18
EPCC	Extended Polyhedral Conic Classifiers	28–31, 33, 40
EPCC-L1-C	Extended Polyhedral Conic Classifiers with L1 Norm Cone Vertex Estimation	40, 45, 52–54, 56, 57
EPCC-L2-C	Extended Polyhedral Conic Classifiers with L2 Norm Cone Vertex Estimation	40, 45, 52–54, 56, 58, 59
GEPSVM	Generalized Eigenvalue Proximal Support Vector Machines	5
kNN	k Nearest Neighbor	1, 16, 17
PCA	Principal Component Analysis	9
PCC	Polyhedral Conic Classifiers	21, 24–30, 40
QP	Quadratic Problem	13
SMO	Sequential Minimal Optimization	13
SVD	Singular Value Decomposition	9
SVDD	Support Vector Data Description	5
SVM	Support Vector Machines	1, 3, 10, 11, 13, 18, 29, 32, 35, 40

1. GİRİŞ VE AMAÇ

Günümüzde sınıflandırma insan hayatının büyük bir bölümünde kendini göstermektedir. Sınıflandırma temel olarak, posta kodlarını otomatik olarak okuyabilen ve harf karakterlerini sıralayan bir prosedür ile karakter tanımada, finansal bilgilere göre bireylerin kredi durumunu belirleyen bir sistem ile kredi başvurusu değerlendirmesinde, hastanın hastalığı ile ilgili ilk teşhisi koyarak acil müdahele yapılabilmesini sağlaması ile çeşitli tıbbı alanlarda, web sitelerinde veya internet gazetelerinde haberlein istenilen konulara göre ayırılmasında, kamera ile trafik yollarının kontrol edilmesi veya polislerin suçluları daha kolay ayırt edebilmesi için görüntü ve ses tanımında, askeri uygulamalar için uydu görüntü analizinde ve daha bir çok günümüz konularında kullanılabilir.

Bilim, endüstri, askeri ve ticari alanlarda ortaya çıkan sınıflandırma sorunları karmaşık ve çoğunlukla çok kapsamlı verilerin kullanılması ile ilgilidir. Sınıflandırma konusu tüm alanlarda büyük önem taşıdığından her geçen gün daha gelişmiş sınıflandırma yöntemleri sunulmaktadır.

Ayrıştırma ve sınıflandırıma konusu ilk olarak Fisher'ın doğrusal ayırım çalışmaları ile *klasik sınıflandırıma* olarak ortaya çıkmıştır. Daha sonra *modern sınıflandırma* yöntemi olarak ele alınan sınıflandırıcılarda daha esnek sınıf modelleri kullanılmıştır ve çoğunlukla her sınıf içindeki özniteliklerin ortak dağılımı tahmin edilmeye çalışılmıştır. Günümüzde ise yaygın olarak kullanılan Destek Vektör Makineleri (SVM), En Yakın Komşu (kNN), Bayes Sınıflandırma, Karar Ağaçları ve Yapay Sinir Ağları gibi sınıflandırıcılar tercih edilmektedir.

Bu tez çalışmasındaki amaç, koni tepe noktası tahmini yapılarak daha gürbüz çok yüzlü konik sınıflandırıcı oluşturmaktır. Koni tepe noktası tahmininde kullanılan çok yüzlü sınıflandırıcı modeli L1 ve L2 koni vasıtasıyla hiperdüzlem bölümlerden ve onların genişletilmiş versiyonlarından oluşmaktadır ve pozitif veriler için kabul bölgesini tanımlamak için kullanılmaktadır. Bu sınıflandırıcının kullanma nedeni ise, pozitif sınıflarını negatiflerden ayırmak için daha uygun ve kompakt konveks bölge şekillerini oluşturmasından kaynaklanmaktadır. Ayrıca, sınıflandırıcı modeli bütünsel optimum sonuç olduğunu sağlaması, büyük problemlere uygun ölçekte olması, gürbüz boşluk bazlı cost fonksiyonu kullanarak çakışmayı önlemesi gibi özelliklere sahiptir.

Bu tez çalışması 4 ayrı bölümden oluşmaktadır. Bunlar sırayla; Literatür Araştırması, Materyal ve Yöntem, Bulgular ve Tartışma ve en son ise Sonuçlar ve Öneriler bölümüdür. Her bölümün genel içeriği aşağıda belirtilmiştir.

İlk olarak, makine öğrenimi ve sınıflandırmanın genel tanımları, yaygın olarak kullanılan çeşitli sınıflandırma modelleri ve çok yüzlü konik sınıflandırmanın tarihçesi **Literatür Araştırma Bölümü**'nde detaylı olarak verilmiştir.

Daha sonra, çok yüzlü konik sınıflandırıcı tanımı, matematiksel yaklaşımı, algoritması ve ayrıca genişletilmiş çok yüzlü konik sınıflandırıcının tanımı ve matematiksel gösterimi **Materyal ve Yöntem Bölümü**'nde ele alınmıştır. Bu bölümde, tezin en önemli çalışması olan ve yenilik içeren konusu, koni tepe noktası tahmin algoritması ve optimizasyon işlemleri yer almaktadır.

Bulgular ve Tartışma Bölümü'nde ise, koni tepe noktası tahmini yapılarak elde edilen çok yüzlü konik sınıflandırıcı modelinin hem sentetik hem de gerçek veri tabanına uygulandıktan sonraki sonuçlar ve yorumları verilmiştir. Sentetik veri tabanındaki sonuçlar demo halinde şekil üzerinde belirtilirken, gerçek veri tabanı sonuçları çizelge olarak verilmiştir. Ayrıca kullanılan veri tabanlarının detaylı bir şekilde açıklaması da bulunmaktadır.

Ve son olarak, tezin son bölümünü oluşturan **Sonuçlar ve Öneriler Bölümü**'nde bu tez çalışmasından elde edilen sonuçlardan bahsedilmiştir. Önerilen yöntemin daha sonraki çalışmalara yararlı bilgiler sağlaması ve geliştirilmesi için önerilerde bulunularak tez tamamlanmıştır.

2. LİTERATÜR ARAŞTIRMASI

Çok yüzlü kümesi bir konveks kümedir ve sınırlı sayıda kapalı yarı uzay kesişiminden elde edilmektedir. Çok yüzlülüler sahip oldukları farklı özellikler ile bir çok alanda kullanılmaktadır. Örnek olarak, sınırlı nokta kümesinin konveks zarfı bir çok yüzlü oluşturmaktadır. Çok yüzlülerin bir diğer önemli özelliği ise $\mathbb{R}^d$'de bağlantılı her konveks altküme çok yüzlü kümesi ile iyi tahmin edilebilir ve bu konu ise örüntü tanıma alanında çok yüzlü bölgelerinin öğrenilmesini ilgi çekici kılmaktadır (Manwani ve Sastry, 2010). Bir çok ikili sınıflandırmalarda, pozitif örneklerin tümü tek bir konveks bölgede yoğunlaşmıştır ve negatif örnekler ise o bölgenin etrafındadır. Böylece çok yüzlü kümesi tarafından *tek-sınıf* bölgesi iyi bir şekilde elde edilir. Bu tür durumlarda sınıflandırıcı öğrenim problemi ile başa çıkmanın bir yöntemi problemi büyük boşluk *tek-sınıf* sınıflandırma problemi olarak fomülleştirmektir (Tax ve Duin, 1999). Bu yöntem bir çeşit iyi tanınmış *Destek Vektör Makinesi (SVM)* (Burges, 1998) yöntemidir. Bu tür yöntemlerde uygun seçilmiş kernel fonksiyonu ile birlikte hiçbir negatif örnek içermeyen ve tüm pozitif örnekleri kapsayan bölge öğrenilebilmektedir. SVM yönteminin genel olarak iyi bir sınıflandırıcı olmasına rağmen, doğrusal olmayan kernel fonksiyonu ile sınıflandırıcı, orijinal öznitelikler uzayında sınıf sınırları için iyi bir geometrik kavram sağlamayabilir. Bu kavramlar sınıflandırıcının farklı bölgelerdeki öznitelik uzayında yerel davranışını anlamak için yararlıdır. SVM sınıflandırıcı ile ilgili detaylı bilgiler bir sonraki bölümde verilmiştir.

Çok yüzlü kümeleri öğrenmek için bir diğer iyi bilinen yaklaşım ise *Karar Ağaç* yöntemidir. İkili sınıflandırma probleminde, eğik Karar Ağaçları her sınıf bölgesini çok yüzlü küme birleşmesi olarak göstermektedir (Rokach ve Maimon, 2005; Duda, P. E. Hart ve Stork, 2000). Tüm pozitif örnekler çok yüzlü tek bir kümeye ait olduğunda, her bir karar düğümü bir alttaki yaprakların negatif sınıf göstergesidir ve sadece tek bir yol pozitif sınıf yapraklarına giden bu ağacı öğrenmek için Karar Ağaçları öğrenme algoritması oluşturulmuştur. Karar Listesi olarak da adlandırılabilen bu tür bir Karar Ağacı tamamen çok yüzlü kümesini göstermektedir. Karar Ağaçları algoritmasında her düğümde optimum hiperdüzlemi öğrenmek için izlenen *Top-Down Greedy* yöntemi genel bir Karar Ağacı algoritması çok yüzlü tek kümeyi öğrenmek için başarısızlığa uğramaktadır. İkili sınıf sınıflandırıma probleminde verilen örneklerde eğer tüm pozitif örnekleri ve hiçbir negatif örneği içermeyen konveks çok yüzlü küme varsa çok yüzlü olarak ayrılabilmektedir (Megiddo, 1988). Eğitim kümesi çok yüzlü olarak ayrılabilmekte ise, Karar Ağacını sabit yapının karar listesi olarak yeniden formülleştirilebilmektedir. Çok yüzlü kümesi oluşturmak için gerekli hiperdüzlem sayısını bildiğimizin varsayılması gerekmektedir. Bu

tür karar listelerini öğrenmek için kısıtlı optimizasyon yöntemleri Orsenigo ve Vercellis, 2007; Astorino ve Gaudioso, 2002; Dundar vd., 2008'de kullanılmıştır.

Bu optimizasyon problemleri konveks küme öğrenmemize rağmen konveks değildir. Tüm pozitif örnekler verilen her doğrusal eşitsizlik kümesini (kesişimleri çok yüzlü küme oluşturan yarı uzaylar) sağlamaktadır. Ancak, her negatif örnek bu eşitsizliklerden bir veya daha fazlasını sağlayamamaktadır ve hangi eşitsizliği sağlamadığı ihtimali de bilinmemektedir. *Kredi atama problemi* olarak da adlandırılan bu problem çok yüzlü kümelerinin öğrenimini zor bir hale getirmektedir (Megiddo, 1988). Astorino ve Gaudioso, 2002'de bu problemi ilk olarak yanlış sınıflandırılmış negatif örnekler için doğrusal programlama ile çözmüştür. Bu yaklaşım hesaplama açısından zordur. Eğer çok yüzlü kümesinin dışarısına düşen her nokta için hangi doğrusal eşitsizliğin sağlandığı önceden bilinirse (çok yüzlü sınıflandırıcının her hiperdüzlem için negatif örnekler ayrı olarak verilirse), sorun daha kolaylaşır. Bu durumda, problem K doğrusal sınıflandırma problemini oluşturur. Ama bu varsayım çok gerçek dışıdır. Dundar vd., 2008 çalışmasında her hiperdüzlem karşılıklı alt sınıflandrıma problemi için küçük bir negatif örnek altkümenin belirli olduğunu varsayarak periyodik optimizasyon algoritması (bir seferde K sınıflandırıcıdan bir sınıflandırıcı optimize etmeyi) önermiştir. Bazı pratik uygulamalarda, her hiperdüzleme karşılık gelen negatif örnek altkümesini bilme varsayımı henüz gerçekçi değildir.

Diğer taraftan, büyük veri sınıflandırıcıları (Cortes ve Vapnik, 1995; Cevikalp, Triggs ve Franc, 2013; Ertekin vd., 2011) ve daha bir çok makine öğrenme tabanlı sınıflandırma algoritmaları *kapalı küme* tanıma için tasarlanmıştır (Scheirer vd., 2013). Kapalı küme tanımasında tüm test sınıfları eğitim aşamasında bellidir. Bu uygulamada, genellikle sınıflandırıcı hiçbir ret işlemini desteklemez ve her test örneğini eğitim kümesindeki örnekler ile farklı olmasına rağmen mutlaka bir sınıfa atamaktadır. Ancak, daha gerçekçi tanıma durumlarında eğitim aşamasındaki tüm test sınıflarına erişim yoktur veya bazen sadece belirli bir sınıf ile ilgilenilmektedir. Örneğin, görsel nesne tanıma problemlerinde nesne sınıfına odaklanılmaktadır ve nesne sınıfına ait olmayan herhangi bir örnek arka plan olarak düşünülmektedir. Her iki durum için, eğitim aşamasında tüm test sınıflarını bilmek mümkün değildir. Daha genel tanıma problemlerinde eğitim sırasında bazı belirli sınıflar vardır ve test örnekleri eğitim aşamasında belirli olmayan sınıflardan gelebilmektedir. Bu tür bir kuruluma *açık küme* tanıma denir (Scheirer vd., 2013) ve sınıflandırıcının görülmemiş sınıflara ait yeni test örneklerini ret etmesi beklenir.

Boşluk tabanlı sınıflandırıcılar belli sınıf örnekleri ile karar sınırı arasındaki mesafeyi maksimum yapmayı amaçlamaktadır. Bu yöntemlerde, ayırma amacıyla doğrusal bir hiperdüzlem kullanılmaktadır. Bilinen verilerden uzak bölgelere ait veriler de bu

verilerin nasıl etiketlenmesi gerektiğine dair bir temel olmamasına rağmen bilinen sınıflara atanmaktadır. Sonuç olarak, bilinmeyen sınıflardan örnekler geldiğinde bu sınıflandırıcılar genelde test aşamasında başarısız olur. Bu çalışmada, boşluk tabanlı sınfılandırıcılar ile modellenmesi mümkün olmayan kompakt pozitif sınıf bölgesini tahmin edebilen uygun sınıflandırıcılar önerilmektedir. Test örnekleri, tahmin edilen nesne bölgesine olan uzaklığa bağlı olarak sınıflandırılmıştır. Ayrıca, bu sınıflandırıcılar ret edilme sonucunu desteklemektedir. Bu önerilen sınıflandırıcılar görsel nesne tanımasında önemli derecede doğrusal SVM sınıflandırıcıları geçmektedir. Üstelik, açık küme tanımasında boşluk tabanlı sınıflandırıcıya göre daha uygun sınıflandırıcılardır.

Son zamanlarda yapılan bir çok çalışmada geleneksel sınıflandırıcılardan vazgeçilip kayıp fonksiyonu kullanarak daha sıkı bir pozitif sınıf modeli tasarlamak için nesne tanıma sınıflandırıcıları önerilmiştir. Bu tür sınıflandırıcılar *tek-sınıf* sınıflandırıcılar olarak adlandırılmaktadır ve pozitif sınıf modelini sadece pozitif sınıf örnekleri ile öğrenebilmektedir. Bu yönde bir çok yöntem önerilmiştir. Destek Vektör Veri Tanımı (SVDD) yöntemi (Tax ve Duin, 2004) pozitif veri etrafında kapalı sınır bulmayı hedeflemektedir. Bu amaçla, SVDD sınıflandırıcı pozitif sınıf örneklerinin çoğunluğunu içeren kompakt bir hiperküre bulmaktadır. Cevikalp ve Triggs, 2012 çalışmasında insan ve yüz tanıma için ardışık konveks model sınıflandırıcı kullanarak negatif örneklerden kompakt pozitif bölgeyi ayırmaktadır. Olvi L. Mangasarian ve Wild, 2006 yaptığı çalışmada pozitif sınıf örneklerine en iyi yerleşen ve aynı zamanda diğer negatif örneklerden mümkün olduğunca uzak bir hiperdüzlem bulan Genelleştirilmiş Özdeğer Yakınsal Destek Vektör Makine (GEPSVM) sınıflandırıcıyı önermiştir. En iyi şekilde yerleşen farklı hiperdüzlem sınıflandırıcı seçenekleri Jayadeva vd., 2007, Cevikalp ve Triggs, 2013 çalışmalarında verilmiştir.

Bu çalışmalara yakın olan kolorektal kanser tanımada (Dundar vd., 2008), pozitiflerin altgrup negatif örneklere karşı sınıflandırılması için hiperdüzlem sınıflandırıcı kümesi optimize edilerek çok yüzlü kabul bölgeleri öğrenilmiştir. Bu tür bir sınıflandırıcı negatif kümesinin bölünmesini gerektirir ama bu işlem hem büyük ölçekli problemlerde zordur hem de negatif kümesinin iyi belirlenmiş altkümelere bölünebilmesi problem oluşturabilir. Pozitifleri yaklaşık olarak sınırlayabilen çok yüzlü oluşturmak için bir çok çeşitli yöntem vardır (Gasimov ve Ozturk, 2006; A. M. Bagirov vd., 2011; Kantchelian vd., 2014; Manwani ve Sastry, 2010) ama yine yerel optimum veya çakışma, kümelenmeye bağlı veya büyük ölçekli uygulamalarda uygunsuz yapan etiketleme gibi problemler mevcuttur. Bu çalışmaların aksine, bu tez çalışmasında kullanılan sınıflandırıcı modeli, bütünsel optimum sonuç olduğunu sağlaması, büyük problemlere uygun ölçekte olması, sınıflandırma için negatif örneklere ihtiyaç olmaması ve gürbüz boşluk bazlı cost

fonksiyonu kullanarak çakışmayı önlemesi gibi özelliklerden oluşan bir konveks formulasyona sahiptir.

Literatürde çok yüzlü konik sınıflandırıcı için bir çok çalışma mevcuttur. Adil M. Bagirov vd., 2013 çalışmasında, örnek sınıflar arasındaki doğrusal parçalı sınırları bulmak için bir algoritma geliştirmişlerdir. Bu algoritma iki aşamadan oluşmaktadır. İlk aşamada, sınıf içerisinde yer alan veri noktalarını belirlemek için çok yüzlü konik kümesi kullanılmıştır. İkinci aşamada ise, bu noktalar çıkartılmıştır ve kalan veri noktaları kullanılarak doğrusal parçalı sınır hesaplanmıştır. Doğrusal parçalı sınırlar bir hiperdüzlemden başlayarak aşamalı olarak hesaplanmıştır. Bu tür bir algoritma büyük veri tabanlarındaki hesaplama zorluğunu oldukça azaltmaktadır.

Daha sonra, Ozturk ve Kasimbeyli, 2013 çalışmasında L1 normlu genişletilmiş çok yüzlü konik sınıflandırıcılar için iki aşamalı bir yaklaşım önermişlerdir. İlk aşamada, farklı merkez noktaları ve normlara göre çeşitli konik fonksiyonlar oluşturulmuştur. Daha sonra, ikinci aşamada sınıflandırıcı fonksiyonunu bulmak için iki-objektif tamsayılı programlama modeli çözülmüştür.

Ozturk, Adil M. Bagirov, vd., 2015 çalışması ile çok yüzlü konik ayırmaya bağlı doğrusal parçalı bir sınıflandırıcı geliştirilmiştir. Bu çalışmadaki sınıflandırıcı çok yüzlü konik fonksiyonları kullanarak sınıflar arasında doğrusal olmayan sınırlar oluşturmaktadır. Sınıfları ayıran çok yüzlü konik fonksiyon sayısı belli olmadığından ayırma fonksiyonlarını oluşturmak için kademeli bir yaklaşım önerilmiştir. Çok yüzlü konik fonksiyonları düzgün ve konveks olmayan hata fonksiyonunun minimize edilmesi ile bulunabilmektedir. Hata fonksiyonu minimizasyonunda iyi bir başlangıç noktası elde etmek için kademeli yaklaşıma bağlı olarak özel bir yöntem önerilmiştir. Bu tür bir yaklaşım, bütünsel veya bütünsele yakın hata fonksiyonu minimumlarının bulunmasını ve sınıf ayırımı için gerektiği kadar az çok yüzlü konik fonksiyon hesaplamasını sağlamaktadır. Bu çalışmada, düzgün olmayan optimizasyon için türevsiz bir yöntem olan ayrık gradyan yöntemi uygulanmıştır.

2.1 Makine Öğrenimi

Makine öğrenimi bilgisayar biliminin altdalıdır ve bilgisayarlara açıkça bir programlama olmadan öğrenme yeteneğini verir. Örüntü tanıma çalışmaları ve yapay zeka hesaplamalı öğrenim teorisinden evrimleşmiş makine öğrenimi, veriye dayalı bilgi öğrenimi ve karar tahmini yapabilecek algoritma yapısı ve tasarımını araştırmaktadır (Kohavi ve Provost, 1998). Bu algoritmalar veriye dayalı tahmin ve kararlar ile girdi örneğinden bir model oluşturmaktadır. Makine öğrenimi istatistik ve olasılık, veri

madenciliği, örüntü tanıma, yapay zeka ve bilgisayar bilimi gibi alanlarla ilgilidir. Başlıca kullanma alanları ise *spam filtreleme*, *optik karakter tanıma*, *arama motorları* ve *bilgisayarlı görme* olarak gösterilmiştir. Genel olarak, makine algılaması, örüntü tanıma, biyoinformatik, tıbbi tanı, konuşma ve elyazısı tanıma, nesne tanıma, yazılım mühendisliği ve robotik gibi bir çok alanda kullanılmaktadır.

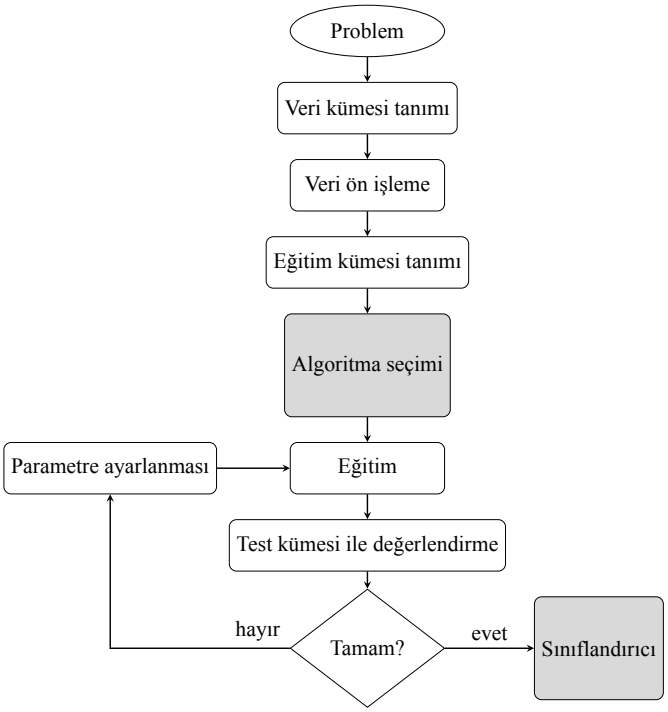
Makine öğrenimi için bir çok uygulama alanı bulunmaktadır en önemli alanlardan biri veri madenciliğidir. Genelde insanlar çoklu öznitelik arasında bağlantı kurma ve analizinde hata yapmaya eğilimlidir ve belirli problemlere çözüm bulmak zorlaşmaktadır. Makine öğrenimi ise böyle problemlere başarılı bir şekilde uygulanarak sistem verimliliğini arttırmaktadır. Makine öğrenimi algoritmalarında her veri kümesindeki her örnek aynı öznitelik kümesi kullanılarak gösterilmektedir. Öznitelikler devamlı, kategorik veya ikili olarak farklı türlerden oluşmaktadır. Makine öğrenimi algoritmaları hedeflenen sonuca göre farklı kategorilere ayrılabilmektedir. Gözetimli öğrenme, gözetimsiz öğrenme ve yarı gözetimli öğrenme sıkça kullanılan üç algoritma türleridir.

2.1.1 Gözetimli öğrenme

Gözetimli öğrenme, etiketli eğitim verilerini kullanarak girdiler ve çıktılar arasındaki optimal fonksiyonu bulan bir makine öğrenme yöntemidir. Gözetimli öğrenme teknikleri yeni test veri örneklerinin etiket tahminini yapabilmek ve tahmin fonksiyonunu bulabilmek için veri örnekleri ile ilgili eğitim verilerini ve sınıf etiketlerini kullanmaktadır. Kısaca, gözetimli öğrenme girdileri hedef çıktılara eşleyen bir işlev üretmektedir ve etiketli örnekler kullanarak sınıflandırıcıları oluşturmaktadır. Bu algoritma ise Şekil (2.1)'de verilmiştir.

Bu algoritmada ilk adım veri toplama olarak belirtilmiştir. Toplanan verilere göre hangi özniteliklerin daha bilgilendirici olduğu tahmin edilerek veya basitçe *Brute-Force* yöntemi ile tüm değerler hesaplanarak en doğru bilgiler ile öznitelikler ayrıştırılmaktadır. Fakat, *Brute-Force* yöntemi ile toplanan veriler girdi olarak atanmaya uygun olmayabilir ve önemli bir ön işleme gerektirebilir (Zhang vd., 2003).

İkinci adım olarak veri hazırlama ve veri önişleme aşamaları vardır. Bu konu üzerindeki çalışmalarda, kayıp bilgiyi elde tutmak için birçok farklı yöntem kullanılmaktadır (Batista ve Monard, 2003). Bu yöntemlerden biri örnek seçimidir. Örnek seçimi sadece gürültülü bilgilerden kurtulmak için değil ayrıca büyük veri kümesi öğreniminin kullanışsızlığını yok etmek için kullanılmaktadır. Bu veri kümelerindeki örnek boyutu minimize edilirken, algoritma kalitesini korumak için örnek seçimi bir optimizasyon problemidir (Yu ve Liu, 2004). Büyük veri tabanlarında veriyi küçültmek algoritmanın daha fonksiyonel ve etkili olmasını sağlamaktadır. Büyük veri tabanında örnekleme için bir çok



Şekil 2.1: Gözetimli öğrenme algoritması (Kotsiantis, 2007)

yöntem mevcuttur. Öznitelik altküme seçim prosedürü mümkün olduğunca veriden alakasız ve gereksiz özellikleri tanımlamak ve çıkarmak için yapılmaktadır (Yu ve Liu, 2004). Bu yöntem ile verinin boyutu düşerken algoritmanın daha hızlı ve daha etkili bir şekilde çalışması sağlanmaktadır. Bir çok özniteliğin bir başka özniteliğe bağlı olması durumu bazen gereksiz yere gözetimli öğrenme sınıflandırma modelinin doğruluk oranını etkilemektedir. Bu problemi çözmek için temel öznitelik kümesinden yeni bir öznitelik kümesi oluşturulması Markovitch ve Rosenstein, 2002 çalışmasında verilmiştir. Bu yöntem öznitelik oluşturma veya öznitelik dönüştürme olarak adlandırılmaktadır. Yeni oluşturulan öznitelikler daha kesin sınıflandırıcılar oluşturmaktadır. Bununla beraber, anlamlı öznitelik oluşturulması sınıflandırıcıya daha iyi bir anlaşılırlık katkısı sağlamaktadır.

2.1.2 Gözetimsiz öğrenme

Gözetimsiz öğrenme, etiketsiz verilerden gizli yapıyı tanımlamak için sonuç çıkaran bir makine öğrenim işlevidir. Eğitim için verilen örnekler etiketsiz olduğundan, potansiyel çözümü değerlendirmek için herhangi bir hata sinyali yoktur ve bu da gözetimli öğrenme ile gözetimsiz öğrenme arasındaki en büyük farktır (Ghahramani, 2004).

Gözetimsiz öğrenimde makine, girdileri kabul ederken gözetimli hedef çıktısı elde etmez veya kendi çevresinden herhangi bir karşılık almaz. Bu da makinenin çevresinden bir

geri bildirim almadan öğrenimini nasıl gerçekleştirdiği konusunda bir gizem oluşturmaktadır. Buna rağmen, karar verme, gelecek girdi tahmini ve girdilerin bir başka makine ile etkin haberleşmesinde kullanılmak için makine hedefinin genel kavramına dayalı olarak gözetimsiz öğrenme için uygun bir sistem geliştirmek mümkündür. Bir anlamda gözetimsiz öğrenme, saf yapılandırılmamış gürültünün ötesinde düşünülen veride model bulma gibi düşünülebilir. Boyutsal indirgeme ve olasılık teorisine dayanan kümeleme gözetimsiz öğrenme için iki basit klasik örnektir (Nowlan, 1990). En çok kullanma alanları veri nesne kümelemesi (D. Wang ve Hinton, 1999) ve veri sıkıştıma işlemidir. Kümeleme için kullanılan başlıca yöntemlerden biri k-ortalama, karışık modelleme, hiyerarşik kümelemedir. Temel Bileşen Analizi (PCA) ve Tekil Değer Ayrışması (SVD) yöntemleri ise gözetimsiz öğrenmede sıkça kullanılmaktadır.

2.1.3 Yarı gözetimli öğrenme

Yarı gözetimli öğrenme algoritması ise uygun işlev veya sınıflandırıcılar oluşturmak için etiketli ve etiketsiz örnekleri birlikte ele almaktadır. Bu öğrenme şeklinde genel olarak etiketli örnek sayısı etiketsiz örnek sayısından daha azdır. Tüm etiketsiz örnekleri etiketleme açısından zor olduğu için bu yöntem pratik bir çözüm olarak kullanılmıştır (Mitchell ve Blum, 1998).

2.2 Sınıflandırma

Sınflandırma insan için kolay iken makine için karmaşık bir görevdir. Makine öğreniminde sınıflandırma yeni bir örneğin veri eğitim kümesine dayanarak hangi kategori kümesine ait olduğunu tespit etme problemidir. Gelen e-mailin *spam* ve *spam olmayan* sınıfına ataması veya hasta özelliklerine (cinsiyet, kan basıncı, diğer belirtiler) göre hasta için bir teşhis ataması sınıflandırma örnekleridir. Sınıflandırma bir örüntü tanıma örneğidir. Makine öğreniminde *sınıflandırma*, uygun biçimde belirlenmiş örneklerin eğitim kümesinin kullanılabilir bir öğrenme şekli olan *gözetimli öğrenme* örneği olarak düşünülür ve verileri esas benzerliklere veya uzaklığa göre kategorilere gruplandırmayı içeren *gözetimsiz öğrenme* ise *kümeleme* olarak tanımlanır (Alpaydın, 2004).

Genelde, örnekler ölçülebilir özellikler kümesi olarak bilinen açıklayıcı değişkenler veya *öznitelikler* olarak analiz edilir. Bu özellikler kategorik (kan türleri için A, B, AB ve O), sıralı (büyük, orta ve küçük), tam sayısal değer (bir e-maildeki sözcük sayısı), gerçel değer (kan basıncı ölçüsü) olarak sınıflandırılabilir. *Sınıflandırıcı* ise sınıflandırma algoritmasında girdi verisini uygun sınıfa eşleştirmekte uygulanan matematiksel fonksiyon olarak tanımlanabilir.

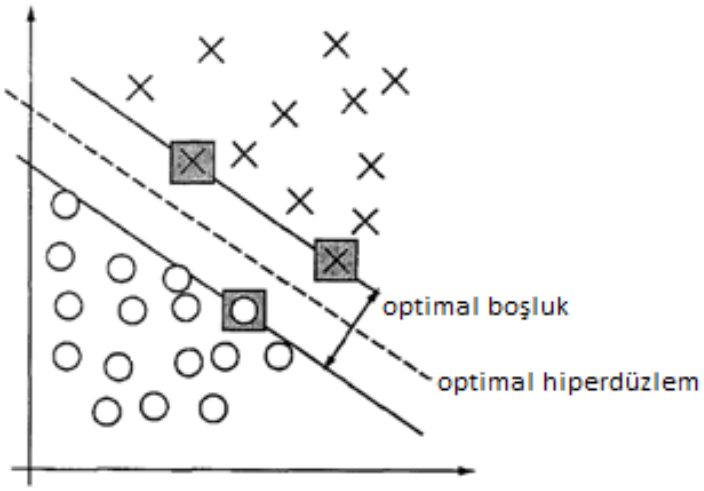
Makine öğreniminde sınıflandırma için eğitim algoritması seçimi kritik bir adımdır. Bu seçim ve sınıflandırıcının değerlendirilmesi genel olarak tahmin doğruluğuna (doğru tahminin toplam tahmin sayısına bölünen yüzdesi) bakılarak yapılmaktadır. Verilerin eğitim ve test kullanımı için farklı yöntemler vardır. İlk yöntem eğitim kümesinin 2/3'nün eğitim için ve diğer kalan 1/3'lük bölümünü ise performans tahmini veya test için kullanılmasıdır. *Cross-Validation* olarak bilinen ikinci yöntemde ise eğitim kümesi ayrışık ve eşit boyutta altkümelere bölünmektedir ve her altküme için sınıflandırıcı diğer tüm altkümelerin bileşiminde eğitilir. Her altkümenin hata oranı ortalaması sınıflandırıcının hata oranı tahmini göstermektedir. *Leave one out-Validation* olarak bilinen bir diğer yöntem ise *Cross-Validation*'ın özel bir modelidir. Tüm test altkümeleri tek bir örnekten oluşmaktadır. Bu yöntem daha karmaşık olmasına rağmen en kesin hata oranı tahmini gerektiğinde kullanılmaktadır. Hata oranını etkileyen bir çok faktör mevcuttur. İlgili özniteliklerin kullanılmaması, daha büyük eğitim kümesine ihtiyaç duyulması, küme boyutunun çok büyük olması, seçilen algoritmanın uygunsuzluğu veya parametre ayarlanması gibi problemler hata oranını etkilemektedir.

Veri sınıflandırılması örüntü tanıma, yapay zeka, istatistik, kavramsal psikoloji, görüntü analizi ve tıp gibi bir çok alanda önemli bir araştırma ve uygulama konusudur. Sınıflandırma için literatürde yaygın olarak kullanılan ve geliştirilen yöntemlerin başında Bayes Sınıflandırma, En Yakın Komşu, Karar Ağaçları, Destek Vektör Makineleri ve Yapay Sinir Ağları sayılabilir. Her problem için farklı bir sınıflandırıcı algoritması daha uygun sonuçlar üretebilmektedir ve bu sebeple en iyi sınıflandırma yöntemi olabilecek tek bir yöntem mevcut değildir.

2.2.1 Destek Vektör Makineleri

Destek Vektör Makineleri genel olarak Support Vector Machines (SVM) olarak adlandırılmaktadır. Makine öğreniminde SVM sınıflandırma ve regresyon analizi için kullanılan verileri analiz eden eğitim algoritmaları ile ilişkili gözetimli öğrenme modelidir (Cortes ve Vapnik, 1995). SVM ile ilgili kapsamlı bilgiler Burges, 1998 ve Cristianini ve Shawe-Taylor, 2000 çalışmalarında bulunabilir. Her iki kategoriden birine ait olarak etiketlenmiş eğitim örnekleri seti göz önüne alındığında (Şekil 2.2), SVM eğitim algoritması yeni örnekleri bir kategoriye ya da diğerine atamaktadır ve bir model oluşturarak olasılıksız ikili doğrusal sınıflandırıcı yapmaktadır. SVM modelinde örnekler uzayda atanmış noktalar gösterimidir ve farklı kategorideki örnekler optimal boşluk ile ayrılmaktadır. İki sınıfa ait verileri birbirinden en uygun şekilde ayırmak için karar sınırları olarak hiperdüzlemleri belirlemektedir. Daha sonra yeni örnekler aynı uzaya atanarak optimal boşluğun hangi tarafına ait olduğu tahmin edilmektedir. Hiperdüzlemlere en yakın veriler iki sınıf arasındaki sınırı belirleyen destek vektörlerini oluşturmaktadır.

Uygulanan SVM modelleri doğrusal SVM ve doğrusal olmayan SVM olarak ikiye ayrılmaktadır. Şekil (2.2)'de verilen SVM modeli ise doğrusal olarak ayrılabilen veriler için optimal hiperdüzlem ve boşluğu göstermektedir.



Şekil 2.2: Doğrusal olarak ayrılabilen veriler için optimal hiperdüzlem (Cortes ve Vapnik, 1995)

Eğer eğitim kümesi doğrusal olarak ayrılabilir ise, bu durumda (ω, b) aşağıdaki fonksiyonlar gibi olur:

$$\omega^T x_i + b \geq 1, \quad \forall x_i \in P$$
$$\omega^T x_i + b \leq -1, \quad \forall x_i \in N$$

Karar yönetimi $f_{\omega,b} = sgn(\omega^T x + b)$ olarak verilir ki ω ağırlık vektörü ve b ofset veya sapma ($-b$ ise eşik) olarak tanımlanır. İki sınıf birbirinden doğrusal olarak ayrıştırılabiliyorsa optimal ayırıcı hiperdüzlemi bulmak için hiperdüzlemin ağırlık vektörünün norm karesi minimize edilir. Minimizasyon işlemi ikinci derece konveks programlama ile oluşturulmaktadır.

$$min_{\omega,b} \quad \phi(\omega) = \frac{1}{2} \|\omega\|^2$$
$$\textit{kısıtlama} \quad y_i(\omega^T x_i + b) \geq 1 \quad i = 1, ..., l$$

Doğrusal olarak ayırılabilen veri kümesinde optimal hiperdüzlem ayırıcı bulunduğunda optimal boşlukta kalan veri noktaları destek vektör noktaları olarak bilinmektedir ve çözüm ise bu noktaların doğrusal kombinasyonu olarak gösterilmektedir. Diğer veri noktaları ise gözardı edilir. Bu sebeple, SVM'in kompleks yapısı eğitim aşamasındaki öznitelik

sayısından etkilenmemiş olur. SVM öğrenim algoritmasında seçilen destek vektör sayısı genellikle azdır. Öznitelik sayısının eğitim örnekleri sayısından daha fazla olması durumundaki öğrenim görevlerinde SVM daha kullanışlıdır.

Bir çok veri kümesinde, maksimum boşluğun SVM'e aday hiperdüzlemler arasından seçimine izin vermesine rağmen yine de SVM ayırıcı hiperdüzlemi bulamayabilir çünkü veri yanlış sınıflandırılmış örnekler içerebilir. Bu problem ancak bazı eğitim örneklerindeki yanlış sınıflandırılmaları kabul eden *yumuşak boşluk* ile çözülebilmektedir (Veropoulos vd., 1999). Bu yöntem için kısıtlamalarda yapay değişken olarak ξ_i, $i = 1, ..., N$ tanımlayarak aşağıdaki eşitsizlikler elde edilebilir:

$$\begin{aligned} \omega.x_i - b &\geq +1 - \xi_i \quad \forall\ y_i = +1 \\ \omega.x_i - b &\leq -1 + \xi_i \quad \forall\ y_i = -1 \\ \xi_i &\geq 0, \end{aligned}$$

Hatanın oluşması için ξ_i sayı birimini geçmesi gerekmektedir ve bu nedenle $\sum_i \xi_i$ eğitim hatası sayısında üst sınırı oluşturmaktadır. Bu durumda Lagrange aşağıdaki gibi ifade edilir.

$$L_p \equiv \frac{1}{2} \|\omega\|^2 + C \sum_i \xi_i - \sum_i \alpha_i \{y_i(x_i.\omega - b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i$$

Bu denklemdeki μ_i Lagrange çarpanıdır ve ξ_i'in pozitiflik uygulaması için tanımlanmıştır. Ancak, bir çok gerçek dünya problemleri eğitim verisinde pozitif örnekleri negatiflerden başarılı şekilde ayırabilen hiç bir hiperdüzlem bulunmayan ve doğrusal olarak ayrılamayan veriyi içermektedir. Verilerin ayrıştırılamaması problemi için çözüm ise verilerin daha büyük boyutlu uzaya eşleştirilerek orada bir ayırıcı hiperdüzlem tanınması ile olabilir. Bu yüksek boyutlu uzay eğitim örnekleri ile kaplı girdi uzayına karşı *dönüştürülmüş öznitelik uzayı* olarak adlandırılmaktadır.

Uygun olarak seçilen yeterli boyuttaki dönüştürülmüş öznitelik uzayı ile herhangi bir tutarlı eğitim kümesi ayırılabilir. Dönüştürülmüş öznitelik uzayındaki doğrusal ayırma orijinal girdi uzayındaki doğrusal olmayan ayırıma denk gelmektedir. Hilbert Uzayı H adı verilen bir başka sonsuz boyutlu uzay için eşleştirme $\phi : \mathbb{R}^d \rightarrow H$ olarak belirtilir. Eğitim algoritması sadece H uzayındaki verinin skaler çarpımına yani $\phi(x_i).\phi(x_j)$ şeklindeki fonksiyonlara bağlı olacaktır. Eğer bir *kernel fonksiyonu* K $K(x_i, x_j) = \phi(x_i).\phi(x_j)$ varsa, eğitim algortiması için sadece K kullanmak yeterlidir ve açıkça ϕ'yi belirtmeye ihtiyaç oluşmayacaktır. Böylelikle, kernel fonksiyonu öznitelik uzayında skaler çarpıma yukarıda belirtilen eşleştirmeyi uygulamadan direk olarak hesaplamasına izin veren özel bir fonksiyon sınıfıdır. Hiperdüzlem oluşturulduğunda, sınıflandırma için kernel fonksiyonu yeni noktaları öznitelik uzayına eşleştirmek için kullanılmaktadır.

Kernel fonksiyonu, eğitim kümesi örneklerinin sınıflandırılması için dönüştürülmüş öznitelik uzayını tanımladığı için en uygun kernel fonksiyonunu seçmek önemlidir. Genton, 2001 birçok kernel sınıfı tanımlamıştır ama en iyi kernel sınıfı olarak herhangi bir bilgi verilmemiştir. Yaygın olarak bilinen uygulamalardan biri eğitim kümesinde potansiyel ayar aralığını tahmin etmek ve çapraz geçerlilik (Cross Validation) kullanarak en iyi kerneli bulmaktır. Kernel fonksiyonu uygun şekilde olduğu sürece, eğitim kümesinden kernel-uyarılmış dönüştürülmüş öznitelik uzayında tam olarak hangi özniteliklerin kullanıldığı bilinmemesine rağmen SVM yine de doğru şekilde çalışacaktır. En çok kullanılan kernel fonksiyonları ise aşağıda belirtilmiştir.

- $K(x, y) = (x.y + 1)^P$
- $K(x, y) = (e)^{-\|x-y\|^2/2\sigma^2}$
- $K(x, y) = tanh(\kappa x.y - \delta)^P$

Bu problemin standart ikinci dereceden problem (QP) yöntemleri ile çözülmesi büyük matris işlemlerinin yanısıra zaman alan numerik hesaplamaları içermektedir ve büyük problemlerde ise genelde yavaş ve kullanışsızdır. Ardışık Minimal Optimizasyon (SMO) ikinci dereceden SVM problemini fazla matris kullanmadan hızlı bir şekilde çözebilen basit bir algoritmadır (Platt, 1999). SMO tüm QP problemini altproblemlerine ayrıştırmaktadır. Keerthi ve Gilbert, 2002 orijinal SMO'ye göre çoğu koşullarda daha hızlı olan iki modifiye SMO versiyonunu önermiştir.

Sonuç olarak, SVM'in eğitim optimizasyon problemi global minimumuna ulaşmaktadır ve yerel minimumda sonuçlanmasını önlemektedir. Ancak, SVM yöntemleri ikilidir ve çoklu sınıf probleminde ise problem çoklu ikili sınıflandırma problem setine dönüştürülmelidir.

2.2.2 Karar Ağaçları

Gözetimli öğrenimde en çok kullanılan sınıflandırıcı yöntemlerden biri de Karar Ağaçlarıdır. Karar Ağaçları, örnekleri öznitelik değerlerine bağlı olarak sınıflandıran ağaçlardır. Karar Ağacındaki her düğüm sınıflandırılacak olan örneğin özelliğini ve her dal düğümünün alabileceği değeri göstermektedir. Örnekler kök düğümünden başlayarak sınıflandırılır ve öznitelik değerine göre sıraya dizilir. Eğitim veri kümesini en iyi şekilde bölebilen öznitelik kök düğüm olarak belirlenir. Bilgi kazanımı (information gain) ve gini indeks gibi eğitim veri kümesini en iyi şekilde bölebilen bir çok farklı öznitelik bulma yöntemi mevcuttur (Moore, 1987). Myopic ölçüleri her özniteliği bağımsız olarak tahmin

ederken, Relief algoritması (Kononenko, 1994) başka özniteliklerin içeriği olarak tahmin etmektedir. Ancak, çalışmaların çoğunluğu en iyi olarak gösterilebilecek tek bir yöntemin olmadığını göstermiştir (Murthy ve K., 1998). Belirli bir veri kümesinde hangi metriğin kullanılması gerektiği yöntemler karşılaştırılmasında büyük önem taşımaktadır. Aynı işlem bölünen verinin her bölümü için tekrarlanarak eğitim verisini aynı sınıfın altkümesine bölünene kadar altağaç oluşturulmaktadır.

Karar Ağaçlarını oluşturmak için literatürde en iyi bilinen algoritma Quinlan ve Cameron-Jones, 1995 tarafından geliştirilen *Layered* algoritmasıdır. Karar Ağaçlarının en kullanışlı özelliği anlaşılırlıklarıdır. Bu yöntem ile bir örneğin ait olduğu bir sınıfa nasıl sınıflandırıldığını anlamak kolaydır. Karar Ağacı testlerin bir hiyerarşisini oluşturduğundan, sınıflandırma sırasında bilinmeyen öznitelik değeri genel olarak daha sonraki düğüm dallarını geçerek bilinmeyen öznitelik değerinin bulunduğu yere gelir ve her dal sınıf dağılımını verir. Çıktı farklı sınıf dağılımının kombinasyonudur ve bunların toplamı 1 olmaktadır. Karar Ağaçlarındaki yapılan tahmin, farklı sınıflara ait örneklerin en az bir özniteliklerinde farklı değerleri olduğu şeklindedir. Karar Ağaçları ayrık ve kesin olan özniteliklerde daha iyi performans göstermeye eğilimdedir (Kotsiantis, 2007).

2.2.3 Bayes Sınıflandırma

Makine öğreniminde Bayes Sınıflandırma yöntemi problem çözümü için olasılık kullanan bir yöntemdir. Bayes Sınıflandırmada verilerin ait olduğu sınıflar belirlidir. Sınıflandırma işleminde genel olarak bir örüntü elde vardır. Buradaki işlem de bu örüntüyü daha önceden tanımlanmış sınıflara atamaktır. Her örüntü öznitelik kümesi tarafından temsil edilmektedir. Bu yöntem daha karmaşık Yapay Sinir Ağları gibi yöntemler ile karşılaştırabilir sonuçlar vermektedir. Bu sınıflandırıcı, Bayes teoremini öneren Thomas Bayes (1702-1761) adına *Bayes Sınıflandırıcı* olarak adlandırılmıştır ve bu teoreme dayanarak sadeleştirilmiştir. Bayes teoremi aşağıda verilen Denklem (2.1) ile ifade edilir.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

B olayı gerçekleştiği koşulda A olayının gerçekleşme olasılığı $P(A|B)$ olarak ifade edilir. $P(B|A)$ ise A olayı gerçekleştiği koşulda B olayının gerçekleşme olasılığıdır. Ayrıca, $P(A)$ ve $P(B)$ ifadeleri A ve B olaylarının olasılıklarıdır. Bu teorem Bayes Sınıflandırıcı için uygulandığında, B öznitelik kümesi ve A sınıf değişkenidir. Bu durumda, $P(A)$ önsel olasılık, $P(A|B)$ ardıl olasılık ve $P(B|A)$ sınıf koşullu olasılık olarak tanımlanır. $P(A)$ önsel olasılık eğitim kümesi kullanılarak elde edilir. $P(B|A)$ saf Bayes ve Bayes güven ağı gibi iki farklı Bayes uygulaması ile hesaplanabilir. $P(A|B)$ ardıl olasılık A ve B kümelerinin her bileşimi için hesaplanır. Daha sonra test verisi en büyük olasılığa karşılık gelen sınıfa atanır (Pang-Ning vd., 2006). Ama, yeterli öznitelik kümesi oluşturmak için

büyük bir eğitim kümesine ihtiyaç vardır. Bu nedenle, ardıl olasılığı doğru olarak tahmin edebilmek bazen zordur.

Bayes Sınıflandırıcısı genel olarak biyomedikal alanında, hastalık tanımlanmasında (Lakoumentas vd., 2012), elektrokardiyografi (EKG) grafiğinin sınıflandırılmasında (Wiggins vd., 2008), elektroensefalografi (EEG) grafiklerinin ayrıştırılmasında (Z. Wang vd., 2011), genetik araştırmalarında (Malovini vd., 2012), yığın mesaj tanımlanmasında (Almeida vd., 2011), metin ayrıştırılmasında (Sebastiani, 2003), ürün sınıflandırma gibi diğer veri madenciliği alanlarında kullanılmaktadır. Metin sınıflandırmasında özellikle büyük başarılar elde edilmesini sağlamıştır. Metin sınıflandırması için en bilinen örneklerden biri spam filtrelemesidir. Bayes spam filtresi spam e-maili spam olmayan e-mailden ayırıştırma konusunda en çok kullanılan yöntemlerden biri olmuştur (Adam vd., 2002).

Bayes Sınıflandırıcı için bir çok avantaj ve dezavantaj mevcuttur (Berger vd., 1994). Avantaj olarak aşağıdaki liste sıralanabilir.

- Önsel bilgi ve veriyi doğal ve prensipli bir yöntem ile birleştirilmesini sağlamaktadır. Bir parametre hakkındaki geçmiş bilgiler birleştirilerek ileriki analiz için önsel dağılım oluşturulabilir. Yeni gözlemler elverişli olduğunda, önceki sonsal dağılım önsel olarak kullanılabilir.
- Asimptotik yaklaşıma dayanmadan, veriler üzerinde koşullu ve kesin olan çıkarımlar sağlanır. Küçük örnek çıkarımı büyük örneklerdeki gibi aynı şekilde ilerler. Bayes analizi fonksiyonlarda muhtemel parametre kullanmadan parametrelerin her fonksiyonunu direk olarak tahmin edebilmektedir.
- Olasılık ilkesine uymaktadır. Eğer ayrı iki örnek orantılı olabilirlik fonksiyonuna uyum sağlarsa her iki örneğin tüm çıkarımları aynı olmalıdır.
- Hiyerarşik modeller ve veri kaybı problemleri gibi modellere uygun ayarı sağlamaktadır.

Bu avantajların yanısıra, Bayes Sınıflandırıcı için bir çok dezavantaj da mevcuttur. Bunlardan bazıları;

- Bu sınıflandırıcı önselin nasıl seçilmesi gerektiğini belirtmez. Önsel seçimi için herhangi bir doğru yöntem yoktur. Bayes çıkarımları, sübjektif önseli formullenmiş

matematiksel önsele dönüştürmek için yetenek gerektirir. Bu nedenle, dikkatli ilerlenmezse yanıltıcı sonuçlar oluşabilir.

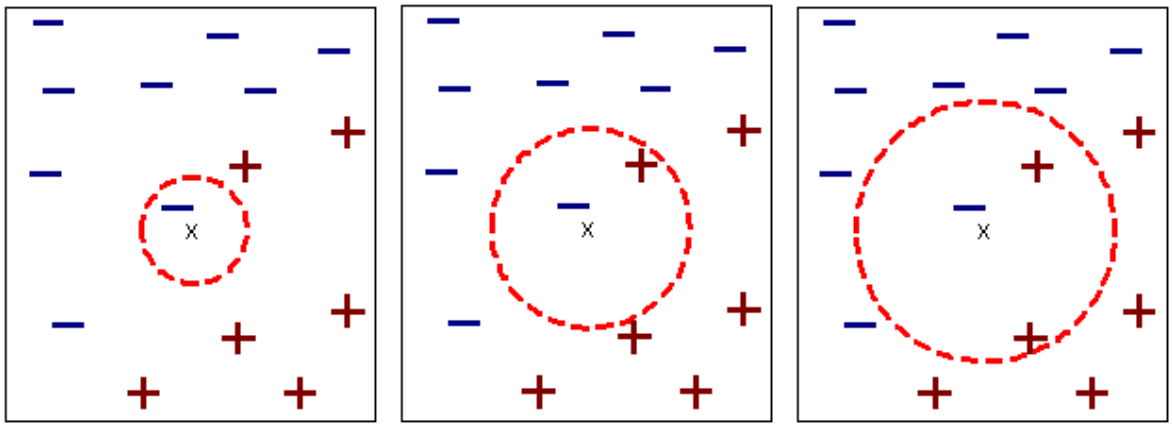
- Bayes Sınıflandırıcı, önsel tarafından aşırı derecede etkilenen sonsal dağılım oluşturabilir. Pratik açıdan bakılınca, seçilen önsel geçerliliğini kabul etmeyen konu uzmanlarını ikna etmek bazen zor olabilir.
- Özellikle çok sayıda parametreli modellerde yüksek hesaplama maliyetine sahiptir. Ayrıca, simülasyonlar aynı rasgele örnekler kullanmadıkça biraz farklı sonuçlar sağlamaktadır.

2.2.4 En Yakın Komşu

Sınıflandırma problemi için bulunan bir çok yöntem arasında en temel ve ilk yöntemlerden biri En Yakın Komşu algoritmasıdır. Bu algoritma Seyyar Satıcı (Travelling Salesman) problemine bir çözüm olarak geliştirilmiştir. En Yakın Komşu formulasyonu ve özelliklerinin analize katkısı ilk olarak (Number ve Change, 1952) tarafından yapılmıştır. Johns, 1959 En Yakın Komşu kuralını deneysel bayes kuralı örneği olarak kullanmıştır. Kanal, 1963'de En Yakın Komşu kuralının örüntü tanıma problemlerinde kullanılmasını önermiştir. En Yakın Komşu sınıflandırıcıları etiketlenmiş örnek kümesi için herhangi bir önişleme süreci gerektirmez. Bu sınıflandırıcı girdi vektör örneğini en yakın olan komşu sınıfına atamaktadır.

Problem ile ilgili daha fazla önsel bilgi ile sınıflandırma algoritmasının gerçek durumu yansıtması sağlanabilir. Örnek olarak, eğer tüm sınıfın önsel olasılığı ve koşullu durum yoğunluğu bilinmekteyse, Bayes teoremi beklenen yanlış sınıflandırma oranını minimize etmesiyle optimum sonuçlar oluşturur (Duda ve P. E. Hart, 1973). Bununla birlikte, bir çok örüntü tanıma problemlerinde girdi örüntünün sınıflandırması her sınıfın örnek boyutunun nerede küçük olduğuna ve muhtemelen gerçek dağılım olasılığını temsil etmediğine bağlıdır. Bu durumda bir çok teknik, kümeleme ve ayırıcı analiz (Duda ve P. E. Hart, 1973) gibi, öznitelik uzayındaki mesafe veya benzerlik kavramına dayanmaktadır. Bir çok durumda K En Yakın Komşu *kNN* algoritması (T. Cover ve P. Hart, 1967) sınıflandırmak için kullanılır. Karar kuralı, k-en yakın vektör komşusu (Öklid anlamında) olarak ifade edilen sınıf etiketini girdi örüntüsüne atayan parametreli olmayan basit bir süreç sağlamaktadır. *kNN* kuralı yetersiz bir prosedürdür. Ancak, sonsuz örnek durumunda hata oranının *1NN* kuralı için Bayes optimum hata oranının iki katı kadar üstten sınırlandırılmış olduğunu göstermiştir. Ayrıca, k artması ile bu hata oranı asimptotik olarak optimum orana yaklaşmıştır (Fukunaga ve Hostetler, 1975). *kNN* algoritması tanıtılmasından itibaren bir çok araştırma yapılarak geliştirilmiştir (Hellman, 1970; T. T. M. Cover, 1968) . Bu ilgiyi arttırmasının sebebi bir çok küçük örnek boyutundaki

asimptotik davranışından daha çok hesaplama kolaylığı ve belki de şaşırtıcı iyi sonuçlar elde edilmesidir (Dasarathy, 1980). *kNN* sınıflandırıcı veri tarafından karakterize edilmiş problemlerde kullanılmaktadır. *kNN* sınıflandırıcı kullanılmasındaki en önemli problemlerden biri sınıf etiketinin girdi vektörüne atanmasında her örnek vektörü için eşit ağırlık kullanılmasıdır. Bu problem çoğunlukla örnek kümelerin örtüştüğü durumlarda zorluk oluşturmasına sebep olmaktadır. Bir diğer problem ise girdi vektörü bir sınıfa atandığında o sınıfa ait olduğunu belirten bir güç ve dayanıklılığın olmamasıdır. Yeni x



Şekil 2.3: Sınıflandırılacak x verisi için k-En Yakın Komşu analizi. Soldan sağa sırayla; 1-En Yakın Komşu, 2-En Yakın Komşu, 3-En Yakın Komşu (Lanzi, 2012)

verisini sınıflandırmak için verinin diğer eğitim verilerine mesafesi hesaplanarak k-En Yakın Komşu belirlenmektedir (Şekil 2.3). En yakın komşuların çoğunluk sınıf etiketleri kullanılarak yeni verinin sınıf etiketi belirlenmektedir.

2.2.5 Yapay Sinir Ağları

Yapay Sinir Ağları insan beyninin bilgi işlemesindeki sinir sisteminin kullandığı yöntemden yararlanarak geliştirilmiş bir sınıflandırma yöntemidir. İnsan sinir hücrelerinin içerdiği nöronlar farklı yapılar şeklinde birleşerek ağları oluşturmaktadır. Bu sinir ağları öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptir. Sınıflandırmada kullanılan Yapay Sinir Ağları da böyle bir model oluşturmaktadır (Lee vd., 2009; Behnke, 2003).

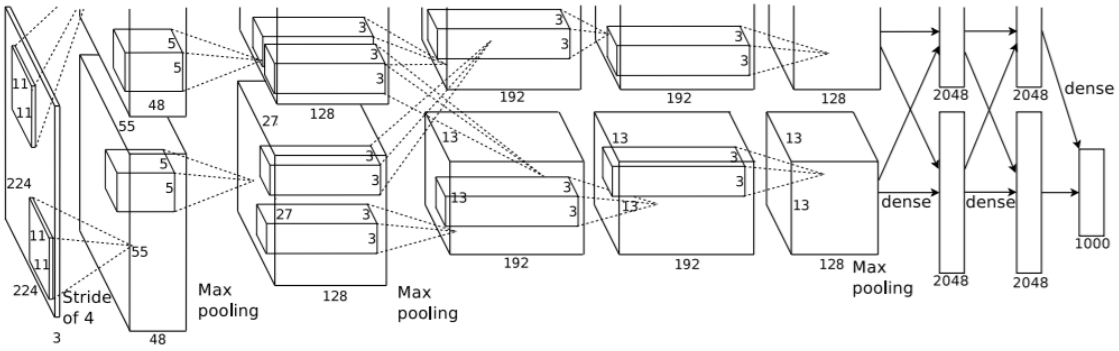
Yaygın olarak kullanılan yüzeysel sınıflandırıcılarda, girdi verisi için insan analizine bağlı olarak elle tasarlanmış anahtar öznitelik çıkarma yöntemi kullanılır ve daha sonra hedef sınıfını tanımak ve sınıflandırmak için bu öznitelikler veri vektörü olarak genel sınıflandırıcıya uygulanır. Bir diğer anlamda sınıflandırıcı girdi verisine değil tamamen

özniteliğin nasıl oluşturulduğuna bağlıdır. Modern sınıflandırma yöntemi olarak bilinen Konvolüsyonel Sinir Ağları (CNN) öğrenilen öznitelikleri saklı katmanlar şeklinde eğitir; düşük katman detaylarından (kenarlar, köşeler, vb.) başlayarak yüksek seviye detaylarına (şekil, karakter, vb.) kadar sade bir sınıflandırıcıya daha iyi veri göstergesi elde etmek için inceler. Kapasitesi değişken derinlik ve genişlik ile kontrol edilebilir ve imgenin niteliği hakkında güçlü ve çoğunlukla doğru tahminler yapabilir. Benzer-boyut katmanlı standart ileri besleme sinir ağları ile kıyasla konvolüsyonel sinir ağlarının daha az bağlantıları ve parametreleri vardır ve bu nedenle, eğitmek daha kolaydır.

Derin konvolüsyonel sinir ağları kıvrımlı katmanları alt-örnekleme katmanları ile değişen, temel görme beyin zarının içindeki sade ve karmaşık hücreleri anımsatan bir ileri besleme Yapay Sinir Ağlarıdır. İleri beslemeli Yapay Sinir Ağlarının en gelişmiş halidir ve görüntü tanımada oldukça başarılı sonuçlar vermiştir. Bu sinir ağları (LeCun vd., 1998) birden fazla aşamadan oluşan eğitilebilir çok aşamalı mimarilerdir. Her aşamanın giriş ve çıkışında öznitelik haritaları denilen dizi setleri bulunmaktadır. Her aşama iki temel katmandan oluşmaktadır: konvolüsyon ve öznitelik havuzlama. Genellikle bir, iki ya da üç katmanlı aşamaların ardından bir sınıflandırma modülünden oluşmaktadır. Her katman bir önceki katmanın sonucuna göre öznitelik çıkarır ve tüm katmanları birleştirerek eğitir ve öznitelik hiyerarşisini öğrenebilir. Derin konvolüsyonel sinir ağları, konvolüsyon ve altörnekleme katmanlarının yapısına ve ağların eğitim yöntemine bağlı olarak değişmektedir. Derin konvolüsyonel sinir ağları son ağ katmanı çıktısına bağlı olarak direk olarak bir sınıflandırıcı gibi kullanılabilirken (Bengio vd., 2013; Lecun vd., 2015; Huang ve LeCun, 2006), öte yandan çıkarılan öznitelikleri bir sınıflanıdırcıya (örneğin, SVM) besleyerek tek başına öznitelik çıkarma olarak olarak da kullanılabilir.

Derin konvolüsyonel sinir ağları ilk olarak Fukushima, 1980 tarafından tanıtıldı. Bu yöntem daha sonra LeCun vd., 1998 tarafından geliştirildi ve Cireşan vd., 2011; Simard vd., 2003 çalışmaları sonucunda düzeltilerek sadeleştirildi. Ama daha sonra özellikle bilgisayarla görü alanında Destek Vektör Makinelerin (SVM) kullanışı ile eski popülerliğini kaybetmişti. Ancak, Krizhevsky vd., 2012 Imagenet büyük veri tabanları görsel tanımadaki (ILSVRC) imge sınıflandırmasında önemli derecede yüksek doğruluk oranı ile CNN kullanımını yeniden gündeme getirmiştir. Önerilen yöntemde kullanılan CNN yapısı Şekil (2.4)'de gösterilmiştir. Bu başarı, büyük miktarda etiketli imge için büyük bir CNN eğitimi ve Montavon vd., 1998'te önerilen CNN'de yapılan bir kaç değişiklik sonucunda ulaşılmıştır. Son zamanlarda, bu yöntemin el yazısı ile yazılmış sayılardan (MNIST), elle yazılmış karakterlere, 3 boyutlu oyuncaklar (NORB) ve yüzlere kadar değişen çeşitli veri tabanlarında kullanılmıştır. Derin sinir ağları sınıflandırma için geleneksel yaklaşımlardan önemli farklılıklar göstermektedir. Birincisi, yüzeysel mimarilere göre daha karmaşık modeller öğrenme kapasitesine sahip derin mimarilerdir. Bu eğitim algoritmaları el ile

öznitelik hazırlamaya gerek kalmadan otomatik olarak öznitelik çıkarır ve sonuç olarak güçlü bir öğrenmeye izin verir.

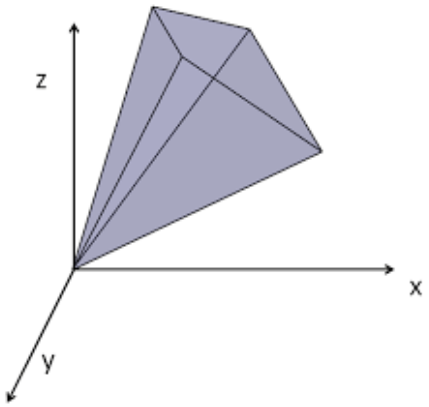


Şekil 2.4: Derin konvolüsyonel sinir ağları katmanlarının mimari örneği (Krizhevsky vd., 2012)

3. MATERYAL VE YÖNTEM

3.1 Çok Yüzlü Koni

Çok yüzlü geometrisi matematikte $\{x|Ax \leq b\}$ kümesi, A matrisi $\in R^{m\times n}$ ve b vektörü $\in R^{m\times 1}$ olarak tanımlanırken çok yüzlü koni ise $\{x|Ax \leq 0\}$, A matrisi $\in R^{m\times n}$ olarak belirtilmektedir. Tüm yüzeylerin birleştiği nokta olan tepe noktası ise çok yüzlülerin yapısında önemli bir rol oynamaktadır.



Şekil 3.1: Çok yüzlü koni geometrik şekli (Chekuri, 2009)

3.1.1 Konveks küme

Konveks bir küme, bir bölgedeki herhangi iki nokta arasında oluşabilecek düz çizginin yine aynı bölge içerisinde kalmasıdır. Matematiksel olarak konveks kümeler x ve $y \in R^n$ noktalarının konveks kombinasyonuna eşittir.

$$\alpha x + (1-\alpha)y \quad 0 \leq \alpha \leq 1$$

Bir optimizasyon probleminde konvekslik önemli bir rol oynamaktadır çünkü konveks kümenin önemli bir özelliği doğrusal olmayan bir optimizasyon işleminde konveks fonksiyonundan elde edilen sonuç yerel optimum ise aynı zamanda bütünsel optimumdur.

3.1.2 Çok yüzlü konveks küme

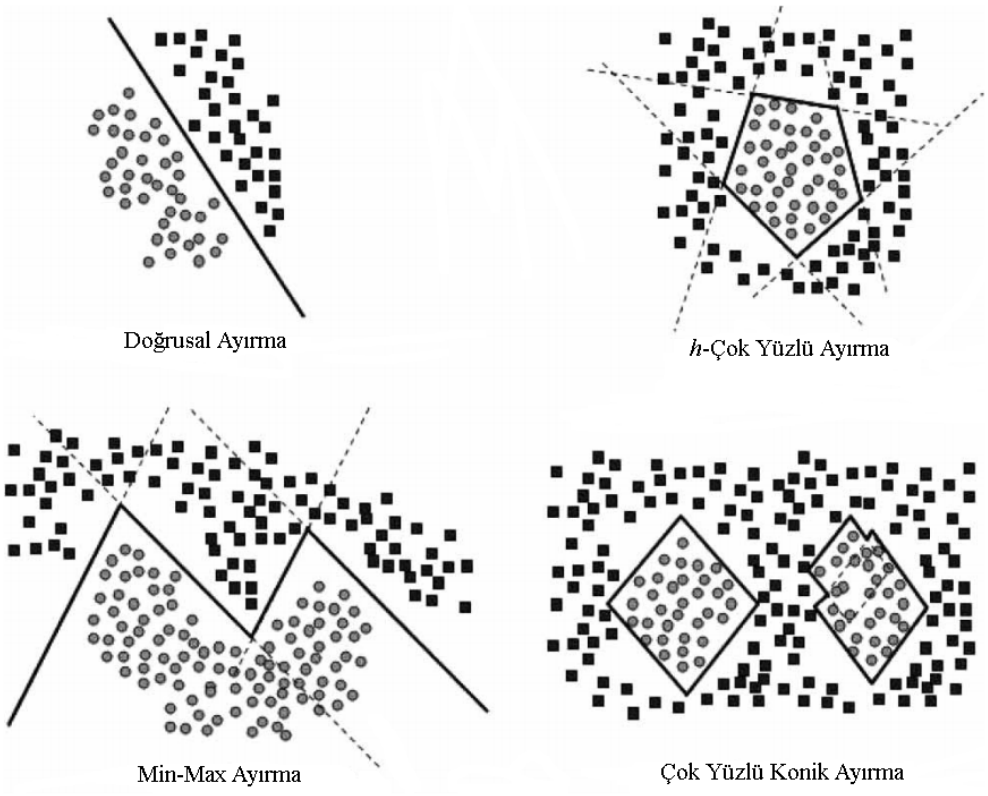
Bir sınırlı çok yüzlü, sonlu noktalar kümesinin konveks zarfıdır. n boyutlu gerçel uzayda ($\mathbb{R}^n$) boş olmayan sonlu iki A ve B kümelerinin ayırma problemini inceleyelim. Bu

tür kümelerin konveks zarfları ayrık ise $co(A) \cap co(B) = \varnothing$, kesin bir ayırma düzlemi oluşturulabilir. Yani çok yüzlü kümeler bir konveks kümedir ve kapalı yarı yüzeylerin sonlu kümelerinin kesişimiyle oluşmaktadır. Konveks zarfları kesişiyorsa yanlış sınıflandırma miktarını en aza indirebilecek hiperdüzlem elde etmek için bir doğrusal programlama yöntemi uygulanabilir. Bu tür bir düzlem bulmak için önemli bir yöntem K. Bennett ve O. L. Mangasarian, 1992 tarafından önerilmiştir. Benzer yaklaşıma dayalı algoritmalar ise Bradley vd., 1999 ve Adil M. Bagirov, 2005 çalışmaları ile geliştirilmiştir. Çok yüzlü kümelerin en önemli özelliklerinden biri herhangi bir konveks altkümesi de bir çok yüzlü kümesi ile yaklaştırılabilir ve bu özellik ise çok yüzlü bölgelerinin öğrenimini örüntü tanımada önemli bir konu yapmaktadır.

İki A ve B kümeleri hiperdüzlem ile ayrılamıyorsa $co(A) \cap co(B) \neq \varnothing$, ve A'nın dışbükey zarfı ile B kümesi kesişmiyorsa $co(A) \cap (B) = \varnothing$ h-çok yüzlü ayırma ile ayrılabilir (Megiddo, 1988). A küme elemanlarını içeren konveks çok yüzlü (h-yarı uzay kesişimi) ve çok yüzlü dışında kalan B elemanlarını oluşturan h hiperdüzlemler vardır. Astorino ve Gaudioso, 2002 çalışmasında h-çok yüzlü ayırma ile ilgili bir algoritma geliştirmişlerdir. Bu algoritmada alt problemleri bulan azalma yönlü doğrusal programlamanın iteratif çözümüne dayalı dışbükey veya içbükey olmayan parçalı doğrusal bir hata fonksiyonu tanıtılmıştır. Adil M. Bagirov, 2005 h-çok yüzlü ayırma yönteminin (Astorino ve Gaudioso, 2002) bir genellemesi olarak kabul edilebilir min-max ayırma kavramını tanıtmıştır. Eğer A ve B kümeleri ayrık ise o zaman min-max ayrılabilir olduğu gösterilmiştir. Adil M. Bagirov, 2005 bu duruma bir hata fonksiyonu tanımlamıştır ve minimizasyonu için bir algoritma geliştirmiştir.

Gasimov ve Ozturk, 2006 çalışmasında, $A - C$ dışbükey ve $(A - C) \cap B = \varnothing$ olan dışbükey koni C tarafından kısmen dizilmiş gerçek normlu uzayda iki boş olmayan A ve B kümelerini ele almışlardır. $A - C$ ve B ayrımı için özel bir çok yüzlü fonksiyon kullanılarak dışbükey olmayan vektör optimizasyon problemlerinin minimum noktaları böyle bir ayırma yöntemi ile nitelendirilmiştir. Gasimov ve Ozturk, 2006 tarafından kullanılan fonksiyonlar geliştirilerek $\mathbb{R}^n$ 'de verilen iki sonlu A ve B ayrık kümelerinin ayırma fonksiyonu olarak çok yüzlü konik fonksiyonları tanımlamışlardır. Bu doğrusal fonksiyonlara bir bölüm eklenerek genişletilmiş L1 norm oluşturulmuştur. Bu fonksiyonun grafiği en fazla 2^n yarı uzay kesişimi içeren bir altdüzey kümesine sahip çok yüzlü konidir. Böylece, çok yüzlü konik fonksiyonları kullanılarak doğrusal olmayan ayırma fonksiyonu üreten iteratif bir PCC algoritması geliştirilmiştir (Gasimov ve Ozturk, 2006). Bu algoritma doğrusal programlamanın alt problem çözümlerine dayanmaktadır. Her iterasyondaki alt problemlerin çözümü A kümesinin belirli bir bölümünü B kümesinden ayıran çok yüzlü konik fonksiyon ile sonuçlanmaktadır. Bu bölümü A kümesinden çıkararak, algoritma sonraki iterasyonlara geçmektedir. Ortaya çıkan ayırma fonksiyonu tüm oluşturulan

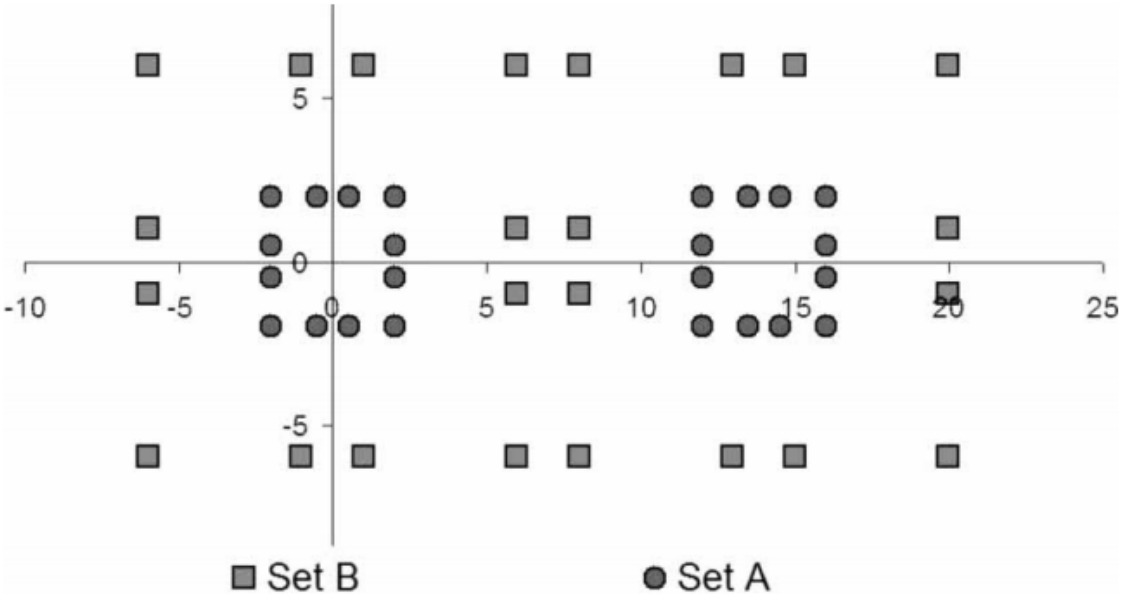
fonksiyonların noktasal minimumu olarak tanımlanır. Algoritma sınırlı iterasyon sayısında sonlanmaktadır ve iki sınırlı kümeyi ayırmak için gereken en fazla iterasyon sayısı her kümedeki eleman sayısını geçmemelidir.



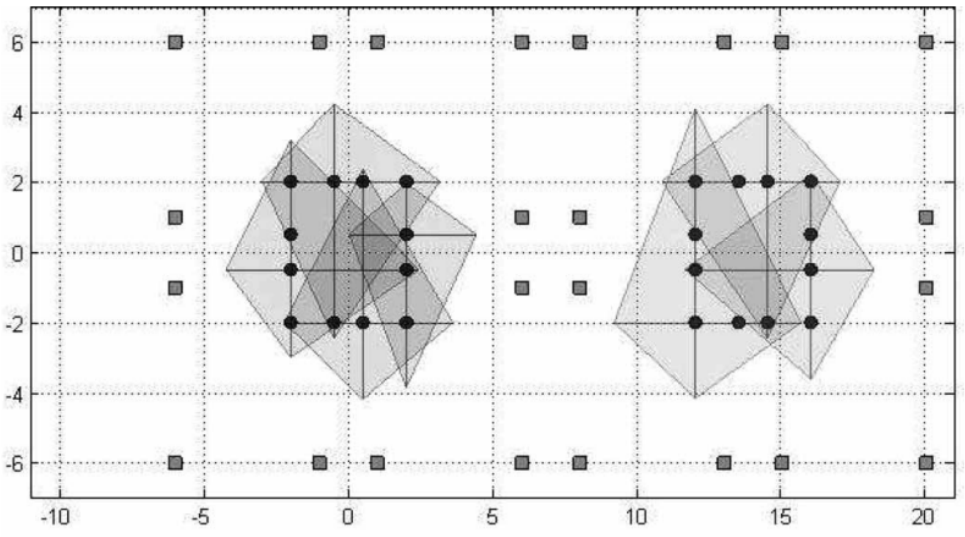
Şekil 3.2: Dört ayırma yönteminin geometrik gösterimi (Gasimov ve Ozturk, 2006)

Çok yüzlü konik sınıflandırıcılarının geometrik olarak nasıl bir yapı oluşturduğu (Gasimov ve Ozturk, 2006) çalışmasında örnek olarak verilmiştir. A ve B kümelerini bu yöntem ile ayırmak için kullananılan veri kümesi Şekil (3.3)'de verilmiştir.

A kümesini B kümesinden ayırmak için A kümesine uygulanan çok yüzlü konik sınıflandırıcı ile verilerin çok yüzlü şekilde tanımlanması Şekil (3.4)'de 2 boyutlu olarak ve Şekil (3.5)'de ise 3 boyutlu olarak gösterilmiştir.



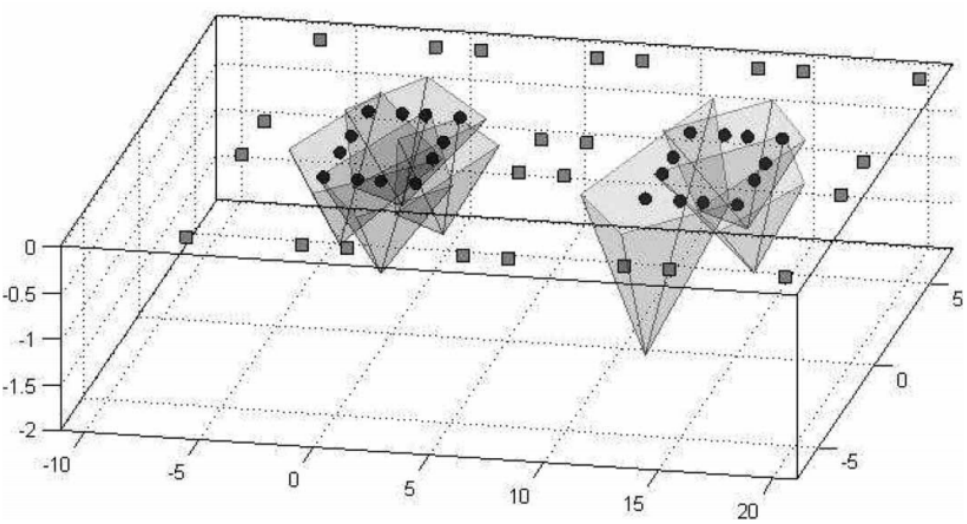
Şekil 3.3: A ve B veri setlerinden oluşan veri kümesi (Gasimov ve Ozturk, 2006)



Şekil 3.4: Çok yüzlü konik sınıflandırıcının 2 boyutlu görünümü (Gasimov ve Ozturk, 2006)

3.2 Çok Yüzlü Konik Fonksiyon (PCC)

$\mathbb{R}^n$'de iki ayrık kümeyi ayırmak için kullanılan çok yüzlü fonksiyonlardan özel bir fonksiyon elde edilmiştir. Bu $f_{(\omega,\gamma,c,b)} : \mathbb{R}^n \to \mathbb{R}$ fonksiyonu aşağıdaki gibi tanımlanmıştır



Şekil 3.5: Çok yüzlü konik sınıflandırıcının 3 boyutlu görünümü (Gasimov ve Ozturk, 2006)

ve *Polyhedral Conic Classifier (PCC)* olarak adlandırılmaktadır.

$$f_{(\omega,\gamma,c,b)}(x) = \omega^T(x-c) + \gamma \left\|x-c\right\|_1 - b \tag{3.1}$$

Burada $\omega, c \in \mathbb{R}^n$ ve $\gamma, b \in \mathbb{R}$ olarak tanımlanmıştır. Bu denklemde ω ve x'in sayısal çarpımı $\omega' x = \omega_1 x_1 + \cdots + \omega_n x_n$ şeklinde ve $x \in \mathbb{R}^n$ vektörünün L1 normu ise $\left\|x\right\|_1 = |x_1| + \cdots + |x_n|$ şeklinde verilmiştir.

Teorem 1 *Denklem* (3.1) *de tanımlanan* $f_{(\omega,\gamma,c,b)}$ *fonksiyon grafiği* $(c, -b)$ *tepe noktasına sahip çok yüzlü bir konidir.*

İspat 1 *Bu teoremi ispatlamak için öncelikle;*

(i) fonksiyon grafiği $(c, -b) \in \mathbb{R}^n \times \mathbb{R}$ *tepe noktalı bir konidir, ve (ii) fonksiyonun her altdüzey kümesi dışbükey çok yüzlüdür.*

(i) bölümü için, $(f_{(\omega,\gamma,c,b)})$ *grafik*$-(c,-b)$ *şeklinde bir küme düşünelim:*

$$(f_{(\omega,\gamma,c,b)})\textit{grafik}-(c,-b) = \{(x-c, \alpha+b) : \omega^T(x-c) + \gamma \left\|x-c\right\|_1 - b = \alpha\}$$

olsun ve $x - c = y$ *ve* $\alpha + b = \beta$ *olarak kullanalım;*

$$(f_{(\omega,\gamma,c,b)})grafik - (c,-b) = \{(y,\beta) : \omega^T(y) + \gamma \left\|y\right\|_1 = \beta\}$$

açıktır ki bu fonksiyon orijinde olan tepe noktası ile bir konidir.
Eğer $(y, \beta) \in f_{(\omega,\gamma,c,b)}$ *grafik* $-(c, -b)$ *ise,* $\omega^T(y) + \gamma \|y\|_1 = \beta$ *olur.*

Bu neden ile her $\lambda > 0$ *için:*

$$\lambda\omega^T(y) + \lambda\gamma \|y\|_1 = \lambda\beta,$$

veya

$$\omega^T(\lambda y) + \gamma \|\lambda y\|_1 = \lambda\beta,$$

Bu fonksiyon $(\lambda y, \lambda\beta)$ *noktasının* $(f_{(\omega,\gamma,c,b)})$ *grafik*$-(c, -b)$ *ait olduğunu göstermektedir. Sonuç olarak bu fonksiyon orijinde bulunan tepe noktasına sahip bir konidir.*

İkinci bölüm (ii) için reel bir sayı α *düşünelim. O halde Denklem* (3.1) *göre* $f_{(\omega,\gamma,c,b)}$ *fonksiyonun altdüzey kümesi:*

$$S_\alpha = \{x \in \mathbb{R} : f_{(\omega,\gamma,c,b)}(x) = \omega^T(x - c) + \gamma \|x - c\|_1 - b \leqslant \alpha\}$$

L1 norm tanımı kullanılarak;

$$S_\alpha = \{x \in \mathbb{R} : f_{(\omega,\gamma,c,b)}(x) = \omega^T(x - c) - b \leqslant \alpha\}$$

olarak yazılabilir, ayrıca;

$$\omega^T(x - c) = \sum_{i=1}^{n} (\omega_i + \gamma sgn(x_i - c_i))(x_i - c_i)$$

S_α *altdüzey kümesinin en fazla* 2^n *yarı-uzay kesişimi olduğunu ve sonuç olarak bir dışbükey çok yüzlü olduğunu göstermektedir.* □

Yapılan ispat sonucunda $f : \mathbb{R}^n \to \mathbb{R}$ tanımlı fonksiyon grafiği bir koni ve her $\alpha \in \mathbb{R}$ için tüm altdüzey kümeleri $S_\alpha = \{x \in \mathbb{R}^n : f(x) \leqslant \alpha\}$ çok yüzlüdür. Bu teoreme göre Denklem (3.1) 'de verilen her fonksiyon çok yüzlü bir konidir ve PCC olarak adlandırılmaktdır.

3.2.1 Çok yüzlü konik fonksiyon algoritması

A ve B kümeleri $\mathbb{R}^n$'de tanımlı iki küme ve $I = \{1, ..., m\}$ ve $J = \{1, ..., p\}$ olsun;

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

PCC algoritmasında her iterasyon sonucunda Denklem (3.1)'deki fonksiyon oluşturulmaktadır ve ω, γ ve b parametreleri belirlenmiş doğrusal programlama alt

probleminin çözümü olarak hesaplanmaktadır. Altdüzey kümesinin tüm B elemanları dışarıda kalacak ve mümkün olduğunca A elemanlarını içeride kapsayacak şekilde tüm uzayı ikiye bölen çok yüzlü konik fonksiyon bu parametreler kullanılarak tanımlanır. Algoritma sonraki elemanları A'dan çıkararak bir sonraki iterasyona geçmektedir ve modifiye edilmiş bu küme için yeni bir ayırma fonksiyonunu oluşturmaktadır. Bu işlem boş küme elde edilene kadar devam etmektedir. Sonuçta oluşan ayırma fonksiyonu tüm oluşturulan fonksiyonların noktasal minimumu olarak tanımlanmaktadır.

Algoritma 1 Genel PCC Algoritması

Başlangıç: $i = 1, l_i = L, A_i = A$

1. Adım A_i'den rastgele bir a^i elemanı seçilerek P_i alt problemi çözülür:

$$(P_i) \quad \min \left(\frac{y' e_m}{m} \right) \tag{3.2}$$

Kısıtlama:

$$\omega'(a^l - a^i) + \gamma \left\| a^l - a^i \right\| - \xi + 1 \leq y_l, \quad \forall l \in L_i, \tag{3.3}$$

$$-\omega'(b^j - a^i) - \gamma \left\| b^j - a^i \right\| + \xi + 1 \leq 0, \quad \forall j \in J, \tag{3.4}$$

$$y = (y_1, y_2, ..., y_m) \in \mathbb{R}^m_+, \omega \in \mathbb{R}^n, \gamma \in \mathbb{R}, \xi \geq 1 \tag{3.5}$$

(P_i)'nin çözümü olarak $\omega^i, \gamma^i, \xi^i, y^i$ alındığında:

$$f_i(x) = f_{(\omega^i, \gamma^i, \xi^i, y^i)}(x) \tag{3.6}$$

olur.

2. Adım $L_{i+1} = \{i \in L_i : f_i(a^l) + 1 > 0\}, A_{i+1} = \{a^l \in A_i : i \in L_{i+1}\}, i = i + 1$ olsun ve eğer $A_i \neq \varnothing$ ise **1. Adım**'dan devam edilir.

3. Adım A ve B kümelerini ayıran $f(x)$ fonksiyonu tanımlanır:

$$f(x) = \min_i f_i(x) \tag{3.7}$$

son.

Algoritma (1)'de, her i iterasyonunda, algoritma A_i'den rastgele bir a^i elemanı seçmektedir ve P_i doğrusal alt-problemini çözerek ω^i, γ^i ve ξ parametrelerini hesaplamaktadır. Tüm bu parametreler daha sonra Denklem (3.6)'de f_i fonksiyonunu tanımlamak için kullanılmaktadır. Teorem (1)'e göre $(x, z) \in \mathbb{R}^n \times \mathbb{R}$ elemanlarını içeren

ve $z = f_i(x)$ olan f_i fonksiyon grafiği $(a^i, -\xi^i)$ tepe noktasına sahip bir konidir. Kısıtlama (3.5)'de belirtilen $\xi \geq 1$ koşulu koni tepe noktasının $z = 0$ hiperdüzleminin altında yani $\mathbb{R}^n \times (0, -\infty)$ yarı-uzayında yerleştirilmesi gerektiğini sağlamaktadır. Kısıtlama (3.3)'de $\{x : f_i(x) \leq -1\}$ altdüzey kümesine göre a^i elemanı ve a^i'e yakın tüm A^i elemanlarının çok yüzlü içerisinde olmak zorunda olduğunu sağlamaktadır. A^i elemanlarının a^i'e yakınlığı (3.2) objektif fonksiyonun optimal değeri tarafından tanımlanmaktadır. $\{x : f_i(x) \leq -1\}$ altdüzey kümesi objektif fonksiyon değerinin sıfıra yakın olana kadar a^i'in yanısıra bir çok A^i elemanını kapsayacaktır. Böylelikle objektif fonksiyon (3.2), (3.3) ve (3.5) kısıtlamalarında, eğer minimum sıfır ise, tüm A^i elemanlarının f_i'nin altdüzey kümesi $\{x : f_i(x) \leq -1\}$ tarafından kuşatılmış olduğunu sağlamaktadır. Diğer taraftan, kısıtlama (3.4) ise her iterasyonda B kümesi elemanlarının $\{x : f_i(x) < 1\}$ altdüzey kümesinin dışında kalma zorunluluğunu getirmektedir. Teorem (1)'de anlatılan çok yüzlü konik fonksiyon özelliği gereğince her iterasyonda böyle bir *ayrılabilirlik* mümkün olduğu dikkate alınmalıdır. Algoritmada belirtilen ayırma yöntemi PCC ayırımı olarak adlandırılmaktadır (Gasimov ve Ozturk, 2006).

Teorem 2 *PCC algoritması sınırlı iterasyon sayısı sonucunda sonlanmaktadır ve Denklem (3.1)'de tanımlı* $f : \mathbb{R}^n \to \mathbb{R}$ *fonksiyonu* A *ve* B *kümelerini* $f(a) < 0, \ \forall a \in A$ *ve* $f(b) > 0, \ \forall b \in B$ *açısından tam olarak ayırmaktadır.*

İspat 2 *İlk olarak* f_i*'nin* $(\omega_i, \gamma_i, \xi_i) \in \mathbb{R}^n \times \mathbb{R}_+ \times [1, \infty)$ *bir çözümü olduğunu ve* A_i*'den en az bir eleman (örneğin* a_i *elemanını) ve tüm* B *kümesini ayırdığını gösterelim.* $\omega_i = 0$ *ve* $\gamma_i = 1$ *olarak alındığında* $f_i(x) = \gamma \left\|x - a^i\right\|_1 - 1$ *fonksiyonu ki* $f_i(a^i) = -1 < 0$ *ve* $f_i(b^j) = \gamma \left\|b^j - a^i\right\|_1 - 1, \forall j \in J$ *olarak elde ederiz.* $b^j \in B$ *olduğundan* $\gamma \left\|b^j - a^i\right\|_1 - 1 > 0, \forall j \in J$ *olur. Bu nedenle* γ *yeterince büyük olduğunda* $\gamma \left\|b^j - a^i\right\|_1$ *terimi büyük yapılabilir. Yeteri kadar büyük* γ *için* $f_i(b^j) > 0, \forall j \in J$ *olur, yani* $f_i(x) = \gamma \left\|x - a^i\right\|_1 - 1$ *fonksiyonu* a^i *ve* B *kümesini* $f_i(a^i) < 0$ *ve* $f_i(b) > 0, \forall b \in B$ *açısından ayırmaktadır.*

$\widetilde{A}_i$*, i'nci iterasyonunda* P_i *problem çözümü kullanılarak oluşturulan* f_i *fonksiyonu tarafından* B *kümesinden ayrılmış elemanlara sahip* A_i*'nin altkümesi ve* A_{i+1}*,* f_i *fonksiyonu tarafından* B *kümesinden ayrılamamış elemanlara sahip* A_i*'nin altkümesi olsun. Eğer bu küme boş değilse, algoritma devam etmektedir.* A *kümesinin sınırlı sayıda elemanları olduğu için bu işlem sınırlı sayıda iterasyondan sonra sonlanmaktadır. Sonuç olarak* A *kümesinden* $\widetilde{A}_1, \widetilde{A}_2, ..., \widetilde{A}_I$ *kısımları ve aşağıdaki özelliklere sahip* $g_1, g_2, ..., g_I$ *fonksiyonları elde edilmektedir.*

$$A = \bigcup \widetilde{A}_i$$

$$f_i(a) < 0, \forall a \in \widetilde{A}_i,$$

$$f_i(b) > 0, \forall b \in \widetilde{B}, i = 1, 2, .., I.$$

Fonkiyon $f(x) = \min_i \{f_i(x)\}$ *için* $a \in \widetilde{A}_i$ *gibi her* $a \in A$ *için* $i \in \{1, 2, ..., I\}$ *mevcuttur,* $f_i(a) < 0$ *ise* $f(a) = \min_i \{f_i(a) < 0\}$. *Bunun yanısıra, tüm* $i \in \{1, 2, ..., I\}$, $b \in B$ *için* $f_i(b) > 0$ *olduğundan* $f(b) > 0$ *olur.* □

PCC algoritması özellikle istatistik ve makine öğrenimi yaklaşımlarına dayanmaktadır ve veri sınıflandırması için uygulanabilmektedir. Veri sınıflandırma sorunlarını çözmek için uygulanan matematiksel programlama tekniklerinin iki temel yaklaşımı vardır. İlki dış yaklaşımdır ve doğrusal olmasının pek önemi olmayan belli bir fonksiyonu ile verilen eğitim kümesinin ayrılmasına dayanır. İkincisi ise iç yaklaşımdır. Bu yaklaşımda verilen eğitim kümeleri küme merkezleri tarafından yaklaştırılır. Yeni veri vektörleri en yakın kümeye atanır ve sonucunda bu kümeyi içeren kümeye atanmış olur. Algoritma (1)'de verilen yöntem dış yaklaşıma dayanmaktadır.

3.3 Genişletilmiş Çok Yüzlü Konik Fonksiyon (EPCC)

Bu tez çalışmasında kullanılan sınıflandırıcı modeli Gasimov ve Ozturk, 2006'de önerilen çok yüzlü konik fonksiyonundan - özellikle L1 konileri aracılığıyla hiperdüzlem bölümleri- ve pozitif kabul edilen bölgelerini tanımlamak için onların uzantısından yararlanarak oluşturulmuştur. Seçilen model, nispeten iyi lokalize edilmiş pozitif sınıfın daha kapsamlı negatif sınıfından ayırmak için uygun bir takım kompakt konveks (uygun ağırlıklar için) bölge şekillerini sağlamaktadır, bu doğal olarak gürbüz boşluk tabanlı öğrenime izin vermektedir ve serbest parametre sayısı makul kalarak çakışma (overfitting) ve çalışma süresini kontrol etmektedir.

Genişletilmiş çok yüzlü konik fonksiyon Denklem (3.8) olarak verilmiştir ve *Extended Polyhedral Conic Classifier (EPCC)* olarak adlandırılmaktadır.

$$f_{(\omega,\gamma,c,b)}(x) = \omega^T(x - c) + \gamma^T|x - c| - b \tag{3.8}$$

Bu fonksiyonda test elemanı olarak $x \in \mathbb{R}^d$, $c \in \mathbb{R}^d$ koni tepe noktası, $\omega \in \mathbb{R}^d$ ağırlık vektörü ve b ise ofsettir. Genişletilmiş çok yüzlü konik fonksiyonunda $|u| = (|u_1|, ..., |u|_d)^T$ bileşen yönlü modülü ve $\gamma \in \mathbb{R}^d$ ilgili ağırlık vektörünü belirtmektedir.

Belirtilen çok yüzlü konik fonksiyonlarında (3.1) ve (3.8), pozitifler için $f(x) < 0$ ve negatifler için $f(x) > 0$ karar bölgeleri kullanılmıştır. Benzer şekilde, boşluk tabanlı eğitim yöntemleri pozitifler için $f(x) \leq -1$ ve negatifler için $f(x) \geq +1$ uygulamaktadır. Her iki çok yüzlü konik sınıflandırıcıda, pozitif bölge esasen c'de ortalanmış L1 koni vasıtasıyla

bir hiperdüzlem kesitidir, özellikle $x \in \mathbb{R}^d$ bölgesinde $z = \omega^T(x-c) - b$ hiperdüzlemi L1 koni $z = \gamma \left\|x-c\right\|_1$ PCC veya EPCC için diyagonal-ölçekli L1 koni $z = \gamma^T|x-c| = \left\|diag(\gamma)(x-c)\right\|_1$ üstünde kalmaktadır.

PCC yöntemi için $b > 0,\ \gamma > 0,\ \left\|\omega\right\|_\infty < \gamma$ ($\left\|\omega\right\|_\infty = max_{i=1}^d|u_i|,\ \ \infty$ norm), $f(x) < \tau$ bölgesi konveks, $\mathbb{R}^d$'de kompakttır ve c tepe nokasını içermektedir. Benzer olarak, EPCC için $b > 0,\ \gamma > 0,\ |\omega| < \gamma_i,\ i = 1, ..., d$ ve her τ için $f(x) < \tau$ bölgesi tekrar konveks, kompakt ve c tepe noktasını içermektedir. PCC ve EPCC yapıları bu şekilde uygulanmıştır.

Geometrik olarak, verilen kısıtlamalar altında sonuç bölgeleri çok yüzlü $2d$ tepe noktası ile sınırlıdır ve her pozitif ve negatif boyunca c'den başlayarak yarı-eksen koordine edilmiştir. Çizgiler karşıt tepe noktasında birleşmektedir böylelikle c'de kesişerek, tüm boyutu b tarafından yönetilen biçimsiz ama yine eksene hizalanmış *uçurtma* şekilli çok yüzlü bir bölge oluşturmaktadır. EPCC bölge genişlikleri bağımsız olarak her eksen boyunca orantılanabilir olmasına rağmen PCC'de sınırlı derecede yönlere bağımlılık (anizotropi) hala mümkün iken bölgeler birlikte birleştirilir.

PCC'de girdi olarak alınan öznitelik vektörleri üzerinden boşluk tabanlı sınıflandırıcıları tanımlamak için öznitelik vektörü $\tilde{x} \equiv \begin{pmatrix} x-c \\ \left\|x-c\right\|_1 \end{pmatrix} \in \mathbb{R}^{d+1}$ vektörüne, ağırlık vektörü ise $\tilde{\omega} \equiv \begin{pmatrix} -\omega \\ -\gamma \end{pmatrix} \in \mathbb{R}^{d+1}$ vektörüne arttırılmaktadır ve $\tilde{b} = b$ olarak kullanılmaktadır. Daha sonra, PCC karar fonksiyonu pozitifler için bilinen doğrusal SVM formu $\tilde{\omega}^T\tilde{x} + \tilde{b} > 0$ ve negatifler için $\tilde{\omega}^T\tilde{x} + \tilde{b} < 0$ kullanmaktadır.

Benzer şekilde EPCC için, öznitelik vektörü $\tilde{x} \equiv \begin{pmatrix} x-c \\ |x-c| \end{pmatrix} \in \mathbb{R}^{2d}$ vektörüne, ağırlık vektörü ise $\tilde{\omega} \equiv \begin{pmatrix} -\omega \\ -\gamma \end{pmatrix} \in \mathbb{R}^{2d}$ vektörüne arttırılmaktadır ve $\tilde{b} = b$ olarak alınmıştır. Pozitifler için yine aynı SVM formu $\tilde{\omega}^T\tilde{x} + \tilde{b} > 0$ ama bu sefer $2d$ boyutunda kullanılmaktadır. PCC ve EPCC için yukarıdaki ∓ 1 boşluklar bilinen ± 1 SVM boşluklarına aktarılır ve maksimum boşluk eğitimi için standart SVM kullanılmasına izin verir. Bu nedenle artırılmış öznitelik vektörlerinde bilinen SVM karesel programalanın çalışması yeterli olacaktır.

$$\begin{aligned} arg\,min_{\tilde{\omega},\tilde{b}} \quad & \frac{1}{2}\tilde{\omega}^T\tilde{\omega} + C_+\sum_i \xi_i + C_-\sum_j \xi_j \\ s.t. \quad & \tilde{\omega}^T\tilde{x}_i + \tilde{b} + \xi_i \geq +1, \quad i \in I_+, \\ & \tilde{\omega}^T\tilde{x}_j + \tilde{b} - \xi_i \leq -1, \quad j \in J_-, \end{aligned} \tag{3.9}$$

Pozitif ve negatif eğitim örnekleri için $I_{\pm}$ indeks kümesi, ξ'ler örneklerin boşluk kısıtlama ihlali için arttıran yapay değişkenler ve $C_{\pm}$ ilgili ceza ağırlığıdır.

PCC ve EPCC öznitelik vektörlerini yukarıdaki eğitim işlemine ekleyerek sırayla Çok Yüzlü Konik Sınıflandırıcı (PCC) ve Geliştirilmiş Çok Yüzlü Konik Sınıflandırıcı (EPCC) yöntemlerini elde ederiz. Görünürde doğrusal formda olmalarına rağmen bu sınıflandırıcılar aslında asimetrikdir. Pozitifler içeride ve negatifler ise etrafında kalmaktadır. Çok yüzlü konik bölgeleri genellikle kompakttır ve pozitifte merkezlenmiş durumundadır.

Algoritma 2 Genel Koni Tepe Noktası Tahmin Algoritması

Başlangıç: $y_i \in \{-1, +1\}$ sınıf etiketi, n_+ pozitif örnek sayısı, n_- negatif örnek sayısı, $n = n_+ + n_-, \omega_0, \gamma_0$, seçilen tolerans değeri $(\tau) = 0.5, \epsilon > 0, \alpha_0 = [(1/n_+)$*(1'ler vektörü)],

while $\|(|\omega| - |\omega_{eski}|)\| < \tau$ **do**
    1.Aşama:
    for $i = 1 : n$ **do**
        random veri noktası (x_t) seçilir,
        eğitilecek veri ve etiketi oluşturulur,
        if $y_i(\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_{1(2)} + b) < 1$ yani (Hinge<1) **then**
            EPCC-L1-C ve EPCC-L2-C türev işlemleri: ∇F
        else (Hinge>1)
            $\nabla F = \alpha/n$
        end
        α_{eski} ve α_{yeni} arasında Gradient Stochastic Descent yöntemi uygulanarak α güncellenir: $\alpha_{yeni} = \alpha_{eski} - \epsilon\nabla F$
    end
    2.Aşama:
    Elde edilen α_{yeni} ile birlikte çok yüzlü koni parametreleri ω, γ ve b güncellenir,
end

3.4 Koni Tepe Noktası Tahmini

Genel olarak PCC ve EPCC yöntemlerinde tepe noktası c parametresi pozitif verilerin ortalaması olarak alınmıştır. Bu tez çalışmasında belirlenen amaç çok yüzlü konik sınıflandırıcılar ile daha başarılı sonuç elde etmek için koni tepe noktası tahminini optimal olacak şekilde yapabilmektir. Çok yüzlü konik sınıflandırıcının koni tepe noktası tahmini için c parametresinin tahmin edilmesi gerekmektedir. Koni tepe noktası ($c = X_{pos}\alpha$) pozitif

örneklerin doğrusal kombinasyonu olarak yazılabilir. Burada X_{pos} pozitif örnekleri içeren bir matristir ve α ise doğrusal kombinasyon katsayılarını göstermektedir. Değişken parametre α tahmin edilerek koni tepe noktası tahmin edilebilir. Bu tahmin için çok yüzlü konik objektif fonksiyonunun α'ya göre türevi alınır. Daha sonra Stochastic Gradient Descent yöntemi kullanılarak α güncellenir. Güncellenen α ile eğitim kümesi yeniden düzenlenir ve SVM algoritması kullanılarak ω, γ, b çok yüzlü koni parametreleri güncellenir ve bu aşamalar iteratif olarak devam ettirilir. Elde edilen güncel çok yüzlü koni parametreleri ile bir önceki döngüdeki değerlerinin arasındaki fark çok küçük olduğunda döngü sonlanır ve elde edilmiş olan güncel parametreler koni tepe noktasını belirler. Koni tepe noktası L1 ve L2 normlu EPCC için ayrı olarak hesaplanmıştır. Genel koni tepe noktası tahmini Algoritma (2) olarak verilmiştir. Algoritmanın detayları ilerleyen bölümlerde ayrı ayrı ele alınarak incelenmiştir.

3.4.1 L1 normlu genişletilmiş çok yüzlü konik sınıflandırıcı (EPCC-L1-C)

3.4.1.1 Objektif fonksiyon

Çizelge 3.1: Objektif Fonksiyon Parametreleri

Parametre	Boyut	Tanım
i	$\mathbb{R}$	İteraston sayısı
$n_{\pm}$	$\mathbb{R}$	Pozitif ve negatif örnek sayısı
y_i	± 1	Sınıf etiketi
$C_{\pm}$	$\mathbb{R}$	Ağırlık çarpanı
ω, γ	$\mathbb{R}^d$	Ağırlık vektörleri
x_t	$\mathbb{R}^d$	Rastgele seçilmiş veri örneği
X_{pos}	$\mathbb{R}^d \times \mathbb{R}^{n_+}$	Pozitif örnekleri içeren matris
α	$\mathbb{R}^{n_+}$	Doğrusal kombinasyon katsayılarını içeren vektör
$c = X_{pos}\alpha$	$\mathbb{R}^d$	Koni tepe noktası
b	$\mathbb{R}$	ofset (sapma)
$\lvert u \rvert_1$	$[\lvert u_1 \rvert \ \lvert u_2 \rvert \ \cdots \ \lvert u_d \rvert]$	L1 norm bileşen yönlü modülü
$\lvert u \rvert_2$	$[u_1^2 \ u_2^2 \ \cdots \ u_d^2]$	L2 norm bileşen yönlü modülü

L1 normlu genişletilmiş çok yüzlü konik sınıflandırıcı için koni tepe noktası tahmini ile elde edilen yöntemi EPCC-L1-C olarak adlandıralım. EPCC-L1-C objektif fonksiyonu

Denklem (3.10) olarak verilmiştir ve F olarak belirtilmiştir.

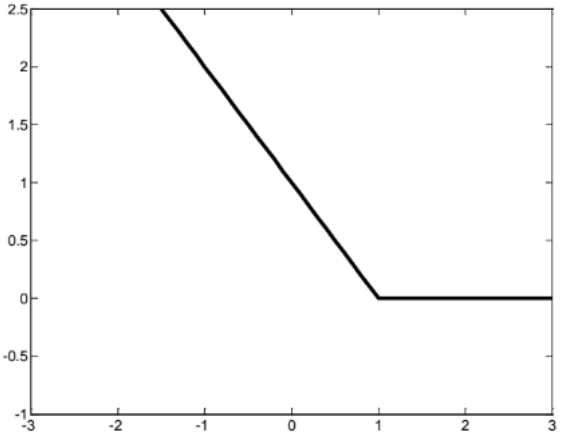
$$min_{\omega,b,\gamma,\alpha} \quad \frac{1}{2}\omega^T\omega + C_+ \sum_{i=1}^{n+} H(y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_1] + b - 1) + C_- \sum_{i=1}^{n-} H(y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_1] + b - 1) + \frac{1}{2}\alpha^T\alpha \tag{3.10}$$

Bu objektif fonksiyonda kullanılan tüm parametrelerin boyutları ve açıklamaları detaylı bir şekilde Çizelge (3.1)'de bulunmaktadır.

Koni tepe noktası objektif fonksiyon (3.10)'de $X_{pos}\alpha$ olarak verilmiştir ve c olarak adlandırılmaktadır. Koni tepe noktası (c) parametresinin tahmini için α vektörünün minimize edilmesi gerekmektedir.

3.4.1.2 Hinge kaybı

Algoritma (2)'ye göre önemli aşamalardan biri Hinge kayıp değerinin hesaplamasıdır. Makine öğreniminde Hinge kaybı *maksimum-boşluk* sınıflandırması için daha çok SVM alanında kullanılmaktadır (Rosasco vd., 2004). Bu çalışmada kullanılan Hinge kaybı Şekil (3.6) ve Denklem (3.11) olarak ifade edilmektedir (Cevikalp ve Elmas, 2016).



Şekil 3.6: Hinge kaybı (Cevikalp ve Elmas, 2016)

$$H(t) = max(0, 1-t) \tag{3.11}$$

Koni tepe noktası tahmini için kullanılan her iki L1 normlu ve L2 normlu EPCC objektif fonksiyonunda kullanılan Hinge kaybı için (3.12)'de verilen eşitsizlik her veri noktası için ayrı ayrı kontrol edilmektedir.

$$\begin{cases} y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_{1(2)}] < 1 & (1.durum) \\ y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_{1(2)}] > 1 & (2.durum) \end{cases} \tag{3.12}$$

1.durumda eğer Hinge kayıp bölümü 1 den küçük ise rastgele seçilmiş veri noktası ve diğer parametreler ile birlikte objektif fonksiyonuna türev işlemi uygulanarak algoritma devam etmektedir. Eğer Hinge kaybı 1'den büyük ise bu durumda fonksiyona devam edilemez ve başka bir veri noktasına geçilir.

1. DURUM: Hinge<1 durumu $y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_1] < 1$

3.4.1.3 EPCC-L1-C fonksiyon türevi

EPCC-L1-C fonksiyon türevi için öncelikle objektif fonksiyonunun (3.10) en önemli bölümü olan $X_{pos}\alpha$ çarpımını ele alarak başlayalım.

$$X_{pos}\alpha = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{n_{pos}1} \\ x_{12} & x_{22} & \cdots & x_{n_{pos}2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1d} & x_{2d} & \cdots & x_{n_{pos}d} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_{pos}} \end{bmatrix} = \begin{bmatrix} x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}} \\ x_{12}\alpha_1 + x_{22}\alpha_2 + \cdots + x_{n_{pos}2}\alpha_{n_{pos}} \\ \vdots \\ x_{1d}\alpha_1 + x_{2d}\alpha_2 + \cdots + x_{n_{pos}d}\alpha_{n_{pos}} \end{bmatrix}$$

Bir sonraki adım, rastgele seçilmiş x_t veri vektörünü $X_{pos}\alpha$ çarpım vektöründen çıkarma işlemidir. Elde edilen bu vektör objektif fonksiyonun L1 norm terimini oluşturur ve L1 norm türevinin uygulanması gereken bölümdür.

$$f(\alpha) = \begin{bmatrix} x_{t1} \\ x_{t2} \\ \vdots \\ x_{td} \end{bmatrix} - \begin{bmatrix} x_{11}\alpha_1 + x_{12}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}} \\ x_{21}\alpha_1 + x_{22}\alpha_2 + \cdots + x_{n_{pos}2}\alpha_{n_{pos}} \\ \vdots \\ x_{1d}\alpha_1 + x_{2d}\alpha_2 + \cdots + x_{n_{pos}d}\alpha_{n_{pos}} \end{bmatrix} = \begin{bmatrix} x_{t1} - x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}} \\ x_{t2} - x_{12}\alpha_1 + x_{22}\alpha_2 + \cdots + x_{n_{pos}2}\alpha_{n_{pos}} \\ \vdots \\ x_{td} - x_{1d}\alpha_1 + x_{2d}\alpha_2 + \cdots + x_{n_{pos}d}\alpha_{n_{pos}} \end{bmatrix}$$

Genel olarak L1 norm bölümünün türev tanımı Denklem (3.13) gibi belirtilmektedir. Burada (.*) çarpımı MATLAB programında olduğu gibi iki vektörün her bir teriminin çarpımını belirtmektedir.

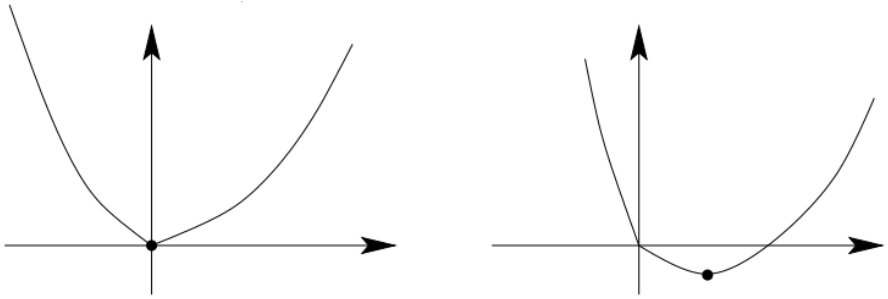
$$|x_t - X_{pos}\alpha|_1' = sign(x_t - X_{pos}\alpha). * (x_t - X_{pos}\alpha)' \tag{3.13}$$

L1 norm türevi işaret belirleme için (Bach ve Jenatton, 2011) çalışmasında verilen yöntem kullanılmıştır. Bu çalışmada örnek olarak verilen $min\ \frac{1}{2}x^2 - xy + \gamma |x|$ karesel problemde

0 noktasında türev için (Şekil 3.7);

$$0_+ : \ g_+ = \gamma - y \qquad 0_- : \ g_- = -\gamma - y$$

Bu durumda;



Şekil 3.7: 0 noktasında bükülme olan parçalı karesel fonksiyon

Eğer $g_+ \geq 0$ ve $g_- \leq 0$ ise, $x = 0$ çözüm olabilmektedir $(|y| \leq \gamma)$
Eğer $g_+ \leq 0$ ise, $x \geq 0$ çözüm olabilmektedir $(y \geq \gamma) \rightarrow \ \ x^* = y - \gamma$
Eğer $g_- \leq 0$ ise, $x \leq 0$ çözüm olabilmektedir $(y \leq -\gamma) \rightarrow \ \ x^* = y + \gamma$

Objektif fonksiyonu (3.10)'deki L1 norm türev bölümünü (Bach ve Jenatton, 2011)'daki yönteme göre uygulanırsa elde edilen genel işaretlenme aşağıdaki gibi olmaktadır. Burada, $X_{pos}(i,:)$ terimi X_{pos} matrisinin i'nci satır vektörüdür.

$$f(\alpha) = \begin{cases} +(x_{ti} - x_{1i}\alpha_1 - x_{2i}\alpha_2 - \cdots - x_{n_{pos}i}\alpha_{n_{pos}}) & eger \ \ x_{ti} - X_{pos}(i,:)\alpha > 0 \\ -(x_{ti} - x_{1i}\alpha_1 - x_{2i}\alpha_2 - \cdots - x_{n_{pos}i}\alpha_{n_{pos}}) & eger \ \ x_{ti} - X_{pos}(i,:)\alpha < 0 \end{cases}$$

Bu durumda f bölümü için elde edilen türev:

$$\frac{\partial f}{\partial \alpha} = \begin{cases} +(X_{pos}(i,:)) & eger \ \ x_{ti} - X_{pos}(i,:)\alpha > 0 \\ -(X_{pos}(i,:)) & eger \ \ x_{ti} - X_{pos}(i,:)\alpha < 0 \end{cases}$$

Objektif fonksiyonundaki diğer terimlerin türevi:

$$\frac{\partial}{\partial \alpha}(\frac{1}{2}\omega^T\omega) = 0$$
$$\frac{\partial}{\partial \alpha}(b) = 0$$
$$\frac{\partial}{\partial \alpha}(\frac{1}{2}\alpha^T\alpha) = \alpha$$
$$\frac{\partial}{\partial \alpha}(\omega^T(x_t - X_{pos}\alpha)) = -X_{pos}^T\omega$$

Genel fonksiyon türevi *1.durum (Hinge<1)* için:

Diğer terimlerin türevi ile birlikte EPCC-L1-C objektif fonksiyonunun (3.10) genel türevi rastgele seçilmiş veri vektörü x_t için aşağıdaki gibi olacaktır:

$$\nabla F = \begin{cases} C_+ \sum_{i=1}^{n_+} y_i[-X_{pos}^T\omega - \gamma^T(X_{pos}(i,:))]+ \\ C_- \sum_{i=1}^{n_-} y_i[-X_{pos}^T\omega - \gamma^T(X_{pos}(i,:))] + \alpha \qquad eger \;\; x_{ti} - X_{pos}(i,:)\alpha > 0 \\ \\ C_+ \sum_{i=1}^{n_+} y_i[-X_{pos}^T\omega + \gamma^T(X_{pos}(i,:))]+ \\ C_- \sum_{i=1}^{n_-} y_i[-X_{pos}^T\omega + \gamma^T(X_{pos}(i,:))] + \alpha \qquad eger \;\; x_{ti} - X_{pos}(i,:)\alpha < 0 \end{cases}$$

3.4.1.4 Stochastic Gradient Descent yöntemi

Türev işleminden sonraki adım Stochastic Gradient Descent adımıdır. Koni tepe noktası ($c = X_{pos}\alpha$) tahmini için objektif fonksiyonun α'ya göre türevi alındıktan sonra α'yı minimize etmek gerekir. Bu minimizasyon işlemi için Stochastic Gradient Descent yöntemi kullanılır ve bu yöntem doğrusal SVM gibi bir çok makine öğrenimi alanında geniş aralıktaki modelleri eğitmek için kullanılan popüler bir algoritmadır. Stochastic Gradient Descent (SGD) türevlenebilir fonksiyonlar toplamından oluşan objektif fonksiyonu minimize etmek için kullanılan bir gradyent düşme optimizasyonun rastgele yaklaşım yöntemidir. Yani, SGD yöntemi iterasyon ile minimum veya maksimumları bulmaya çalışmaktadır.

Koni tepe noktası tahmini için kullanılan Stochastics Gradient Descent yöntemi ise Denklem (3.14) olarak verilmiştir. ϵ için sıfıra yakın küçük bir değer kullanılmıştır. $steplength(s) = 0.000001$ ve $t = 750$ ise rastgele seçilmiş veri noktalarının döngü sayısı olarak alınmıştır.

$$\begin{aligned} \alpha_{yeni} &= \alpha_{eski} - \epsilon\nabla F \\ \epsilon &= steplength/t \end{aligned} \tag{3.14}$$

2. DURUM: Hinge>1 durumu; $y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T\,|x_t - X_{pos}\alpha|_1] > 1$

Bu durumda eğer Hinge 1'den büyük ise, fonksiyon türevi;

Genel fonksiyon türevi *2.durum (Hinge>1)* için;

$$\nabla F = \alpha/n$$

$$\alpha_{yeni} = \alpha_{eski} - \epsilon\nabla F$$

3.4.2 L2 normlu genişletilmiş çok yüzlü konik sınıflandırıcı (EPCC-L2-C)

3.4.2.1 Objektif fonksiyon

L2 normlu genişletilmiş çok yüzlü konik sınıflandırıcı için koni tepe noktası tahmini ile elde edilen yöntemi EPCC-L2-C olarak adlandırılalım. EPCC-L2-C objektif fonksiyonu Denklem (3.15)'de verilmiştir ve F olarak belirtilmiştir. Her parametre hakkındaki bilgilerin detaylı açıklaması Çizelge (3.1) olarak daha önce verilmiştir.

$$min_{\omega,b,\gamma,\alpha} \quad \frac{1}{2}\omega^T\omega + C_+ \sum_{i=1}^{n+} H(y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_2] + b - 1) + \\ C_- \sum_{i=1}^{n-} H(y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_2] + b - 1) + \frac{1}{2}\alpha^T\alpha \tag{3.15}$$

1. DURUM: Hinge<1 durumu $y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_2] < 1$

3.4.2.2 EPCC-L2-C fonksiyon türevi

EPCC-L2-C objektif fonksiyonu olarak verilen Denklem (3.15) türevi için fonksiyonu belirli parçalara bölerek tek tek inceleyelim. Öncelikle L2 norm uygulanması gereken $|x_t - X_{pos}\alpha|_2$ bölüm türevi ile başlayalım.

$X_{pos}\alpha$ çarpımınından elde edilen matris:

$$X_{pos}\alpha = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{n_{pos}1} \\ x_{12} & x_{22} & \cdots & x_{n_{pos}2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1d} & x_{2d} & \cdots & x_{n_{pos}d} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_{pos}} \end{bmatrix} = \begin{bmatrix} x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}} \\ x_{12}\alpha_1 + x_{22}\alpha_2 + \cdots + x_{n_{pos}2}\alpha_{n_{pos}} \\ \vdots \\ x_{1d}\alpha_1 + x_{2d}\alpha_2 + \cdots + x_{n_{pos}d}\alpha_{n_{pos}} \end{bmatrix}$$

L2 norm bölümünü elde etmek için rastgele seçilmiş veri vektörü x_t'den $X_{pos}\alpha$ çarpım vektörünü çıkararak karesini alalım. Bu işlemlerden sonra türev hesaplanabilir.

Çıkarma işlemi:

$$\begin{bmatrix} x_{t1} \\ x_{t2} \\ \vdots \\ x_{td} \end{bmatrix} - \begin{bmatrix} x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}} \\ x_{12}\alpha_1 + x_{22}\alpha_2 + \cdots + x_{n_{pos}2}\alpha_{n_{pos}} \\ \vdots \\ x_{1d}\alpha_1 + x_{2d}\alpha_2 + \cdots + x_{n_{pos}d}\alpha_{n_{pos}} \end{bmatrix} = \begin{bmatrix} x_{t1} - (x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}}) \\ x_{t2} - (x_{12}\alpha_1 + x_{22}\alpha_2 + \cdots + x_{n_{pos}2}\alpha_{n_{pos}}) \\ \vdots \\ x_{td} - (x_{1d}\alpha_1 + x_{2d}\alpha_2 + \cdots + x_{n_{pos}d}\alpha_{n_{pos}}) \end{bmatrix}$$

Kare işlemi:

$$\begin{bmatrix} x_{t1} - (x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}}) \\ x_{t2} - (x_{12}\alpha_1 + x_{22}\alpha_2 + \cdots + x_{n_{pos}2}\alpha_{n_{pos}}) \\ \vdots \\ x_{td} - (x_{1d}\alpha_1 + x_{2d}\alpha_2 + \cdots + x_{n_{pos}d}\alpha_{n_{pos}}) \end{bmatrix}^2 = \begin{bmatrix} (x_{t1} - (x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}}))^2 \\ (x_{t2} - (x_{12}\alpha_1 + x_{22}\alpha_2 + \cdots + x_{n_{pos}2}\alpha_{n_{pos}}))^2 \\ \vdots \\ (x_{td} - (x_{1d}\alpha_1 + x_{2d}\alpha_2 + \cdots + x_{n_{pos}d}\alpha_{n_{pos}}))^2 \end{bmatrix}$$

Kare işlemi için elde edilen vektörün ilk satırının karesini detaylı olarak inceleyelim.

$$\begin{aligned} f = (x_{t1} - (x_{11}\alpha_1 + \cdots + x_{1n_{pos}}\alpha_{n_{pos}}))^2 &= x_{t1}^2 - 2(x_{t1})(x_{11}\alpha_1 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}}) + \\ &\quad [x_{11}\alpha_1 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}}]^2 \\ = x_{t1}^2 + (-2x_{t1}x_{11}\alpha_1 - \cdots - 2x_{t1}x_{n_{pos}1}\alpha_{n_{pos}}) &+ [(x_{11}^2\alpha_1^2 + \cdots + x_{11}\alpha_1 x_{n_{pos}1}\alpha_{n_{pos}}) + \\ &\quad \cdots + (x_{n_{pos}1}\alpha_{n_{pos}}x_{11}\alpha_1 + \cdots + x_{n_{pos}1}^2\alpha_{n_{pos}}^2)] \end{aligned}$$

Türev işlemi:

Elde edilen ilk satır karesinin α_1 ve α_2 için ayrı bir şekilde türevini alarak sonuçları karşılaştıralım. Bu sonuca göre türev için genel bir fonksiyon elde etmeye çalışalım.

İlk satır karesinin α_1 için türevi,

$$\begin{aligned} \frac{\partial f}{\partial \alpha_1} &= -2x_{t1}x_{11} + 2x_{11}^2\alpha_1 + x_{11}x_{21}\alpha_2 + x_{11}x_{31}\alpha_3 + \cdots + x_{11}x_{n_{pos}1}\alpha_{n_{pos}} + \\ &\quad x_{11}x_{21}\alpha_2 + x_{11}x_{31}\alpha_3 + \cdots + x_{11}x_{n_{pos}1}\alpha_{n_{pos}} \\ &= -2x_{t1}x_{11} + 2x_{11}(x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}}) \\ &= 2x_{11}(-x_{t1} + (x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}})) \\ &= 2x_{11}(-x_{t1} + X_{pos}(1,:)\alpha) \end{aligned}$$

İlk satır karesinin α_2 için türevi,

$$\begin{aligned} \frac{\partial f}{\partial \alpha_2} &= -2x_{t1}x_{21} + x_{21}x_{11}\alpha_1 + 2x_{21}^2\alpha_2 + x_{21}x_{11}\alpha_1 + x_{21}x_{31}\alpha_3 \cdots + x_{21}x_{1n_{pos}}\alpha_{n_{pos}} + \\ &\quad x_{21}x_{31}\alpha_3 + \cdots + x_{21}x_{1n_{pos}}\alpha_{n_{pos}} \\ &= -2x_{t1}x_{21} + 2x_{21}(x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}}) \\ &= 2x_{21}(-x_{t1} + (x_{11}\alpha_1 + x_{21}\alpha_2 + \cdots + x_{n_{pos}1}\alpha_{n_{pos}})) \\ &= 2x_{21}(-x_{t1} + X_{pos}(1,:)\alpha) \end{aligned}$$

Bu durumda, ilk satır karesinin α_i için türevi ise,

$$\frac{\partial f}{\partial \alpha_i} = 2x_{i1}(-x_{t1} + X_{pos}(1,:)\alpha)$$

Bu duruma göre L2 normlu bölüm için elde edilen türev aşağıda verilmiştir ve K vektörü olarak adlandırılmıştır.

$$K = (|x_t - X_{pos}\alpha|_2)'$$

$$K = \begin{bmatrix} [2x_{11}(X_{pos}(1,:)\alpha - x_{t1})] + \cdots + [2x_{1d}(X_{pos}(d,:)\alpha - x_{td})] \\ [2x_{21}(X_{pos}(1,:)\alpha - x_{t1})] + \cdots + [2x_{2d}(X_{pos}(d,:)\alpha - x_{td})] \\ \vdots \\ [2x_{n_{pos}1}(X_{pos}(1,:)\alpha - x_{t1})] + \cdots + [2x_{n_{pos}d}(X_{pos}(d,:)\alpha - x_{td})] \end{bmatrix}$$

$$K = 2 \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n_{pos}1} & x_{n_{pos}2} & \cdots & x_{n_{pos}d} \end{bmatrix} \begin{bmatrix} X_{pos}(1,:)\alpha - x_{t1} \\ X_{pos}(2,:)\alpha - x_{t2} \\ \vdots \\ X_{pos}(d,:)\alpha - x_{td} \end{bmatrix}$$

$$K = 2X_{pos}^T(X_{pos}\alpha - x_t)$$

Diğer terimler türevi:

$$\frac{\partial}{\partial\alpha}(\frac{1}{2}\omega^T\omega) = 0$$
$$\frac{\partial}{\partial\alpha}(b) = 0$$
$$\frac{\partial}{\partial\alpha}(\frac{1}{2}\alpha^T\alpha) = \alpha$$
$$\frac{\partial}{\partial\alpha}(\omega^T(x_t - X_{pos}\alpha)) = -X_{pos}^T\omega$$

Genel fonksiyon türevi *1.durum (Hinge<1)* için:

Denklem (3.15)'ndeki tüm terimler ayrı ayrı türevlendikten sonra birleştirilerek genel fonksyion türevi Denklem (3.16) olarak elde edilmektedir.

$$\nabla F = C_+ \sum_{i=1}^{n+}(y_i[-X_{pos}^T\omega + \gamma^T(K)]) + C_- \sum_{i=1}^{n-}(y_i[-X_{pos}^T\omega + \gamma^T(K)]) + \alpha \tag{3.16}$$

Türev işleminden sonra EPCC-L1-C bölümündeki gibi α_{yeni} için Stochastic Gradient Descent yöntemi kullanılarak α yenilenir.

$$\alpha_{yeni} = \alpha_{eski} - \epsilon \nabla F$$
$$\epsilon = steplength/t$$
$$steplength = 0.000001$$
$$t = 750$$

2. DURUM: Hinge>1 durumu; $y_i[\omega^T(x_t - X_{pos}\alpha) + \gamma^T |x_t - X_{pos}\alpha|_2] > 1$ Bu durumda eğer Hinge 1'den büyük ise, fonksiyon türevi;

Genel fonksiyon türevi *2.durum (Hinge>1)* için;

$$\nabla F = \alpha/n$$
$$\alpha_{yeni} = \alpha_{eski} - \epsilon \nabla F$$

Hinge kaybı daha önce Denklem (3.12)'de belirtildiği gibi rastgele seçilen test noktasına göre fonksiyonu etkilemektedir.

EPCC-L1-C ve EPCC-L2-C yöntemleri *UCI Machine Learning Repository*'deki çeşitli veri tabanlarına uygulanarak SVM, PCC-L1, PCC-L2, EPCC-L1 ve EPCC-L2 gibi diğer yöntemler ile karşılaştırılmıştır. Sonuçlar **Bulgular ve Tartışma** bölümünde yer almaktadır.

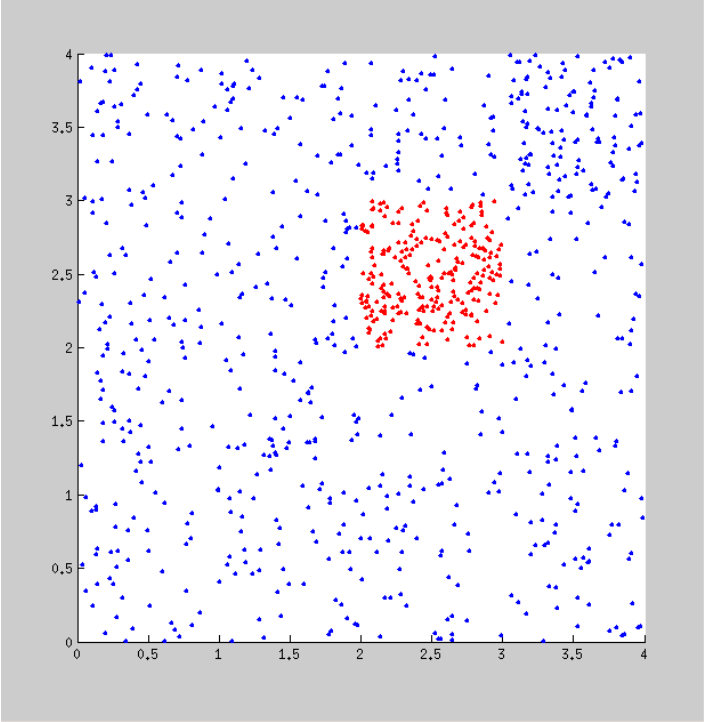
4. BULGULAR VE TARTIŞMA

Koni tepe noktası tahmini için önerilen çok yüzlü konik sınıflandırıcı EPCC-L1-C ve EPCC-L2-C yöntemleri sentetik ve gerçek veri tabanlarına uygulanmıştır. Bu bölümde öncelikle sentetik veri tabanındaki sonuçlar demo şeklinde veri kümesinin koordinat değerlerinde belirtilmiştir. Gerçek deney veri tabanlarından elde edilen sonuçlar ise SVM, L1 ve L2 normlu PCC ve EPCC sınıflandırıcıları ile karşılaştırılarak çizelge olarak verilmiştir. L1 ve L2 normlu PCC ve EPCC yöntemlerinde koni tepe noktası olarak pozitif verilerin ortalaması alınırken EPCC-L1-C ve EPCC-L2-C yöntemlerinde koni tepe noktası tahmini yapılmıştır. Ayrıca, eğitim ve test kümeleri 10 kat çapraz geçerlilik (10-fold cross validation) olarak oluşturulmuştur. Başlangıçta α için pozitif verilerin ortalaması referans alınmıştır. Daha iyi sonuçlar elde etmek için, SVM parametresi (C) 10 ve 50 gibi iki farklı değer ile deney tekrarlanmıştır ve sonuçlar çizelge olarak verilmiştir.

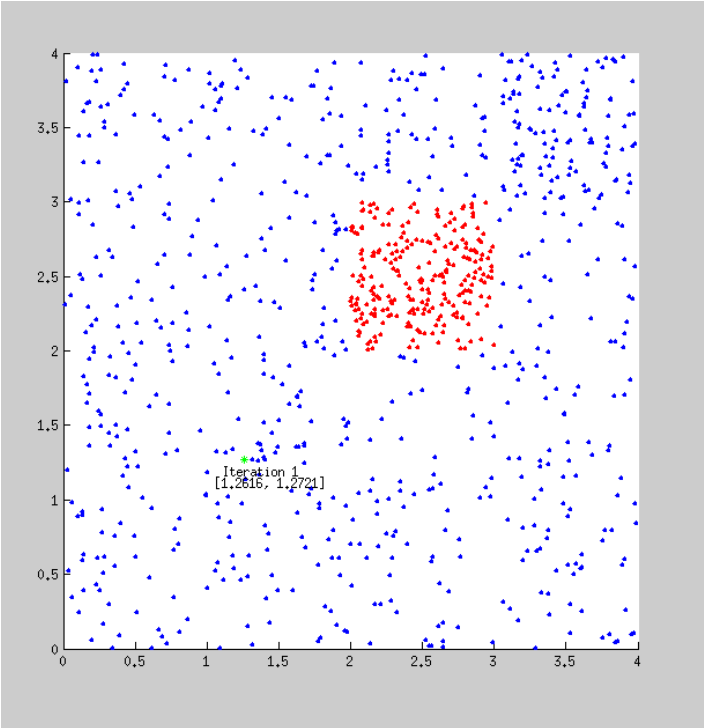
4.1 Sentetik Deney ve Demo

Önerilen yöntemleri gerçek veri tabanlarına uygulamadan önce sentetik bir veri kümesi oluşturularak denenmiştir. Şekil (4.1)'deki gibi pozitif örneklerin kare şeklinde ortada ve negatif örneklerin ise pozitiflerin etrafında rastgele dağılacak şekilde 2 boyutlu bir sentetik veri kümesi oluşturulmuştur. Bu veri kümesi rastgele dağılmış 250 pozitif ve 700 negatif örnekten oluşmaktadır.

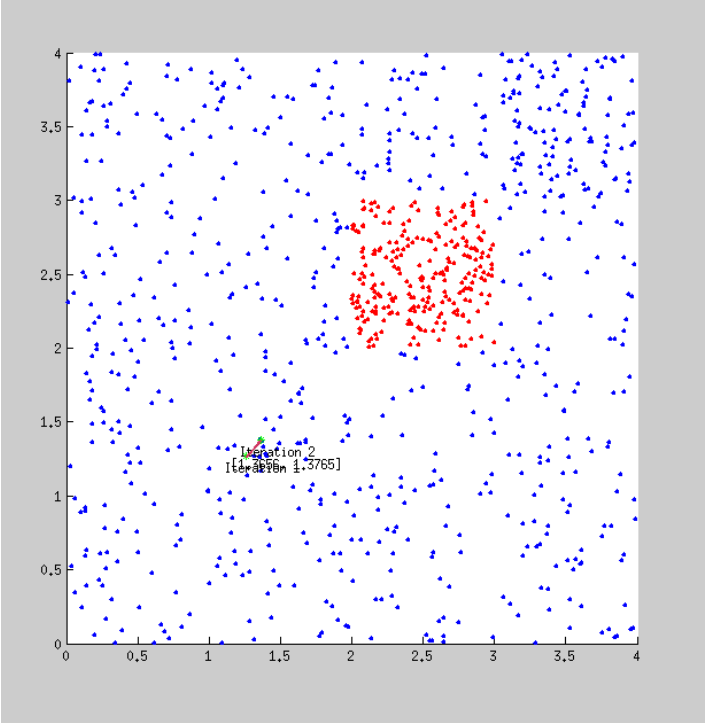
Sentetik veri kümesi, çok yüzlü konik sınıflandırıcının koni tepe noktası tahmin yöntemini uygulamak ve ayrıca PCC ve EPCC yöntemleri ile olan benzerliklerini karşılaştırmak için kullanılmıştır. PCC ve EPCC gibi yöntemlerde çok yüzlü sınıflandırıcı oluşturmak için koni tepe noktası pozitif verilerin ortalaması olarak alınmıştır. Sentetik veri kümesinde pozitif veriler her iki x ve y koordinatında 2-3 koordinat değerleri arasında rastegele dağılmıştır ve ortalaması ise $(x_{mean}, y_{mean}) = [2.5\ 2.5]$'dır. Koni tepe noktası tahmini yapılarak elde edilen çok yüzlü konik sınıflandırıcının bu veri kümesine uygulanması ile birlikte beklenen sonuç her iterasyon sonucunda elde edilen değerin giderek pozitif verilerin ortalamasına yaklaşmasıdır. Her iteraston sonucunda elde edilen değer ve bir önceki iterasyon sonucu ile arasındaki fark ok olarak şekil üzerinde belirtilmiştir. Rastegele dağılan veriler ile elde edilen sentetik veri kümesine ilk olarak EPCC-L1-C yöntemi uygulanmıştır. Şekil (4.2) ile Şekil (4.7) arasındaki tüm şekiller EPCC-L1-C yöntem sonucunu detaylı olarak göstermektedir.



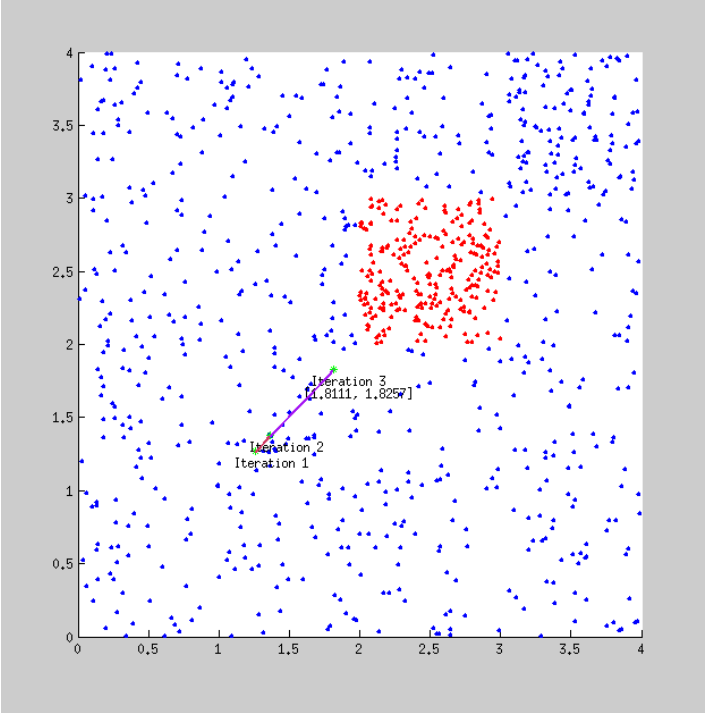
Şekil 4.1: EPCC-L1-C yöntemi için kullanılan pozitif (kırmızı) ve negatif (mavi) örneklerden oluşan 2 boyutlu sentetik veri



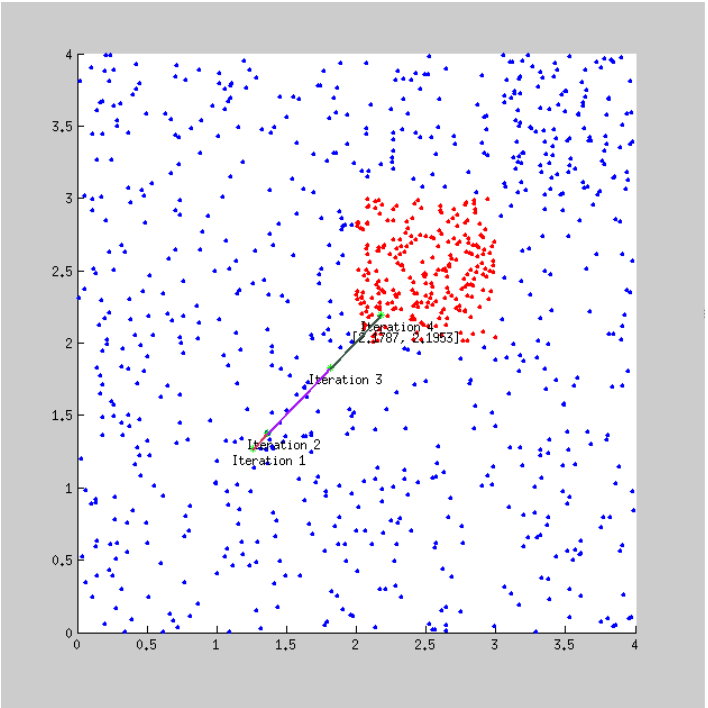
Şekil 4.2: İterasyon-1 sonucunda elde edilen koordinat değeri



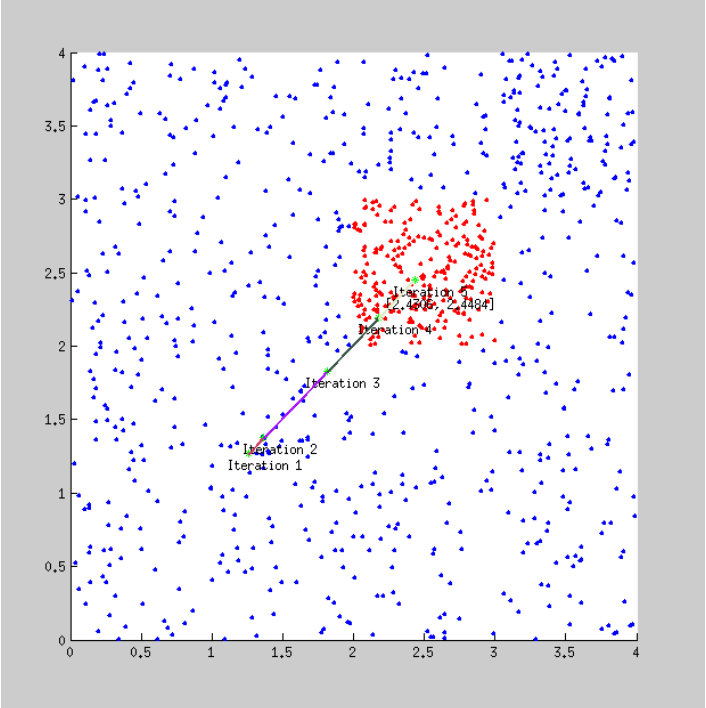
Şekil 4.3: İterasyon-2 sonucunda elde edilen değer ve İterasyon-1 ile arasındaki fark



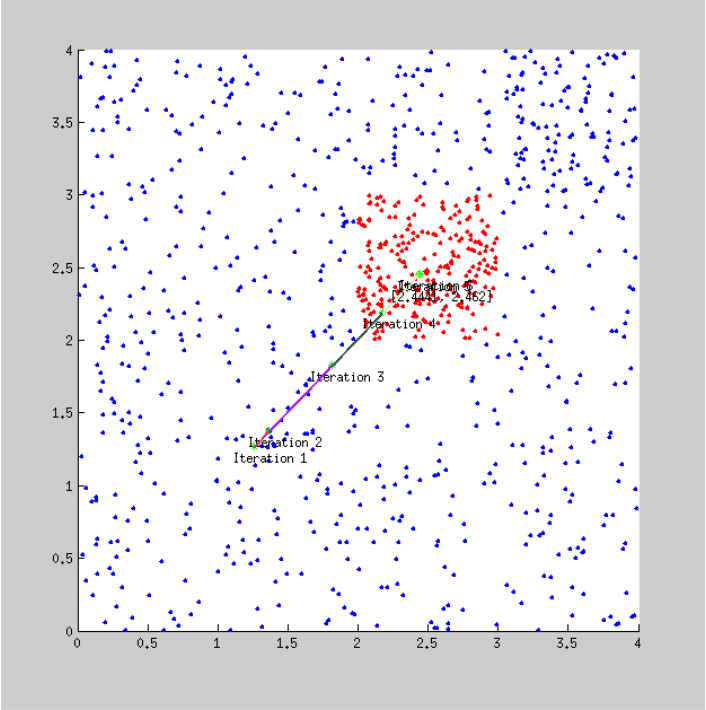
Şekil 4.4: İterasyon-3 sonucunda elde edilen değer ve İterasyon-2 ile arasındaki fark



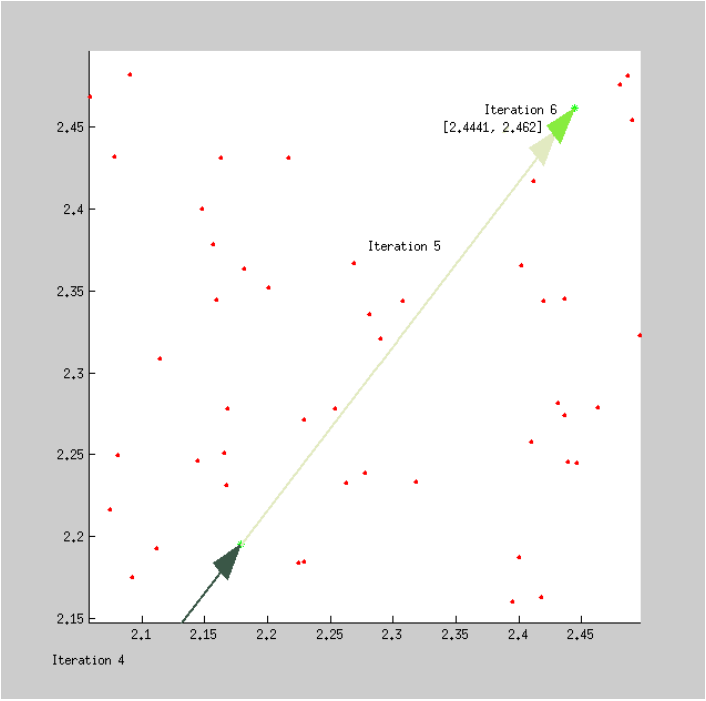
Şekil 4.5: İterasyon-4 sonucunda elde edilen değer ve İterasyon-3 ile arasındaki fark



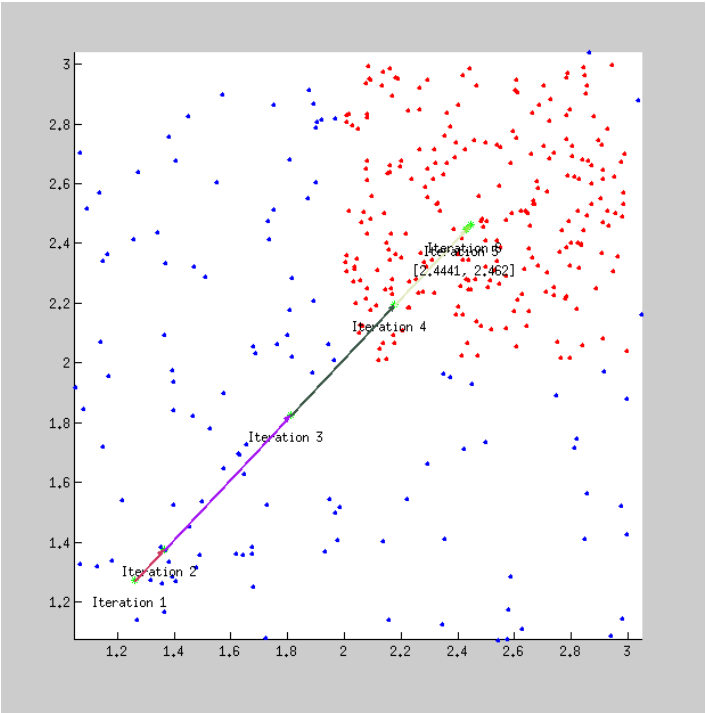
Şekil 4.6: İterasyon-5 sonucunda elde edilen değer ve İterasyon-4 ile arasındaki fark



Şekil 4.7: İterasyon-6 sonucunda elde edilen değer ve İterasyon-5 ile arasındaki fark



Şekil 4.8: Yakınlaştırılmış İterasyon-5 ve İterasyon-6

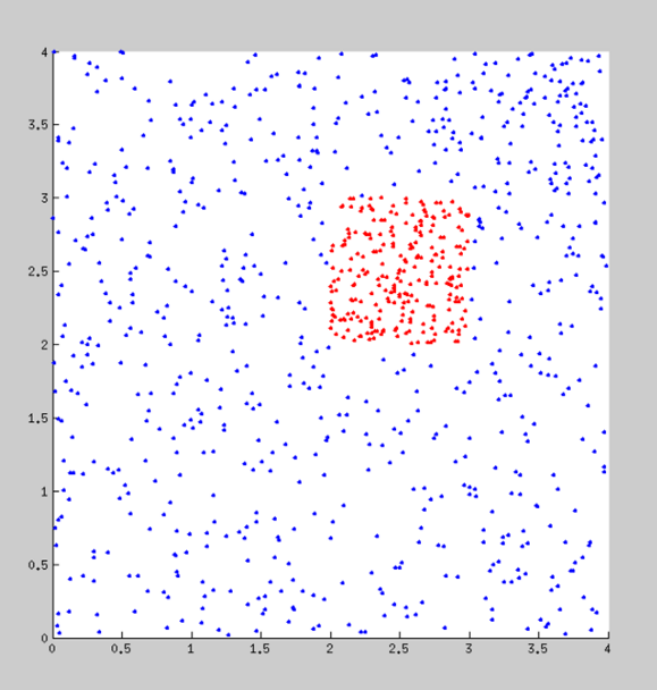


Şekil 4.9: Yakınlaştırılmış tüm iterasyonlar

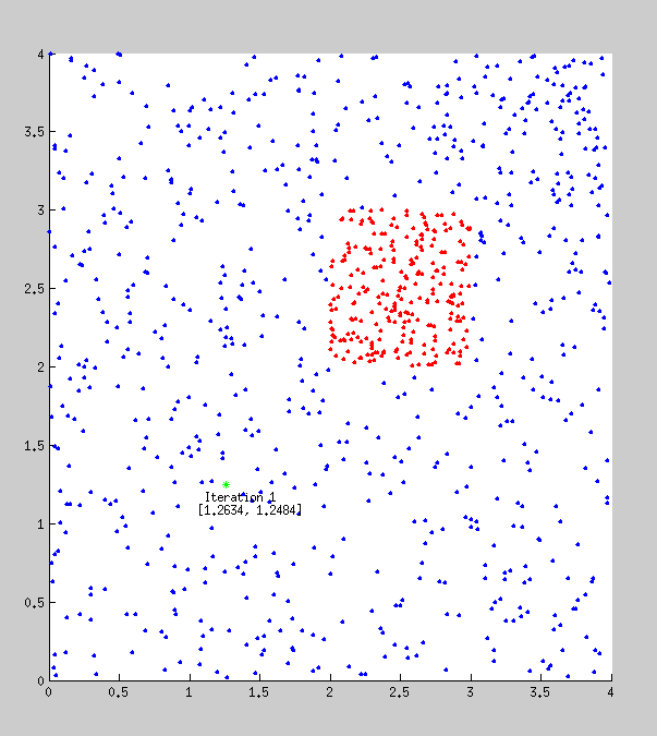
Sentetik veri kümesine uygulanan ilk çok yüzlü konik sınıflandırıcı modeli olan EPCC-L1-C için 6 iterasyon sonucunda elde edilen koordinat değerleri $(x_{L1}, y_{L1}) = [2.4441 \;\; 2.4620]$ olmuştur. Bu sonucun 6 iterasyon sonunda elde edilmesinin nedeni ise $\|\omega - \omega_{old}\| < 0.5$ olduğunda sonuç değerinde çok önemli bir farklılık oluşturmadığından iterasyon döngüsünü durdurmasından kaynaklanmaktadır. Veriler rastgele dağıldığından oluşturulan bir başka veri kümesinde ise döngü farklı iterasyon sayısında durabilmektedir.

Tekrar rastgele dağılan başka bir sentetik veri kümesi elde edilmiştir (Şekil 4.10) ve EPCC-L2-C yöntemi uygulanmıştır. Şekil (4.11) ve Şekil (4.20) arasındaki tüm şekiller EPCC-L2-C yöntem sonucunu her iterasyon için detaylı olarak göstermektedir.

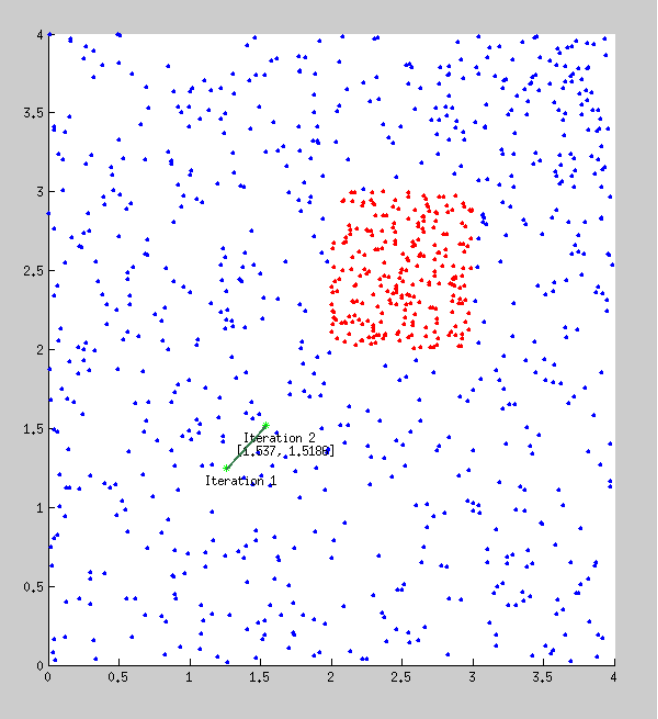
EPCC-L2-C yöntemi ile 10 iterasyon sonucunda ulaşılan koordinat değeri $(x_{L2}, y_{L2}) = [2.3009 \;\; 2.2735]$ olmuştur. Bu durumda, EPCC-L1-C ve EPCC-L2-C yöntemleri karşılaştırıldığında her iki yöntem ile elde edilen koordinat değerleri hedef koordinat değerine $(x_{mean}, y_{mean}) = [2.5 \;\; 2.5]$ yakın olduğu gözlemlenmiştir. Fakat, EPCC-L1-C yöntemi EPCC-L2-C yöntemine göre hedef koordinatına daha yakın sonuç elde etmiştir.



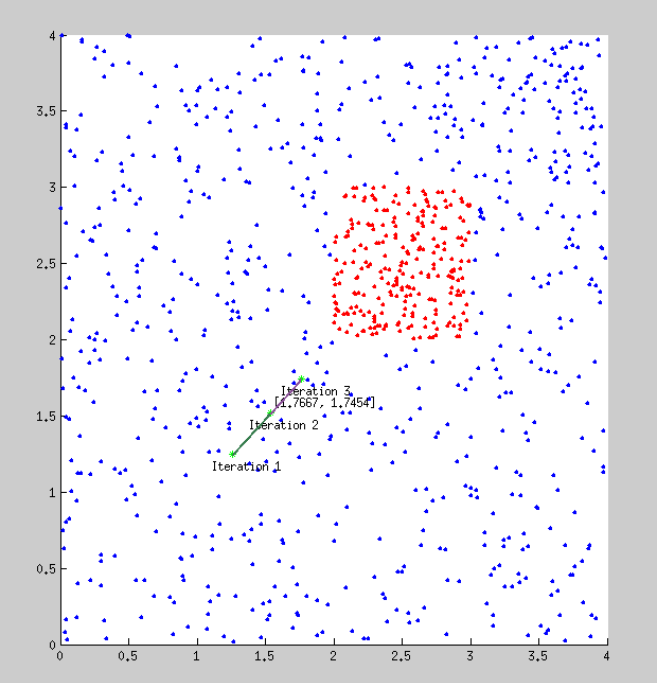
Şekil 4.10: EPCC-L2-C yöntemi için pozitif (kırmızı) ve negatif (mavi) örneklerden oluşan 2 boyutlu sentetik veri



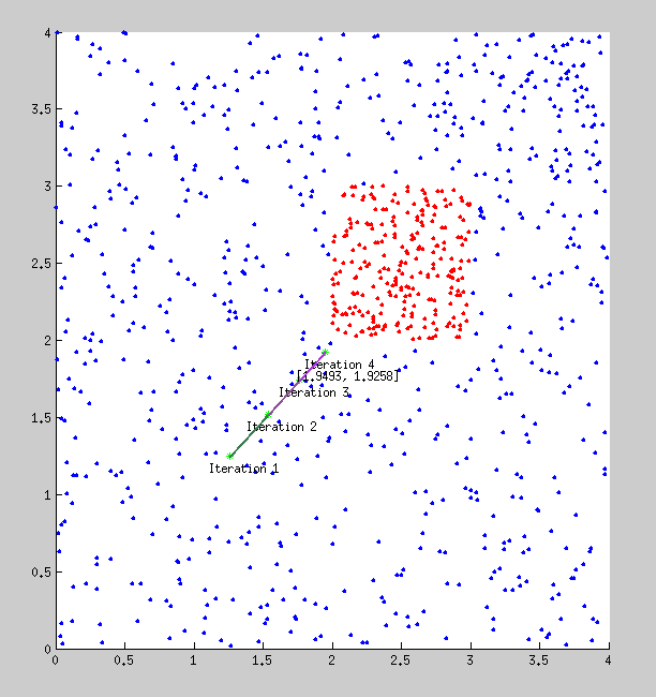
Şekil 4.11: İterasyon-1 sonucunda elde edilen koordinat değeri



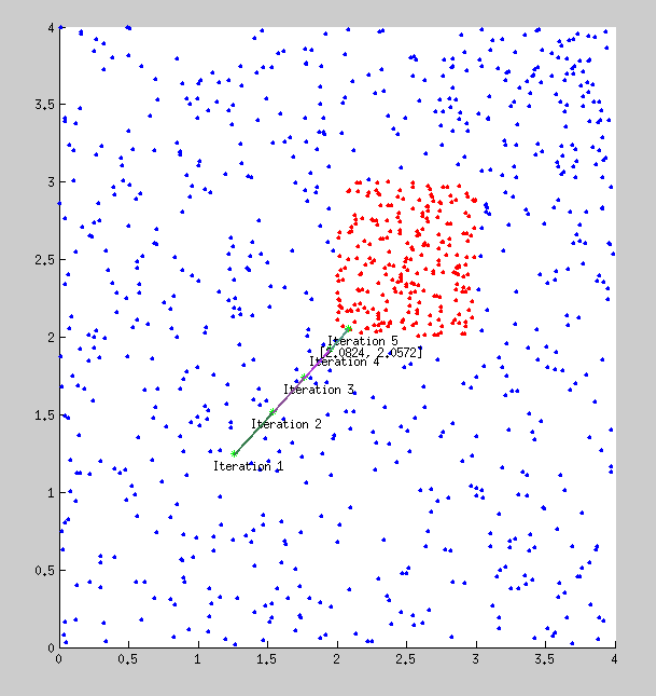
Şekil 4.12: İterasyon-2 sonucunda elde edilen değer ve İterasyon-1 ile arasındaki fark



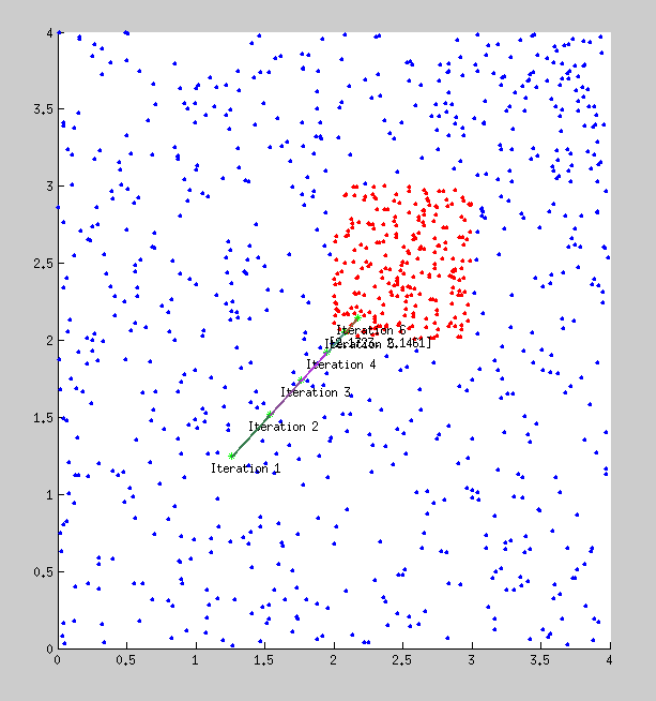
Şekil 4.13: İterasyon-3 sonucunda elde edilen değer ve İterasyon-2 ile arasındaki fark



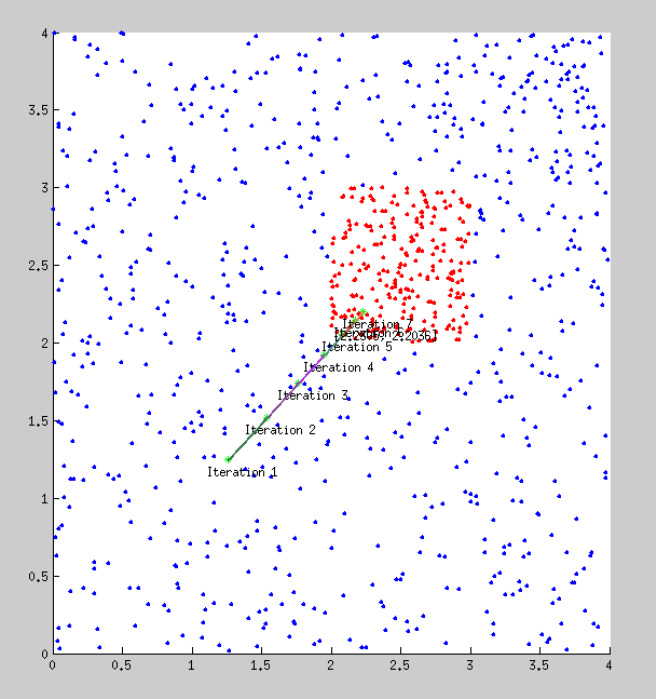
Şekil 4.14: İterasyon-4 sonucunda elde edilen değer ve İterasyon-3 ile arasındaki fark



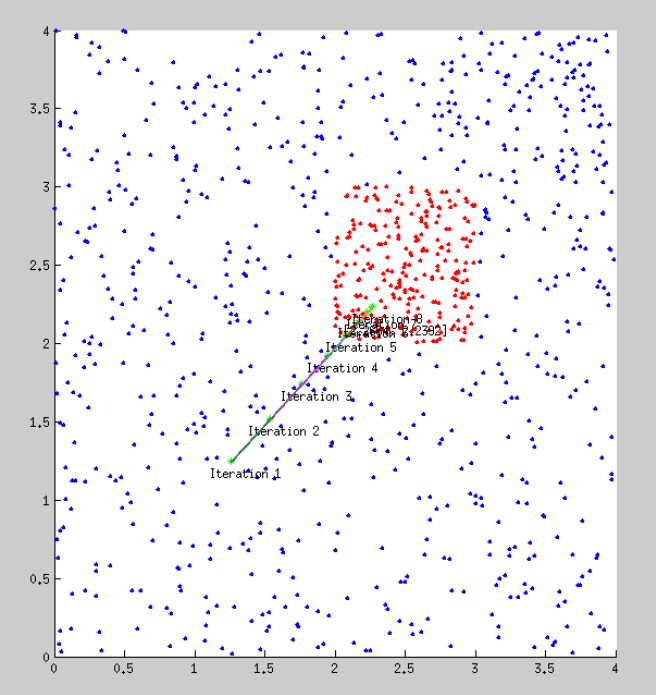
Şekil 4.15: İterasyon-5 sonucunda elde edilen değer ve İterasyon-4 ile arasındaki fark



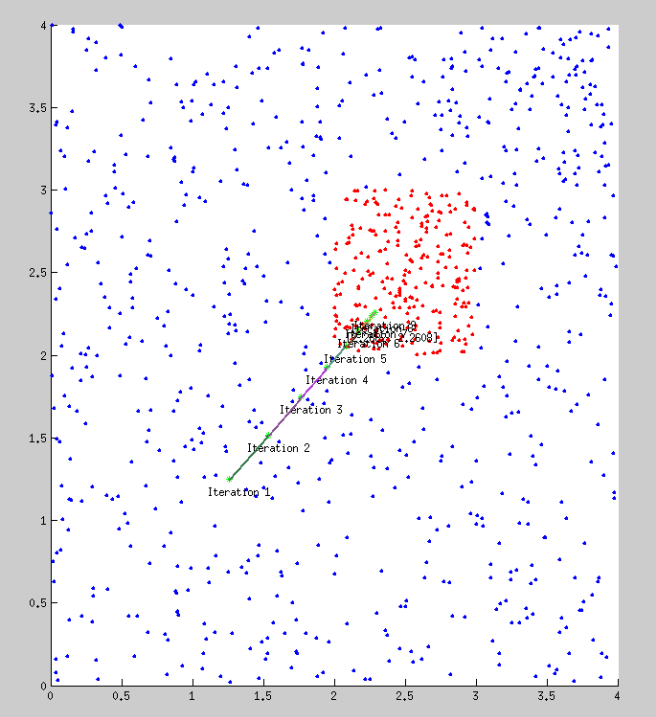
Şekil 4.16: İterasyon-6 sonucunda elde edilen değer ve İterasyon-5 ile arasındaki fark



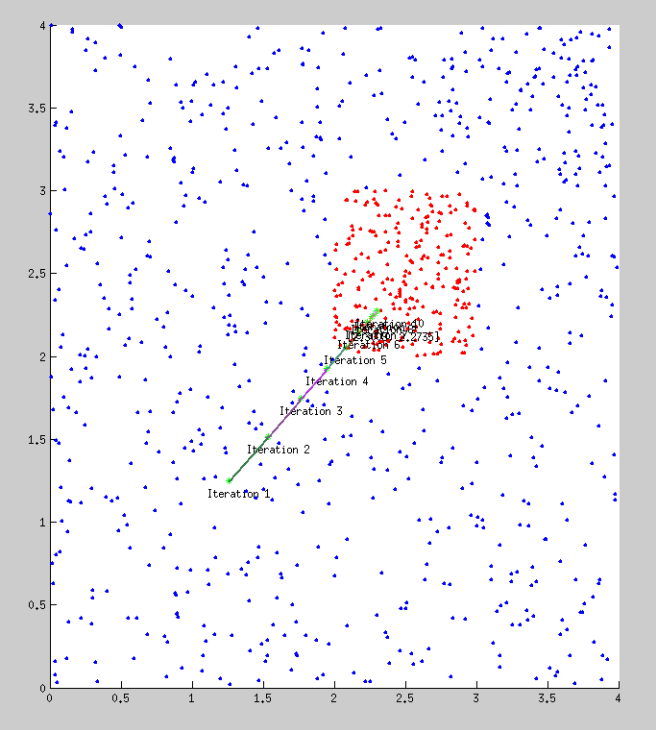
Şekil 4.17: İterasyon-7 sonucunda elde edilen değer ve İterasyon-6 ile arasındaki fark



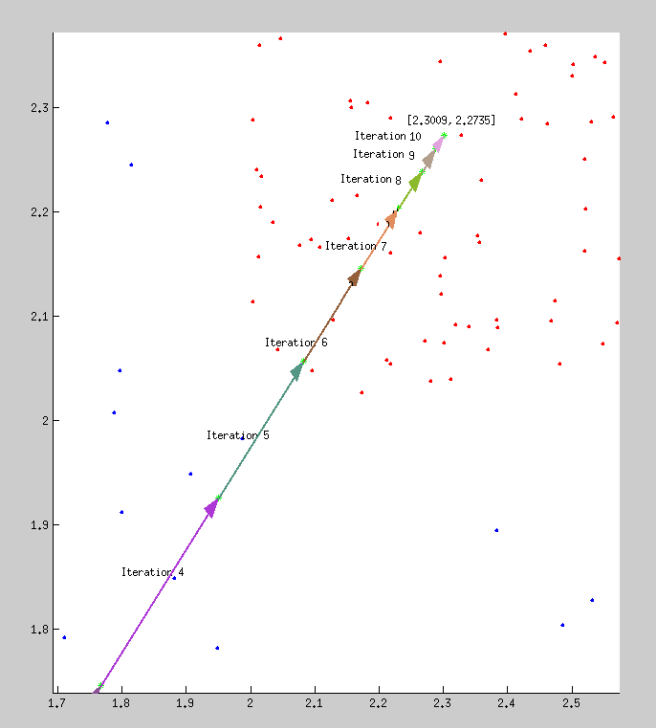
Şekil 4.18: İterasyon-8 sonucunda elde edilen değer ve İterasyon-7 ile arasındaki fark



Şekil 4.19: İterasyon-9 sonucunda elde edilen değer ve İterasyon-8 ile arasındaki fark



Şekil 4.20: İterasyon-10 sonucunda elde edilen değer ve İterasyon-9 ile arasındaki fark



Şekil 4.21: Yakınlaştırılmış iterasyonlar

4.2 Gerçek Deney Veri Tabanları ve Sonuçları

4.2.1 IRIS veri tabanı

Literatürde, IRIS bitki veri tabanı örüntü tanıma için belki de en iyi bilinen veri tabanlarından biridir. Fisher, 1936 tarafından oluşturulan bu veri tabanı her biri 50 örnek içeren 3 sınıftan oluşmaktadır. Her bir sınıf Iris bitkisinin bir türü olarak verilmiştir. Bu sınıflardan biri diğer iki sınıftan doğrusal olarak ayrılabilirken diğer sınıflar doğrusal olarak ayrılamamaktadır. Bu veri tabanı temel ve basittir ve bu tez çalışmasında önerilen yöntem ilk olarak bu veri tabanına uygulanmıştır.

IRIS veri tabanı 4 sayısal ve 1 sınıf etiketi olarak toplam 5 öznitelikten oluşmaktadır. Sınıf etiketi 3 farklı Iris Setosa, Iris Versicolour ve Iris Virginica etiketini içermektedir. Bu öznitelikler detaylı olarak Çizelge (4.1)'de verilmiştir.

Çizelge 4.1: IRIS Veri Tabanı Öznitekleri

Sütün	Öznitelik
1	Çanakyaprak uzunluğu
2	Çanakyaprak eni
3	Taçyaprak uzunluğu
4	Taçyaprak eni
5	Sınıf etiketi

Çizelge 4.2: IRIS Veri Tabanı Ortalama Doğruluk Oranı (%)

IRIS Ortalama Doğruluk Oranı (%) s =0.000001		
Yöntem	**C =10**	**C =50**
PCC-L1	95.99	95.33
PCC-L2	95.33	95.33
EPCC-L1	95.99	95.99
EPCC-L2	95.99	94.66
SVM	94.70	
EPCC-L1-C	98.66	97.66
EPCC-L2-C	97.33	96.66

EPCC-L1-C ve EPCC-L2-C yöntemlerinden alınan sonuç ile diğer yöntem sonuçları karşılaştırıldığında (Çizelge 4.2) IRIS veri tabanı için EPCC-L1-C yönteminin en yüksek doğruluk oranına (%98.66) sahip olduğu gözlemlenmiştir.

4.2.2 PIMA veri tabanı

Ulusal Diyabet ve Böbrek Hastalıkları Enstitüsü tarafından oluşturulan Pima Indians Diyabet veri tabanı örüntü tanımada sıkça kullanılan bir diğer veri kümesidir. Bu veri tabanı

Pima Indians olarak adlandırılan Arizona bölgesinde yaşamakta olan yerli Amerikalılardan elde edilmiştir. Ayrıca, Pima Indians dünyada diyabet hastalığına sahip en yoğun popülasyonu oluşturması sebebi ile diyabet araştırmalarında kullanılan en önemli veri tabanlardan biridir.

PIMA veri tabanı daha büyük veri kümesinin bir bölümü olarak seçilmiş örneklerden oluşmaktadır. Örnek olarak seçilen özniteliklerin başında tüm hastaların 21 yaşından büyük 768 Pima Indians kadından oluşmasıdır. Çizelge (4.3) veri tabanını oluşturan özellikleri detaylı olarak belirtmiştir. Bu veri tabanı için 8 girdi ve 1 çıktı olarak toplamda 9 öznitelik incelenmiştir.

Çizelge 4.3: PIMA Öznitelikleri

Sütün	**Öznitelik**
1	Hamilelik sayısı
2	Glükoz yoğunluğu
3	Kan basıncı
4	Deri kalınlığı
5	Serum insülin
6	Vücut kütlesi
7	Diyabet pedigri fonksiyonu
8	Yaş
9	Sınıf etiketi (0 veya 1)

Çizelge 4.4: PIMA Veri Tabanı Ortalama Doğruluk Oranı (%)

PIMA Ortalama Doğruluk Oranı (%) s =0.000001		
Yöntem	**C =10**	**C =50**
PCC-L1	76.80	76.80
PCC-L2	70.15	70.15
EPCC-L1	76.53	76.53
EPCC-L2	76.89	76.89
SVM	77.10	
EPCC-L1-C	76.41	76.70
EPCC-L2-C	77.30	77.03

Çıktı bilgisi veya sınıf etiketi 0 ise hastanın diyabet olmadığı ve 1 ise diyabet hastalığına yakalanmış olduğunu göstermektedir. Toplam 768 örnek sayısı, 0 sınıf etiketi için 500 örnek ve 1 sınıf etiketi için 268 örnek olarak ayrılmıştır.

PIMA veri tabanı için EPCC-L1-C ve EPCC-L2-C yöntem sonuçları ile diğer yöntem sonuçları karşılaştırıldığında (Çizelge 4.4), EPCC-L2-C yöntemi ile en yüksek doğruluk oranı %77.30 olarak elde edildiği gözlemlenmiştir.

4.2.3 WINE veri tabanı

WINE veri tabanı İtalyada aynı bölgede yetişen ama 3 farklı kültür bitkiden türetilen şarap için kimyasal analiz sonuçlarını içerir. Bu analiz her 3 şarap çeşitinde bulunan 13 bileşen özniteliği belirtmektedir. Bu öznitelikler Çizelge (4.5)'de verilmiştir. İlk öznitelik $1-3$ arası tam sayı değerler ile sınıf etiketi verilmiştir. Her sınıf sırayla 59, 71 ve 48 örnek içermektedir. Bu veri tabanı Riccardo Leardi taradından oluşturulmuştur ve genel olarak sınıflandırmak için kullanılmıştır (Forina vd., 1990).

Çizelge 4.5: WINE Veri Tabanı Öznitelikleri

Sütün	Öznitelik
1	Alkol
2	Malik asit
3	Karbonat
4	Karbonat alkalitesi
5	Magnezyum
6	Toplam fenol
7	Flavanoid
8	Flavanoid olmayan fenol
9	proantosiyanidin
10	Renk yoğunluğu
11	Renk özü
12	Seyreltilmiş şarap
13	Prolin

Çizelge 4.6: WINE Veri Tabanı Ortalama Doğruluk Oranı (%)

WINE Ortalama Doğruluk Oranı (%) s =0.000001		
Yöntem	**C =10**	**C =50**
PCC-L1	95.63	96.87
PCC-L2	93.75	93.13
EPCC-L1	95.63	96.87
EPCC-L2	96.25	95.63
SVM	96.90	
EPCC-L1-C	96.25	98.13
EPCC-L2-C	96.25	96.88

WINE veri tabanı için EPCC-L1-C ve EPCC-L2-C yöntem sonuçları ile diğer yöntem sonuçları karşılaştırıldığında (Çizelge 4.6), EPCC-L1-C yöntemi ile en yüksek doğruluk oranı olan %98.13 elde edildiği gözlemlenmiştir.

4.2.4 WDBC veri tabanı

WDBC veri tabanı Wisconsin Diagnostic Breast Cancer (WDBC) olarak tanınmaktdır. Öznitelikler dijitalleştirilmiş göğüs kütlesi iğne aspirasyon biyopsi imgesinden elde edilmiştir. Bu öznitelikler imgede bulunan hücre çekirdeğinin karakteristik yapısını betimlemektedir. Ayırma düzlemleri çoklu yüzey yöntem ağacı (K. P. Bennett, 1992), doğrusal programlama sınıflandırma kullanarak karar ağacı oluşturan yöntem ile

elde edilmiştir. İlgili öznitelikler 1-4 öznitelik ve 1-3 ayırma düzlem uzayında tam kapsamlı arama yapılarak seçilmiştir. Doğrusal programlama ile elde edilen 3 boyutlu uzayda ayırılan düzlem K. Bennett ve O. L. Mangasarian, 1992'de detaylı şekilde belirtilmiştir. Bu veri tabanı 569 örnek, 32 öznitelik (ID, teşhis, 30 gerçel-değerli girdi öznitelik) ve 357 iyicil (B) ve 212 kötücül (M) olmak üzere sınıf dağılımından oluşmaktadır (Çizelge 4.7).

Çizelge 4.7: WDBC veri tabanında kullanılan her hücre çekirdeği için gerçel-değerli Öznitelikler

Sütün	**Gerçel-değerli Öznitelik**
1	Yarıçap (Merkezden çevresindeki noktalara kadar olan ortalama uzaklık)
2	Karakter (Gri ölçekli değerlerin standart sapması)
3	Çevre
4	Alan
5	Pürüzsüzlük (Yarıçap uzunluğundaki yerel değişiklikler)
6	Yoğunluk
7	İçbükeylik
8	İçbükeylik noktaları
9	Simetri
10	Fraktal boyut

Çizelge 4.8: WDBC Veri Tabanı Ortalama Doğruluk Oranı (%)

WDBC Ortalama Doğruluk Oranı (%) s =0.000001		
Yöntem	**C =10**	**C =50**
PCC-L1	97.37	97.37
PCC-L2	95.25	95.05
EPCC-L1	97.86	97.86
EPCC-L2	97.73	97.53
SVM	97.70	
EPCC-L1-C	97.91	96.99
EPCC-L2-C	97.73	97.55

WDBC veri tabanı için EPCC-L1-C ve EPCC-L2-C yöntemlerinin sonuçlarını diğer yöntem sonuçları ile karşılaştırıldığında (Çizelge 4.8), EPCC-L1-C yönteminin en yüksek doğruluk oranına (%97.91) sahip olduğu gözlemlenmiştir.

4.2.5 VOC veri tabanı

Volatile Organic Compound veri tabanı, VOC olarak bilinmektedir. Bu veriler altı kuartz kristal mikro sensörler tarafından tespit edilen uçucu organik bileşiklerin kimliğini belirleyen gerçek dünya uygulamasından elde edilmiştir. Bu veri kümesi 5 farklı uçucu organik bileşken sınıfı (ethanol, octane, toluene, xylene and trichloroethylene) ve 384 örnekten oluşmaktadır. Eğitim kümesinin boyut küçüklüğü ve tanımlamanın VOC yoğunluğundan bağımsız olması gerektiği gerçekliğinin yanısıra sensor sinyallerinin her birinin bağımlı olması problemi daha da karmaşık ve zor hale getirmektedir.

Volatile Organic Compound veri tabanı tropospheric hydrocarbon ölçüm için genel taslak amacı ile oluşturulmuştur. Bu veri tabanı 202 örnekten elde edilmiştir ve genel olarak katsayı oranı, photolysis frekansları, karıştırma oranı, emme verisi gibi özniteliklerden oluşmaktadır.

Çizelge 4.9: VOC Veri Tabanı Ortalama Doğruluk Oranı (%)

VOC Ortalama Doğruluk Oranı (%) s =0.000001		
Yöntem	**C =10**	**C =50**
PCC-L1	77.72	79.54
PCC-L2	76.27	78.09
EPCC-L1	82.88	91.38
EPCC-L2	90.72	85.04
SVM	79.42	
EPCC-L1-C	84.30	83.81
EPCC-L2-C	83.42	84.80

VOC veri tabanına uygulanan yöntem sonuçları Çizelge (4.9) olarak verilmiştir. Bu sonuçlara göre en yüksek doğruluk oranı EPCC-L1 yöntemi ile elde edilmiştir. EPCC-L1-C ve EPCC-L2-C yöntemleri ise yine bu sonuca yakın bir değer elde etmiştir.

4.2.6 GLASS veri tabanı

GLASS veri tabanı sınıflandırılması kriminolojik araştırmalar için önemli bir konudur. Olay yerinde bulunan camlar kanıt olarak kullanılır ve laboratuvarlarda incelenir. Bu nedenle bir camın hangi sınıfa ait olduğunu tespit etmek önemlidir. Glass veri tabanı her biri 10 öznitelik (ve bir sınıf özniteliği) içeren 214 örnekten oluşmaktadır. Bu öznitelik listesi Çizelge (4.10) olarak verilmiştir. Sınıf dağılımı ise 163 örnek pencere camı ve 51 örnek diğer cam çeşidi olmak üzere ikiye ayrılmaktadır. GLASS veri tabanına uygulanan

Çizelge 4.10: GLASS Veri Tabanı Öznitelikleri

Sütün	Öznitelik
1	Örnek numarası (1-214)
2	RI: Kırılma indisi
3	Na: Sodyum miktarı
4	Mg: Manyezyum
5	Al: Alminyum
6	Si: Silikon
7	K: Potasyum
8	Ca: Kalsiyum
9	Ba: Baryum
10	Fe: Demir
11	Sınıf etiketi

Çizelge 4.11: GLASS Veri Tabanı Ortalama Doğruluk Oranı (%)

GLASS Ortalama Doğruluk Oranı (%) s =0.000001		
Yöntem	**C =10**	**C =50**
PCC-L1	98.09	98.09
PCC-L2	96.57	96.57
EPCC-L1	97.62	97.62
EPCC-L2	96.19	96.19
SVM	95.45	
EPCC-L1-C	97.14	97.62
EPCC-L2-C	94.76	94.29

yöntemlerin sonucu Çizelge (4.11)'de görüldüğü gibi yakın değerlere sahiptir. En yüksek sonuç değeri PCC-L1 yöntemine (%98.09) ait olmasına rağmen EPCC-L1-C yöntemi (%97.62) ile aralarında büyük fark yoktur.

4.2.7 MF veri tabanı

Multi-Feature Digit Veri tabanı (MF) ilk olarak Johns, 1959 çalışmasında kullanılmıştır. Bu veri tabanı Hollanda Hizmet Kurumu harita koleksiyonundan elde edilen elle yazılmış rakamlar (0-9) özniteliğini içermektedir. Her sınıf için 200 örnek olmak üzere toplamda 2000 örnek ve 10 sınıftan oluşan bu veri tabanı ikili imge olarak dijitalleştirilmiştir. Bu sayılar 6 öznitelik veri kümesine dağılmış toplam 649 öznitelikten oluşmaktadır (Çizelge 4.12).

Çizelge 4.12: MF Veri Tabanı Öznitelikleri

Sütün	Öznitelik
1	mfeat-fou: 76 (karakter biçiminin Fourier katsayısı)
2	mfeat-fac: 216 (korelasyon)
3	mfeat-kar: 64 (Karhunen-Love katsayısı)
4	mfeat-pix: 240 (2x3)'luk pencerelerde piksel ortalaması
5	mfeat-zer: 47 (Zernike momenti)
6	mfeat-mor: 6 (biçimsel öznitelik)

Her bir veri kümesinde 2000 örnek ASCI olarak 2000 satırda saklanmaktadır. İlk 200 örnek 0 sınıfına ait ve kalan örnekler her 200 tanesi sıra ile 1-9 arasındaki sınıflara aittir.

Çizelge 4.13: MF Veri Tabanı Ortalama Doğruluk Oranı (%)

MF Ortalama Doğruluk Oranı (%) s =0.000001		
Yöntem	**C =10**	**C =50**
PCC-L1	94.45	94.45
PCC-L2	92.45	91.25
EPCC-L1	95.40	96.30
EPCC-L2	95.60	96.00
SVM	93.90	
EPCC-L1-C	94.90	95.40
EPCC-L2-C	95.65	95.80

Çizelge (4.13), çeşitli yöntem sonuçlarını göstermektedir.En yüksek doğruluk oranına sahip EPCC-L1 yöntemi (%96.30) ile EPCC-L2-C yöntemi (%95.80) arasında büyük bir fark olmadığı gözlemlenmiştir.

4.2.8 LR veri tabanı

Letter Recognition veri tabanı (LR) harf tanıma üzerine David J. Slate tarafından oluşturulmuştur. Hedef, büyük miktarda siyah beyaz dikdörtgen piksel ile gösterilen İngiliz alfabesindeki her bir 26 büyük harfı tanımlamaktır. Karakter imgeleri 20 farklı fontta ve her

harf bu 20 farklı font arasında 20000 benzersiz örnek oluşan bir dosya oluşturmak için rastgele dağılmıştır. Her örnek 16 temel nümerik özniteliğe dönüştürülmüştür ve daha sonra tam sayı değeri 0 ile 15 arasında olacak şekilde ölçeklendirilmiştir. Genel olarak ilk 16000 öğe eğitilmektedir ve sonuç model kullanılarak geriye kalan 4000 öğe için harf sınıfı tahmini yapılmaktadır. LR veri tabanı sınıf etiketi ile birlikte toplamda 17 tamsayı öznitelikten oluşmaktadır (Çizelge 4.14).

Çizelge (4.15)'de verilen çeşitli yöntem sonuçlarına göre en yüksek doğruluk oranı EPCC-L2-C (%79.72) yönteminde elde edilmiştir. Ayrıca EPCC-L2 yönteminin sonucu (%79.34) da bu sonuca yakın bir değerdedir.

Çizelge 4.14: LR Veri Tabanı Öznitelikleri

Sütün	**Öznitelik**
1	Sınıf Etiketi: Büyük harf A'dan Z'ye 26 Sınıf
2	x-box: box yatay pozisyonu
3	y-box: box dikey pozisyonu
4	width: box genişliği
5	high: box uzunluğu
6	onpix: piksel toplam sayısı
7	x-bar: box içinde x ortalama pikseli
8	y-bar: box içinde y ortalama pikseli
9	x2bar: ortalama c variyans
10	y2bar: ortalama y variyans
11	xybar: ortalama x ve y korelasyon
12	x2ybr: ortalma x*x*y
13	xy2br: ortalama x*y*y
14	x-ege: ortalama kenar sayısı soldan sağa
15	xegvy: x-ege ve y korelasyonu
16	y-ege: ortalama kenar sayısı aşağıdan yukarıya
17	yegvx: y-ege ve x korelasyonu

Çizelge 4.15: LR Veri Tabanı Ortalama Doğruluk Oranı (%)

LR Ortalama Doğruluk Oranı (%) s =0.000001		
Yöntem	**C =10**	**C =50**
PCC-L1	65.50	65.50
PCC-L2	33.60	33.60
EPCC-L1	75.41	75.41
EPCC-L2	79.34	79.34
SVM	59.80	
EPCC-L1-C	79.63	79.65
EPCC-L2-C	79.70	79.72

5. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasındaki amaç günümüzde bilim, endüstri, askeri ve ticari alanlarda büyük öneme sahip olan sınıflandırma problemlerini matematiksel programlamalar ile geliştirebilmektir. Literatürde kullanılan çeşitli sınıflandırıcıların yanısıra çok yüzlü konik sınıflandırıcılar, pozitif sınıf verilerini negatiflerden ayırmak için daha uygun ve kompakt konveks bölge şekillerini oluşturması, sınıflandırıcı modeli bütünsel optimum sonuç üretmesi, büyük problemlere uygun ölçekte olması ve gürbüz boşluk bazlı cost fonksiyonu kullanarak çakışmayı önlemesi gibi özelliklere sahip olması nedeni ile uygun bir sınıflandırıcı modelidir. Bu sınıflandırıcıyı geliştirmek için ve yapısının daha sağlam bir şekilde belirlenebilmesi için koni tepe noktası tahmini yapılmıştır. Bu tahmin algoritmasında kayıp fonksionu ve L1 (L2) normlu konilerden yararlanarak daha gürbüz çok yüzlü konik sınıflandırıcı elde edilmiştir.

Koni tepe noktası tahmini sonucunda elde edilen sınıflandırıcı algoritması, *UCI Machine Learning Repository* kaynağındaki çeşitli gerçek veri tabanlarına uygulanarak sonuçları diğer sınıflandırıcı modellerinin sonuçları ile karşılaştırılmıştır. Bu sonuçlara göre koni tepe noktası tahmin algoritması ile çoğu veri tabanında en yüksek doğruluk oranı elde edilmiştir ve bir kaç veri tabanında ise diğer yöntemlere yakın doğruluk oranına ulaşılmıştır. Ayrıca sentetik bir veri kümesi oluşturularak koni tepe noktası tahmin algoritması ile pozitif verilerin ortalaması kullanan algoritma karşılaştırılmıştır. Bu karşılaştırma ile sonuçların birbirine yakın değerlerde olduğu gözlemlenmiştir.

Bu yapılan çalışmalar sonucuna göre, her iki farklı yöntem ile birbirlerine yakın doğruluk değerlerine ulaşılmıştır. Bununla beraber, tepe noktası olarak pozitif verilerin ortalamasının alındığı yöntem koni tepe noktası tahmini yapılarak elde edilen yönteme göre büyük veri tabanlarında daha hızlı çalışmaktadır. Ayrıca koni tepe noktası tahmin yöntemi bir çok optimizasyon işlemini içerdiği için karmaşık bir yöntemdir. Bu nedenler sonucunda, daha hızlı bir şekilde deney sonuçlarına ulaşılmak istenildiğinde, koni tepe noktası pozitif verilerin ortalaması olarak alınması tercih edilebilir. Küçük boyutlu veri tabanlarında ise bu çalışmada gerçekleştirilen yöntemin kullanılması daha uygun olacaktır.

KAYNAKLAR DİZİNİ

Adam, Chai K, Hwee T Ng ve Hai L Chieu (2002). "Bayesian Online Classifiers for Text Classification and Filtering". İn: *Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval*, pp. 97–104. ISSN: 01635840. DOI: 10.1145/564392.564395. URL: http://doi.acm.org/10.1145/564376.564395.

Almeida, Tiago A., Jurandy Almeida ve Akebo Yamakami (2011). "Spam filtering: How the dimensionality reduction affects the accuracy of Naive Bayes classifiers". İn: *Journal of Internet Services and Applications* 1.3, pp. 183–200. ISSN: 18674828. DOI: 10.1007/s13174-010-0014-7.

Alpaydın, Ethem (2004). "Introduction to machine learning". İn: *Methods in Molecular Biology* 1107, pp. 105–128. ISSN: 10643745. DOI: 10.1007/978-1-62703-748-8-7. arXiv: 0904.3664v1.

Astorino, A. ve M. Gaudioso (2002). "Polyhedral Separability Through Successive LP". İn: *Journal of Optimization Theory and Applications* 112.2, pp. 265–293. ISSN: 0022-3239. DOI: 10.1023/A:1013649822153. URL: http://link.springer.com/10.1023/A:1013649822153.

Bach, Francis ve R Jenatton (2011). "Convex optimization with sparsity-inducing norms". İn: *Optimization for Machine Learning*, pp. 1–35. ISSN: 1935-8237. DOI: 10.1561/2200000015. arXiv: 1108.0775v2. URL: http://books.google.com/books?hl=en%7B%5C&%7Dlr=%7B%5C&%7Did=JPQx7s2L1A8C%7B%5C&%7Doi=fnd%7B%5C&%7Dpg=PA19%7B%5C&%7Ddq=Convex+Optimization+with+Sparsity-Inducing+Norms%7B%5C&%7Dots=vcceDmmaC9%7B%5C&%7Dsig=DRcQwnErORdIkJbi5XhnEeWaf6A.

Bagirov, A. M., J. Ugon ve D. Webb (2011). “An efficient algorithm for the incremental construction of a piecewise linear classifier”. İn: *Information Systems* 36.4, pp. 782–790. ISSN: 03064379. DOI: 10.1016/j.is.2010.12.002.

Bagirov, Adil M. (2005). “Max–min separability”. İn: *Optimization Methods and Software* 20.2-3, pp. 277–296. ISSN: 1055-6788. DOI: 10.1080/10556780512331318263. URL: http://www.tandfonline.com/doi/abs/10.1080/10556780512331318263.

Bagirov, Adil M., Julien Ugon, Dean Webb, Gurkan Ozturk ve Refail Kasimbeyli (2013). “A novel piecewise linear classifier based on polyhedral conic and max–min separabilities”. İn: *TOP* 21.1, pp. 3–24. ISSN: 1134-5764. DOI: 10.1007/s11750-011-0241-5. URL: http://link.springer.com/10.1007/s11750-011-0241-5.

Batista, Gustavo E. A. P. A. ve Maria Carolina Monard (2003). “An analysis of four missing data treatment methods for supervised learning”. İn: *Applied Artificial Intelligence* 17.5-6, pp. 519–533. ISSN: 0883-9514. DOI: 10.1080/713827181. URL: http://www.tandfonline.com/doi/abs/10.1080/713827181.

Behnke, Sven (2003). “Hierarchical Neural Networks for Image Interpretation”. İn: *Lecture Notes in Computer Science* 2766, p. 244. ISSN: 03029743. DOI: 10.1287/mksc.1060.0207. arXiv: 9780201398298.

Bengio, Y, A Courville ve P Vincent (2013). “Representation Learning: A Review and New Perspectives”. İn: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8, pp. 1798–1828. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2013.50. arXiv: arXiv:1206.5538v2.

Bennett, K. P. (1992). “Decision Tree Construction Via Linear Programming”. İn: *roceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society*, pp. 97–101.

Bennett, Kristin ve O. L. Mangasarian (1992). “Robust linear programming discrimination of two linearly inseparable sets”. İn: *Optimization Methods and Software* 1.1, pp. 23–34. ISSN: 1055-6788. DOI: 10.1080/10556789208805504.

Berger, James O., Elías Moreno, Luis Raul Pericchi, M. Jesús Bayarri, José M. Bernardo, Juan A. Cano, Julián De la Horra, Jacinto Martín, David Ríos-Insúa, Bruno Betrò, A. Dasgupta, Paul Gustafson, Larry Wasserman, Joseph B. Kadane, Cid Srinivasan, Michael Lavine, Anthony O’Hagan, Wolfgang Polasek, Christian P. Robert, Constantinos Goutis, Fabrizio Ruggeri, Gabriella Salinetti ve Siva Sivaganesan (1994). “An overview of robust Bayesian analysis”. İn: *Test* 3.1, pp. 5–124. ISSN: 11330686. DOI: 10.1007/BF02562676.

Bradley, PS, U.M. Fayyad ve OL Mangasarian (1999). “Data mining: Overview and optimization opportunities”. İn: *INFORMS Journal on Computing* 11.January 1998, pp. 217–238. URL: http://www.ergasya.tuc.gr/Users/matsatsinis/courses/postgrad/Data%20Mining.pdf.

Burges, C.J.C. J Christopher J C (1998). “A tutorial on support vector machines for pattern recognition”. İn: *Data Mining and Knowledge Discovery* 2.2, pp. 121–167. ISSN: 13845810. DOI: 10.1023/A:1009715923555. arXiv: 1111.6189v1. URL: http://www.springerlink.com/index/Q87856173126771Q.pdf.

Cevikalp, Hakan ve Merve Elmas (2016). “Robust transductive support vector machines”. İn: *2016 24th Signal Processing and Communication Application Conference (SIU)*. IEEE, pp. 985–988. ISBN: 978-1-5090-1679-2. DOI: 10.1109/SIU.2016.7495907. URL: http://ieeexplore.ieee.org/document/7495907/.

Cevikalp, Hakan ve Bill Triggs (2012). “Efficient object detection using cascades of nearest convex model classifiers”. İn: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3138–3145. ISSN: 10636919. DOI: 10.1109/CVPR.2012.6248047.

Cevikalp, Hakan ve Bill Triggs (2013). “Hyperdisk based large margin classifier”. İn: *Pattern Recognition* 46.6, pp. 1523–1531. ISSN: 00313203. DOI: 10.1016/j.patcog.2012.11.004.

Cevikalp, Hakan, Bill Triggs ve Vojtech Franc (2013). “Face and landmark detection by using cascade of classifiers”. İn: *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2013*. ISBN: 9781467355452. DOI: 10.1109/FG.2013.6553705.

Chekuri, Chandra (2009). *Combinatorial optimization*. techreport.

Cireşan, Dan C., Ueli Meier, Jonathan Masci, Luca M. Gambardella ve Jürgen Schmidhuber (2011). “Flexible, high performance convolutional neural networks for image classification”. İn: *IJCAI International Joint Conference on Artificial Intelligence*, pp. 1237–1242. ISBN: 9781577355120. DOI: 10.5591/978-1-57735-516-8/IJCAI11-210. arXiv: arXiv:1011.1669v3.

Cortes, Corinna ve Vladimir Vapnik (1995). “Support-Vector Networks”. İn: *Machine Learning* 20.3, pp. 273–297. ISSN: 15730565. DOI: 10.1023/A:1022627411411. arXiv: arXiv:1011.1669v3.

Cover, T. ve P. Hart (1967). “Nearest neighbor pattern classification”. İn: *IEEE Transactions on Information Theory* 13.1, pp. 21–27. ISSN: 0018-9448. DOI: 10.1109/TIT.1967.1053964. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1053964.

Cover, T.M. Thomas M. (1968). “Estimation by the nearest neighbor rule”. İn: *IEEE Transactions on Information Theory* 14.1, pp. 50–55. ISSN: 0018-9448. DOI: 10.1109/TIT.1968.1054098. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1054098%7B%5C%%7D5Cnhttp://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=1054098%7B%5C%%7D5Cnhttp://ieeexplore.ieee.org/

xpls/abs%7B%5C_%7Dall.jsp?arnumber=1054098%7B%5C%7D5Cnhttp://ieeexplore.ieee.org/Xplore/cookiedetectresponse.jsp?rel.

Cristianini, Nello ve John Shawe-Taylor (2000). "An Introduction to Support Vector Machines and other kernel based learning methods". İn: *Ai Magazine* 22.2, p. 190. ISSN: 0738-4602. DOI: citeulike-article-id:114719. arXiv: 0104167 [arXiv:cond-mat].

Dasarathy, Belur V. (1980). "Nosing Around the Neighborhood : A New System Structure and Classification Rule for Recognition in Partially Exposed Environment". İn: *Ieee Transactions on Pattern Analysis and Machine Intelligence* I.1, pp. 369–371. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1981.4767173.

Duda, R O ve P E Hart (1973). *Pattern Classification and Scene Analysis*. **volume** 95, pp. 1–5. ISBN: 0471223611. DOI: 10.2307/1573081. URL: http://www.jstor.org/stable/1573081?origin=crossref.

Duda, R O, P E Hart ve D G Stork (2000). *Pattern Classification*. DOI: 10.1038/npp.2011.9.

Dundar, M. Murat, Matthias Wolf, Sarang Lakare, Marcos Salganicoff ve Vikas C. Raykar (2008). "Polyhedral classifier for target detection". İn: *Proceedings of the 25th international conference on Machine learning - ICML '08*. New York, New York, USA: ACM Press, pp. 288–295. ISBN: 9781605582054. DOI: 10.1145/1390156.1390193. URL: http://portal.acm.org/citation.cfm?doid=1390156.1390193.

Ertekin, Seyda, Léon Bottou ve C Lee Giles (2011). "Nonconvex online support vector machines." İn: *IEEE transactions on pattern analysis and machine intelligence* 33.2, pp. 368–381. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2010.109.

Fisher, R A (1936). “The use of multiple measurements in taxonomic problems”. İn: *Annals of Eugenics* 7.2, pp. 179–188. ISSN: 1469-1809. DOI: 10.1111/j.1469-1809.1936.tb02137.x. arXiv: arXiv:1011.1669v3.

Forina, M, R Leardi, C Armanino ve S Lanteri (1990). “PARVUS: An Extendible Package for Data Exploration, Classification and Correlation”. İn: *Journal of Chemometrics* 4.2, pp. 191–193. ISSN: 0886-9383. DOI: 10.1002/cem.1180040210. URL: http://doi.wiley.com/10.1002/cem.1180040210%7B%5C%%7D5Cnhttp://scholar.google.com/scholar?hl=en%7B%5C&%7DbtnG=Search%7B%5C&%7Dq=intitle:PARVUS:+An+extendable+package+of+programs+for+data+exploration,+classification+and+correlation,%7B%5C#%7D1.

Fukunaga, Keinosuke ve Larry D. Hostetler (1975). “k-Nearest-Neighbor Bayes-Risk Estimation”. İn: *IEEE Transactions on Information Theory* 21.3, pp. 285–293. ISSN: 15579654. DOI: 10.1109/TIT.1975.1055373.

Fukushima, Kunihiko (1980). “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. İn: *Biological Cybernetics* 36.4, pp. 193–202. ISSN: 03401200. DOI: 10.1007/BF00344251. arXiv: arXiv:1011.1669v3.

Gasimov, Rafail N. ve Gurkan Ozturk (2006). “Separation via polyhedral conic functions”. İn: *Optimization Methods and Software* 21.4, pp. 527–540. ISSN: 1055-6788. DOI: 10.1080/10556780600723252. URL: http://www.tandfonline.com/doi/abs/10.1080/10556780600723252.

Genton, Marc G (2001). “Classes of Kernels for Machine Learning: A Statistics Perspective”. İn: *Journal of Machine Learning Research* 2, pp. 299–312. ISSN: 15324435. DOI: 10.1162/15324430260185646.

Ghahramani, Zoubin (2004). “Unsupervised Learning BT - Advanced Lectures on Machine Learning”. İn: *Advanced Lectures on Machine Learning* 3176.Chapter 5, pp. 72–112. ISSN: 1745-1337. DOI: 10.1007/978-3-540-28650-9_5. URL:

http : / / link . springer . com / 10 . 1007 / 978 - 3 - 540 - 28650 - 9%7B%5C_%7D5%7B%5C%%7D5Cnpapers3://publication/doi/10.1007/978-3-540-28650-9%7B%5C_%7D5.

Hellman, Martin E. (1970). "The Nearest Neighbor Classification Rule with a Reject Option". İn: *IEEE Transactions on Systems Science and Cybernetics* 6.3, pp. 179–185. ISSN: 21682887. DOI: 10.1109/TSSC.1970.300339.

Huang, Fu Jie ve Yann LeCun (2006). "Large-scale learning with SVM and convolutional nets for generic object categorization". İn: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. **volume** 1, pp. 284–291. ISBN: 0769525970. DOI: 10.1109/CVPR.2006.164.

Jayadeva, R. Khemchandani ve Suresh Chandra (2007). "Twin support vector machines for pattern classification". İn: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.5, pp. 905–910. ISSN: 01628828. DOI: 10.1109/TPAMI.2007.1068.

Johns, Milton Vernon (1959). *AN EMPIRICAL BAYES APPROACH TO NON-PARAMETRIC TWO-WAY CLASSIFICATION,*

Kanal, L. N. (1963). "Statistical methods for pattern classification". İn: *Semiautomatic Imagery Screening Research Study and Experimental Investigation,* 43.

Kantchelian, Alex, Michael Carl Tschantz, Ling Huang, Peter L Bartlett, Anthony D Joseph ve J D Tygar (2014). "Large-margin Convex Polytope Machine". İn: *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 3248–3256. ISSN: 10495258. URL: http://dl.acm.org/citation.cfm?id=2969033.2969189.

Keerthi, S. S. ve E. G. Gilbert (2002). "Convergence of a generalized SMO algorithm for SVM classifier design". İn: *Machine Learning* 46.1-3, pp. 351–360. ISSN: 08856125. DOI: 10.1023/A:1012431217818.

Kohavi, Ron ve Foster Provost (1998). “Glossary of Terms”. İn: *Machine Learning*. 30.2-3, pp. 271–274. ISSN: 0885-6125. DOI: 10.1023/A:1017181826899. URL: http://dl.acm.org/citation.cfm?id=288808.288815%7B%5C%%7D5Cnhttp://robotics.stanford.edu/%7B~%7Dronnyk/glossary.html.

Kononenko, Igor (1994). “Estimating attributes: Analysis and extensions of RELIEF”. İn: Springer Berlin Heidelberg, pp. 171–182. DOI: 10.1007/3-540-57868-4_57. URL: http://link.springer.com/10.1007/3-540-57868-4%7B%5C_%7D57.

Kotsiantis, Sotiris B. (2007). “Supervised machine learning: A review of classification techniques”. İn: *Informatica* 31, pp. 249–268. ISSN: 09226389. DOI: 10.1115/1.1559160. URL: http://books.google.com/books?hl=en%7B%5C&%7Dlr=%7B%5C&%7Did=vLiTXDHr%7B%5C_%7DsYC%7B%5C&%7Doi=fnd%7B%5C&%7Dpg=PA3%7B%5C&%7Ddq=survey+machine+learning%7B%5C&%7Dots=CVsyuwYHjo%7B%5C&%7Dsig=A6wYWvywU8XTc7Dzp8ZdKJaW7rc$%5Cbackslash$npapers://5e3e5e59-48a2-47c1-b6b1-a778137d3ec1/Paper/p800$%5Cbackslash$nhttp://www.informatica.si/PDF/31-3/11%7B%5C_%7DKotsiantis%20-%20S.

Krizhevsky, Alex, IIya Sulskever ve Geoffret E Hinton (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. İn: *Advances in Neural Information and Processing Systems (NIPS)*, pp. 1–9.

Lakoumentas, John, John Drakos, Marina Karakantza, George Sakellaropoulos, Vasileios Megalooikonomou ve George Nikiforidis (2012). “Optimizations of the naïve-Bayes classifier for the prognosis of B-Chronic Lymphocytic Leukemia incorporating flow cytometry data”. İn: *Computer Methods and Programs in Biomedicine* 108.1, pp. 158–167. ISSN: 01692607. DOI: 10.1016/j.cmpb.2012.02.009.

Lanzi, Pier Luca (2012). *Machine Learning and Data Mining*. techreport.

Lecun, Yann, Yoshua Bengio ve Geoffrey Hinton (2015). “Deep learning”. İn: *Nature* 521.1, pp. 436–444. ISSN: 1548-7091. DOI: 10.1038/nature14539. arXiv: arXiv:1312.6184v5. URL: http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html.

LeCun, Yann, Léon Bottou, Yoshua Bengio ve Patrick Haffner (1998). “Gradient-based learning applied to document recognition”. İn: *Proceedings of the IEEE* 86.11, pp. 2278–2323. ISSN: 00189219. DOI: 10.1109/5.726791. arXiv: 1102.0183.

Lee, Honglak, Roger Grosse, Rajesh Ranganath ve Andrew Y Ng (2009). “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”. İn: *Proceedings of the 26th Annual International Conference on Machine Learning ICML 09* 2008, pp. 1–8. ISSN: 02643294. DOI: 10.1145/1553374.1553453. arXiv: arXiv:1301.3605v3. URL: http://portal.acm.org/citation.cfm?doid=1553374.1553453.

Malovini, Alberto, Nicola Barbarini, Riccardo Bellazzi ve Francesca De Michelis (2012). “Hierarchical Naive Bayes for genetic association studies”. İn: *BMC Bioinformatics* 13.Suppl 14, S6. ISSN: 1471-2105. DOI: 10.1186/1471-2105-13-S14-S6. URL: http://www.biomedcentral.com/1471-2105/13/S14/S6/abstract$%5Cbackslash$nhttp://www.biomedcentral.com/1471-2105/13/S14/S6$%5Cbackslash$nhttp://www.biomedcentral.com/content/pdf/1471-2105-13-S14-S6.pdf.

Mangasarian, Olvi L. ve Edward W. Wild (2006). “Multisurface proximal support vector machine classification via generalized eigenvalues”. İn: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.1, pp. 69–74. ISSN: 01628828. DOI: 10.1109/TPAMI.2006.17.

Manwani, Naresh ve P S Sastry (2010). “Learning Polyhedral Classifiers Using Logistic Function”. İn: *Conference Proceedings* 13, pp. 17–30.

Markovitch, Shaul ve Dan Rosenstein (2002). “Feature generation using general constructor functions”. İn: *Machine Learning* 49.1, pp. 59–98. ISSN: 08856125. DOI: 10.1023/A:1014046307775.

Megiddo, Nimrod (1988). “On the complexity of polyhedral separability”. İn: *Discrete & Computational Geometry* 3.4, pp. 325–337. ISSN: 0179-5376. DOI: 10.1007/BF02187916. URL: http://link.springer.com/10.1007/BF02187916.

Mitchell, Tom ve Avrim Blum (1998). “Combining labeled and unlabeled data with co-training”. İn: *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100. ISSN: 1098-6596. DOI: 10.1145/279943.279962. arXiv: arXiv:1011.1669v3. URL: http://dl.acm.org/citation.cfm?id=279943.279962.

Montavon, Grégoire, Gb Geneviève B. Orr, Klaus-Robert Müller, Y LeCun, L Bottou, Gb Geneviève B. Orr, Kr Muller, Grégoire Montavon, Gb Geneviève B. Orr ve Klaus-Robert Müller (1998). *Neural Networks: Tricks of the Trade*. MAY 2000, p. 432. ISBN: 3540653112. DOI: 10.1007/3-540-49430-8. URL: http://scholar.google.com/scholar?hl=en%7B%5C&%7DbtnG=Search%7B%5C&%7Dq=intitle:Neural+Networks:+Tricks+of+the+Trade%7B%5C#%7D3.

Moore, Dan H. (1987). “Classification and regression trees, by Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. Brooks/Cole Publishing, Monterey, 1984,358 pages, $27.95”. İn: *Cytometry* 8.5, pp. 534–535. ISSN: 0196-4763. DOI: 10.1002/cyto.990080516. URL: http://doi.wiley.com/10.1002/cyto.990080516.

Murthy, Sreerama K. ve Sreerama K. (1998). “Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey”. İn: *Data Mining and Knowledge Discovery* 2.4, pp. 345–389. ISSN: 13845810. DOI: 10.1023/A:1009744630224. URL: http://link.springer.com/10.1023/A:1009744630224.

Nowlan, Steven J (1990). “Maximum Likelihood Competitive Learning”. İn: *Advances in Neural Information Processing Systems 2*, pp. 574–582. ISBN: 1558601007. URL: http://portal.acm.org/citation.cfm?id=109295.

Number, A D ve N E W Limitation Change (1952). “Discriminatory analysis – Nonparametric discrimination: Small sample performance”. İn: *Project No. 21-49-004, Report No. 11, Contract No. AF 41(129)-31, USAF School of Aviation, Randolph Field, Texas*. May. ISSN: 03067734.

Orsenigo, Carlotta ve Carlo Vercellis (2007). “Accurately learning from few examples withapolyhedral classifier”. İn: *Computational Optimization and Applications* 38.2, pp. 235–247. ISSN: 0926-6003. DOI: 10.1007/s10589-007-9041-0. URL: http://link.springer.com/10.1007/s10589-007-9041-0.

Ozturk, Gurkan, Adil M. Bagirov ve Refail Kasimbeyli (2015). “An incremental piecewise linear classifier based on polyhedral conic separation”. İn: *Machine Learning* 101.1-3, pp. 397–413. ISSN: 0885-6125. DOI: 10.1007/s10994-014-5449-9. URL: http://link.springer.com/10.1007/s10994-014-5449-9.

Ozturk, Gurkan ve Refail Kasimbeyli (2013). “A Two Objective Classification Approach based on Conic Functions”. İn: *International Conference on Multiple Criteria Decision Making*. Malaga.

Pang-Ning, Tan, Michael Steinbach ve Vipin Kumar (2006). “Introduction to data mining”. İn: *Library of Congress*, p. 796. ISSN: 00224405. DOI: 10.1016/0022-4405(81)90007-8.

Platt, John C. (1999). *Fast training of support vector machines using sequential minimal optimization*, p. 376. ISBN: 0262194163. DOI: 10.1109/ISKE.2008.4731075. URL: http://dl.acm.org/citation.cfm?id=299105.

Quinlan, J R ve R M Cameron-Jones (1995). “Oversearching and Layered Search in Empirical Learning”. İn:

Rokach, Lior ve Oded Maimon (2005). “Top-down induction of decision trees classifiers - A survey”. İn: *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 35.4, pp. 476–487. ISSN: 10946977. DOI: 10.1109/TSMCC.2004.843247.

Rosasco, Lorenzo, Ernesto De Vito, Andrea Caponnetto, Michele Piana ve Alessandro Verri (2004). “Are loss functions all the same?” İn: *Neural computation* 16.5, pp. 1063–1076. ISSN: 0899-7667. DOI: 10.1162/089976604773135104.

Scheirer, Walter J., Anderson De Rezende Rocha, Archana Sapkota ve Terrance E. Boult (2013). “Toward open set recognition”. İn: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.7, pp. 1757–1772. ISSN: 01628828. DOI: 10.1109/TPAMI.2012.256. arXiv: arXiv:1011.1669v3.

Sebastiani, Fabrizio (2003). “Advances in information retrieval : 25th European Conference on IR Research, ECIR 2003, Pisa, Italy, April 14-16, 2003 : proceedings”. İn: *Lecture notes in computer science* 2633, xv, 624 p. URL: http://link.springer-ny.com/link/service/series/0558/tocs/t2633.htm.

Simard, P.Y., D. Steinkraus ve J.C. Platt (2003). “Best practices for convolutional neural networks applied to visual document analysis”. İn: *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* Pp. 1–6. DOI: 10.1109/ICDAR.2003.1227801.

Tax, David M J ve Robert P W Duin (1999). “Support vector domain description”. İn: *Pattern Recognition Letters* 20.11-13, pp. 1191–1199. ISSN: 01678655. DOI: 10.1016/S0167-8655(99)00087-2.

— (2004). “Support Vector Data Description”. İn: *Machine Learning* 54.1, pp. 45–66. ISSN: 08856125. DOI: 10.1023/B:MACH.0000008084.60811.49.

Veropoulos, Konstantinos, Colin Campbell, Nello Cristianini, vd. (1999). “Controlling the sensitivity of support vector machines”. İn: *Proceedings of the international joint conference on artificial intelligence*, pp. 55–60. DOI: 10.1.1.42.7895. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7895$%5Cbackslash$nhttp://seis.bris.ac.uk/%7B~%7Denicgc/pubs/1999/ijcai%7B%5C_%7Dss.pdf.

Wang, Deliang ve Geoffrey Hinton (1999). “Unsupervised learning: Foundations of neural computation”. İn: *Computers & Mathematics with Applications* 38.5-6, p. 256. ISSN: 08981221. DOI: 10.1016/S0898-1221(99)90165-7.

Wang, Ziheng, Ryan M. Hope, Zuoguan Wang, Qiang Ji ve Wayne D. Gray (2011). “An EEG workload classifier for multiple subjects”. İn: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 6534–6537. ISSN: 1557170X. DOI: 10.1109/IEMBS.2011.6091612.

Wiggins, M., A. Saad, B. Litt ve G. Vachtsevanos (2008). “Evolving a Bayesian classifier for ECG-based age classification in medical applications”. İn: *Applied Soft Computing Journal* 8.1, pp. 599–608. ISSN: 15684946. DOI: 10.1016/j.asoc.2007.03.009.

Yu, Lei ve Huan Liu (2004). “Efficient Feature Selection via Analysis of Relevance and Redundancy”. İn: *Journal of Machine Learning Research* 5, pp. 1205–1224. ISSN: 15324435. DOI: 10.1145/1014052.1014149. URL: http://portal.acm.org/citation.cfm?id=1044700.

Zhang, Shichao, Chengqi Zhang ve Qiang Yang (2003). “Data preparation for data mining”. İn: *Applied Artificial Intelligence* 17.5-6, pp. 375–381. ISSN: 0883-9514. DOI: 10.1080/713827180. URL: http://dx.doi.org/10.1080/713827180.