GUDERMANNIAN KAYIP FONKSİYONU VE GUDERMANNIANBOOST İKİLİ SINIFLANDIRMA YÖNTEMİ

GUDERMANNIAN LOSS FUNCTION AND GUDERMANNIANBOOST BINARY CLASSIFICATION METHOD

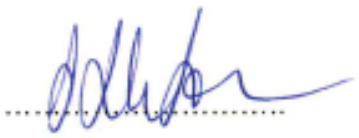
ONUR TOKA

PROF. DR. MERAL ÇETİN
Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
İstatistik Anabilim Dalı için Öngördüğü
DOKTORA TEZİ olarak hazırlanmıştır.

2016

ONUR TOKA'nın hazırladığı "**Gudermannian Kayıp Fonksiyonu ve GudermannianBoost İkili Sınıflandırma Yöntemi**" adlı bu çalışma aşağıdaki jüri tarafından **İSTATİSTİK ANABİLİM DALI**'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Prof. Dr. Aylin ALIN
Başkan ..... 

Prof. Dr. Meral ÇETİN
Danışman ..... 

Prof. Dr. M. Aydın ERAR
Üye ..... 

Prof. Dr. Olcay ARSLAN
Üye ..... 

Prof. Dr. Durdu KARASOY
Üye ..... 

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **DOKTORA TEZİ** olarak onaylanmıştır.

Prof. Dr. Salih Bülent ALTEN
Fen Bilimleri Enstitüsü Müdürü

Eşim GÜLNAZ'a

ve

Oğlum OZAN'a

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçların bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünün bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim

16.12.2016

ONUR TOKA

ÖZET

GUDERMANNIAN KAYIP FONKSİYONU VE GUDERMANNIANBOOST İKİLİ SINIFLANDIRMA YÖNTEMİ

Onur TOKA

Doktora, İstatistik Bölümü

Tez Danışmanı: Prof. Dr. Meral ÇETİN

Aralık 2016, 66 sayfa

Bu çalışmada bir sağlam kayıp fonksiyonu ve ilgili kayıp fonksiyonu üzerinden ikili boosting sınıflandırma yöntemi önerilmiştir.

Sınıflandırma yöntemlerinin amacı genelleştirme becerisine sahip iyi sınıflandırıcı elde etmektir. Boosting yöntemler kayıp fonksiyon ve basit sınıflandırıcılardan oluşur. İteratif algoritmalar olarak girdilere göre sınıf etiketlerini tahmin eder. Kayıp fonksiyonlar boosting algoritmalarda koşullu riski minimize eder. Birçok kayıp fonksiyon sadece yanlış sınıflandırmaya ceza verirken sağlam (robust) kayıp fonksiyonlar sadece yanlış sınıflandırmaya değil aynı zamanda doğru sınıflandırmaya da ceza vererek daha kararlı sınıflandırıcı elde eder. Sağlam kayıp fonksiyonlar, öğrenme verisindeki aykırı değerlere ve bozulmaların olduğu yapıya karşı dayanıklıdır. Bu yüzden, test kümesinde yüksek performansla çalışan yöntemlerdir.

Çalışma, kayıp fonksiyonlar, sağlam kayıp fonksiyonlar ve özellikleri hakkında özet bilgiler içermektedir. Sağlam kayıp fonksiyon özelliklerine sahip olan TanjantBoost algoritması

verilmiş, istatistiksel tutarlılık için algoritmada bir düzeltme yapılmıştır. Son olarak, TanjantBoost yönteminin sınıflandırıcıya yakın gözlemler üzerindeki etkisini başarılı hale getirmek için daha büyük ceza veren kayıp fonksiyon incelemesi yapılmış ve Gudermannian kayıp fonksiyonu ile boosting algoritması olan GudermannianBoost ikili sınıflandırma yöntemi önerilmiştir. GudermannianBoost yönteminin etkin olduğu durumlar farklı benzetim senaryoları ve bazı gerçek veri kümeleri üzerinden tartışılmıştır.

Anahtar Kelimeler: GudermannianBoost Yöntemi, Sağlam Kayıp Fonksiyonu, İkili Sınıflandırma, Boosting Yöntemler

ABSTRACT

GUDERMANNIAN LOSS FUNCTION AND GUDERMANNIANBOOST BINARY CLASSIFICATION METHOD

Onur TOKA

Doctor of Philosophy, Department of Statistics

Supervisor: Prof. Dr. Meral ÇETİN

December 2016, 66 pages

In this study, a robust loss function and binary boosting classification method are proposed.

The purpose of classification methods is to obtain classifier function with generalization ability, i.e., high prediction performance. Boosting methods as iteratively algorithms, which include a loss function and weak classifier, are a way of predicting class labels of given inputs. Loss function is the way of penalizing conditional risk in boosting algorithm. While most of loss functions only penalize the misclassification, some robust loss functions give penalties not only large negative (misclassification) margin but also large positive (accurate classification) margin in order to get more stable classifiers. Robust loss functions stand up to outliers and contaminated part in training data. Therefore, classifiers are the methods which display high prediction performance in testing part.

This study reports brief information about loss functions, robust loss functions and their properties. In addition, there is a correction to ensure statistical consistency on the algorithm of

TangentBoost, one of the methods having all properties of robust loss functions. Finally, in order to get more stable classifiers, Gudermannian loss function, which gives more penalties for both small positive and small negative margin than Tangent loss function does, and GudermannianBoost as a corresponding binary classification boosting method are proposed. The advantages of GudermannianBoost method are discussed based on the applications of some spesific simulation scenarios and some real datasets.

Keywords: GudermannianBoost Method, Robust Loss Function, Binary Classification, Boosting Methods

TEŞEKKÜR

Tez çalışması süresince desteklerini esirgemeyen danışmanım Prof. Dr. Meral ÇETİN'e,

Tez raporları süresince önerileri ve eleştirileri ile katkıda bulunan hocalarım Prof. Dr. Olcay ARSLAN ve Prof. Dr. M. Aydın ERAR'a,

Tez savunması süresince yapmış oldukları incelemeler, öneriler ve eleştiriler ile katkıda bulunan hocalarım Prof. Dr. Durdu KARASOY ve Prof. Dr. Aylin ALIN'a,

Doktora öğrenim süreci boyunca maddi olarak verdiği destekten dolayı Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na (TÜBİTAK'a),

Hacettepe Üniversitesi İstatistik Bölümü ailesine,

İlk adımımdan bugüne desteklerini hep hissettiğim annem Gülüzar, babam Mahir ve kardeşim Olcay'a,

Hayatıma şimdiye kadar verdiği ve bundan sonra vereceği anlam ve desteklerden dolayı eşim Gülnaz'a,

Ailemizin mutluluk kaynağı ve yaşam sevincimiz oğlum Ozan'a sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÇİZELGELER

Sayfa

ŞEKİLLER

Sayfa

SİMGELER VE KISALTMALAR

Simgeler

$\phi(\eta)$	Kayıp Fonksiyon
L(x,y)	Vekil Kayıp Fonksiyon
$C_{\phi}^{*}()$	Risk Fonksiyonu
$f(x)$	Ayırıcı Fonksiyon
$gd(\eta)$	Gudermannian Kayıp Fonksiyonu
$I(\eta)$	Beklenen Kazanç Fonksiyonu
$J(\eta)$	Maksimum Kazanç Fonksiyonu

Kısaltmalar

DVM	Destek Vektör Makineleri
GA	Genetik Algoritma

1. GİRİŞ

Bilim, Türk Dil Kurumu tarafından "Evrenin ya da olayların bir bölümünü konu olarak seçen, deneysel yöntemlere ve gerçekliğe dayanarak yasalar çıkarmaya çalışan düzenli bilgi" olarak tanımlanmıştır. Bilimin ortaya çıkmasının en büyük sebeplerinden biri, bilme isteğidir. Bilinmeyene ulaşmak ve bilinenler sayesinde hayatı kolaylaştırmak isteği bilimin saklı kalmış amaçlarındandır. Bilimsel çalışmaların genişlemesi, bilimde daha detaylı bilgilerin elde edilebilmesi adına bilimin alt dalları oluşturulmuştur. Bilim, ilgi alanlarına ve çalışmaların benzerliklerine göre sınıflandırılarak çeşitli başlıklar altında incelenmiştir. İstatistik, bir bilim dalı olarak geçmişte elde edilen deneyimlerle geleceğe ait tahminler ve yorumlar yapma, geleceğe şekil verme amacını taşır. Ayrıca istatistik, diğer disiplinlerin deneyimler ve gözlemler sonucu ortaya attığı hipotezleri sınar, gelecekte yaşanabilecek olayların ve olguların önceden tahmin edilebilmesi amacını taşır. 16. yüzyıl ortalarında haftalık ölüm oranlarının açıklanması, 17. yüzyılda nüfus ve ekonomik verilerin kayıt altına alınması süreçleri istatistiğin başlangıcı olarak görülmüştür. İstatistiksel çıkarsama ve tahmin yöntemlerinin ortaya atılması, diğer bilim dallarında ortaya konulan hipotezler üzerinden geleceği çıkarsamada aktif olarak kullanılması, istatistiğe bir bilim dalı olma özelliğini kazandırmıştır.

Günümüz dünyasında teknolojik özelliklerinin gelişmesi, verilerin anlık tutularak veri tabanlarına aktarılması, düzenli bilgiler halinde incelenmesi ve yorumlanması, istatistiğin gelişimine katkı sunmuştur. İstatistik, hızla gelişen teknolojinin sonucunda ortaya çıkan devasa veri kümelerin üzerinde çalışmaya başlamış, çok boyutlu verilerin değişkenleri arasındaki ilişkilerden faydalanarak sınıflandırma, kümeleme ve boyut indirgeme problemlerinin başlıca çözüm kaynağı olmuştur. Sınıflandırma özelliğine sahip istatistiksel yöntemler, bir deney sürecin başarılı ya da başarısız olması, deneğe konulacak tanının hasta, hasta değil biçiminde belirlenebilmesi ya da kurumların küçük, orta ve büyük işletme olabileceği gibi birçok alanda kullanılabilir. Ayrıca sınıflandırma yöntemleri sayesinde aykırı değerlerin belirlenmesi ve incelenmesi önem arz etmektedir. Sınıflandırma yöntemleri üzerinden aykırılıklar incelenerek riskli kredi belirleme, az görülen hastalıklarda risk belirleme, istenmeyen e-posta belirleme (spam mail detection) ve sahtekârlık belirleme (fraud detection) analizleri yapılmaktadır. İstatistiksel literatürde sınıflandırma ile ilgili önerilmiş ve geliştirilmekte olan birçok yöntem vardır. Sınıflandırmada yeni önerilerin ortaya atılması birçok alanda kullanılan yöntemlerin daha hızlı şekilde daha az hatayla sınıflandırma yapmasını amaçlamaktadır.

Bu çalışmanın birinci bölümünde istatistiksel öğrenme ve sağlam ayrım tabanlı bazı çalışmalar açıklanmıştır. İstatistiksel öğrenme, matematiksel olarak ilişkilerinin ortaya konulduğu bir süreçtir. Tahmin sürecinin ilk adımı olduğundan bu çalışmada kısaca özetlenmiştir. Ayrım tabanlı sınıflandırıcıların kayıp fonksiyonlar üzerinden ortaya çıkartılması, aykırılıkların öğrenme kümesindeki etkisinin azaltılması ile ilgili çeşitli öneriler de bu bölümün içerisinde yer almıştır.

İkinci bölümde sınıflandırma yöntemlerinde kullanılan bazı kayıp fonksiyonlar verilmiştir. Klasik ve sağlam sınıflandırma yöntemlerinde kullanılan, istatistiksel özelliğe sahip olan ve metodolojinin içinde sıklıkla kullanılan kayıp fonksiyonlar yapıları itibariyle yüzeysel olarak incelenmiştir.

Üçüncü bölümde boosting yöntemler, yöntemlerde kullanılan sağlam kayıp fonksiyonlar verilmiştir. Ayrıca sağlam boosting algoritmalar için gerekli kayıp fonksiyon özellikleri tartışılmıştır. Trigonometrik fonksiyonlardan elde edilmiş TanjantBoost yönteminde olasılık değerlerini 0-1 aralığına almak için bir düzeltme yapılmıştır. TanjantBoost yönteminin sonuçları incelendiğinde sınıflandırıcıya yakın gözlemlere daha hızlı artan ceza verebilecek ve benzer özellikleri taşıyacak olan bir kayıp fonksiyon araştırılmıştır. Bu noktadan hareketle Tanjant kayıp fonksiyon önerisine benzer bir kayıp fonksiyon, Gudermannian kayıp fonksiyon, önerilmiştir. Sağlam kayıp fonksiyonu özellikleri taşıyan Gudermannian kayıp fonksiyonu için regresyon tabanlı boosting algoritması, LojitBoost'a benzer algoritma, GudermannianBoost önerilmiştir. GudermannianBoost yöntemi ikili sınıflandırma problemleri için ayrılabilir veri kümelerinde kullanılmaktadır. Yöntem, benzer bir yöntem olan TanjantBoost'un sınıflandırıcıya yakın yanlış sınıflandırılmış gözlemlere duyarlılığını azaltmaktadır. Dolayısıyla önerilmiş olan kayıp fonksiyon ve LojitBoost mantığından hareket ederek önerilen GudermannianBoost yöntemi, aykırılıkların ve yanlış sınıflandırmaların algoritma adımlarındaki basit sınıflandırıcıya etkilerini azaltmaktadır.

Çalışmanın dördüncü ve son bölümünde ise benzetim ve gerçek veri kümeleri ile uygulamalar yapılmış, GudermannianBoost'un etkin olduğu noktalar açıklanmıştır. Benzetim verileri kullanılarak TanjantBoost yönteminden etkin olabileceği durumlar araştırılmıştır. Ayrıca gerçek veri kümeleri kullanılarak diğer boosting algoritmalarla olan farklılıkları incelenmiştir.

Çalışmada önerilen yeni yöntem, öğrenme kümesindeki iyileştirme sonucunda test kümesinde daha başarılı sonuçlar vermiştir.

1.1. İstatistiksel Öğrenme

Öğrenme kavramı birçok alanda farklı ifadeler ile açıklanabilmektedir. Psikoloji bilimi, öğrenmeyi aktif yaşananlar sonucunda meydana gelen bir kavram olarak açıklamaktadır. Eğitim bilimi ise okuma ya da yaşama eylemleri sonucunda ortaya çıkan bilinçli ya da bilinçsiz olarak düşüncede meydana gelen değişiklikler olarak açıklamaktadır. Sosyolojide Bandura'nın sosyal öğrenme ile ilgili ortaya atmış olduğu kuram, bireylerin yaşayarak öğrenmiş olduğu iyi şeyleri taklit etmesi, kötü şeylerden ise uzak durması olarak açıklanmaktadır. Kavram olarak bilimler arasında farklı şekilde açıklamalara sahip olsa da öğrenme, ortaya çıkan bir durumun gözden geçirilerek kullanışlı hale getirilmesi olarak tarif edilebilir. İstatistiksel öğrenme, yukarıda kavram olarak açıklanan tüm yöntemlerin matematiksel formüle dökülmesidir. Öğrenme veri kümesi (training data set) (geçmişteki deneyimler) kullanılarak elde edilen fonksiyonel ilişkiler (düşüncedeki değişimi yaratma) daha sonrayı tahmin etmekte (elde edilecek kararları, durumları açıklamakta) kullanılır. İstatistiksel öğrenmede asıl amaç tahmindir. İstatistiksel öğrenmenin en basit tanımı, elde edilmiş, gözlenmiş verilerdeki girdi çıktı ilişkilerini modelleyerek gelecekte elde edilecek bilgilerde çıktıları tahmin etmektir.

Matematiksel gösterimlerle açıklanmak istendiğinde (x, y) çiftleri hem geçmiş gözlemlerde hem de gelecek gözlemlerde aynı ancak genellikle bilinmeyen $X \times Y$ üzerinde tanımlı P dağılımından birbirlerinden bağımsız olarak elde edilmektedir. Verilerin ortaya çıkma süreci iki adımla açıklanabilir. Önce girdi değeri x bilinmeyen P_X marjinal dağılımından elde edilir. Daha sonra elde edilen x değerleri için $P(.|X)$ koşullu olasılığıyla Y üzerinden y çıktıları üretilir. Burada $P(.|X)$ koşullu olasılığının bilinmemesi girdi ve çıktı değerleri arasındaki nedensel açıklamanın bilinmemesi anlamına gelmektedir. Bu süreç geçmişte elde edilmiş verilerin ortaya çıkartılma mantığıdır. Bir sonraki adımda ise öğrenme durumu ortaya çıkartılır. Yani, girdi ve çıktı çiftleri (x, y) için tahmin fonksiyonu $f(x)$ elde edilerek fonksiyonun kalitesi belirlenmelidir. $f(x)$, girdi değişkenleri x ile çıktı y'nin ilişkisel fonksiyonudur ve y çıktısının ileride alacağı değerleri tahmin eder. İstatistiksel öğrenmenin bu kısmı fonksiyonun tahmin sürecidir. Sürecin diğer bir adımı, riskin minimize edilmesidir.

En iyi fonksiyonu elde etmek için girdiler-çıktılar sonucunda elde edilen $f(x)$ tahmini ile çıktı değerleri arasındaki uzaklık, daha kavramsal olarak kayıp (loss), ölçülmelidir. Kaybın beklenen değeri, bilinmeyen dağılımdan gelen verilerle elde edilen tahminin riskini ölçer. Sürecin bir sonraki adımında ise öğrenme problemleri vardır. Genel olarak öğrenme problemleri örüntü tanımlama, regresyon ve yoğunluk (density) kestirimi başlıkları altında incelenebilir. Sürecin devamı deneysel riskin minimize edilmesidir [1].

Alt başlıklarda fonksiyonun elde edilişi, tahmin, kayıp ve risk kavramları açıklanmıştır. Ancak daha öncesinde literatürde yapılan çalışmalar ve çalışmaların katkıları ile ilgili bilgiler verilmiştir. Daha sonra ise sağlam kayıp fonksiyonları ve kullanılabileceği yöntemlerle ilgili çalışmalar açıklanmıştır. Devamında kayıp fonksiyonlar, riskler ve sağlam boosting sınıflandırma yöntemleri ilerleyen başlıklarda daha detaylı olarak ele alınmıştır. Çalışmada son olarak, Gudermannian kayıp fonksiyon önerisi, özellikleri, koşulları ve boosting algoritması (GudermannianBoost) verilmiştir.

1.2. Ayrım Tabanlı Sınıflandırma Yöntemlerinde Bazı Sağlam Öneriler

İkili sınıflandırma yöntemlerinde ayrım (margin) tabanlı yöntemler hem zayıf sınıflandırıcılar (weak classifier) hem de destek vektör makineleri (DVM) ile kullanılmaktadır. Birçok bilim dalında sınıflandırma yöntemleri geliştirilmektedir. Özellikle sinyal tanılama, görüntü elde etme gibi uygulama alanlarında ikili sınıflandırma yöntemleri ile ilgilenilmesi istatistiksel tabanlı olmayan birçok yöntemi ortaya çıkartmıştır. Ancak, istatistikçilerin önerilerinde, bayes tutarlılık, tutarlılık gibi istatistiksel; bozulma noktası, etki fonksiyonu gibi sağlam istatistiğin önemli özellikleri üzerinden yöntemler geliştirilmiştir. Son dönemlerde yapılan çalışmalarda kayıp fonksiyonların aykırı değerlerden etkilenmemesi için çeşitli öneriler getirilmektedir. Ayrıca, zayıf sınıflandırıcıların ortaya çıkarmış olduğu ayırıcı fonksiyonun (sınıflandırıcının) test kümelerinde yüksek sınıflandırma başarısı elde etmesi ile ilgili çalışmalar yapılmaktadır. Öğrenme kümelerinde aykırı gözlemlerin olması, özellikle zayıf sınıflandırıcılarda sorunlar yaratmakta, bu sorunların aşılması için çeşitli yöntemler önerilmektedir.

Wu ve Liu [2] çalışmasında, DVM'nin kayıp fonksiyon olarak kullandığı hinge kayıp ölçümünün, aykırı değerlere karşı duyarlı olduğu belirtilmiştir. Veri kümesindeki aykırı değerlerin gereğinden daha çok destek vektör oluşmasına sebep olduğu açıklanmıştır.

Sorunların çözümlenmesi için kesilmiş hinge kaybı önerilmiştir. Önerilen yöntemin klasik DVM yöntemine göre daha sağlam ve istatistiksel olarak tutarlı olduğu gösterilmiştir.

Xu vd. [3] çalışmasında, aşırı uyumu engellemek için düzenlenmiş DVM yöntemi önerilmektedir. Yapılan sağlam sınıflandırma önerisinin aslında DVM'nin cezalandırıcı terimi üzerine getirilen kısıtlama olduğu görülmüş ve yeni önerilerin ceza terimi üzerinden geliştirilebileceği belirtilmiştir.

Carrizosa ve Morales [4], matematiksel optimizasyon yöntemlerinin sınıflandırma yöntemleriyle birleştirilmesinin değişkenleri tanımlamada, sınıflandırıcıları yorumlamada ve veri kümesindeki aykırılıklara cevap vermede etkin olduğunu belirtmişlerdir. Katıştırma (embedding) yöntemleri ile DVM'yi birleştirerek daha etkin sonuçlar elde edilebileceği gösterilmiştir.

Park [5] çalışmasında, ayrım tabanlı sınıflandırıcılar verilmiştir. Karar fonksiyonu ile yanlış sınıflandırmayı minimize edecek şekilde sınıflandırıcı elde edilmeye çalışılmaktadır. Yanlış sınıflandırma tek başına bir amaç olarak alındığında birçok sınıflandırıcı aşırı uyum sorunu yaşamaktadır. Bu problemi çözmek için amaç fonksiyonuna düzenleme ya da büzme (shrinkage) uygulanabilir. Genelleme hatasını minimize etmek, modelin uyumunu kontrol eder. Cezalandırıcı terim iyi bir genelleme yapabilmek adına aşırı uyumu engellemeye çalışır. DVM, cezalandırıcı lojistik regresyon, ağırlıklandırılmış ayrıştırma analizi gibi yöntemler, farklı kayıp fonksiyonlar kullandığından farklı sınıflandırıcılar ve farklı çözümlemeler elde eder. Uyarlanabilen kayıp fonksiyon, yönteme avantaj sağlarsa sınıflandırmada etkinlik arttırılabilir.

Gupta [6] çalışmasında, veri kümesinin arttırılamadığı durumda mevcut sınıflandırma yöntemlerinin başarısız sonuçlar verdiği belirtilmiştir. Özellikle kanser çalışması gibi önemli durumlarda, sınıflandırma yönteminin yanlış sınıflandırma seçme ihtimali göz önünde bulundurulduğunda problemin ne kadar önemli olduğu görülebilir. Önerilen yöntem, verideki gerçek kayıp fonksiyonlarını daha iyi sunabilmeyi ve aykırı değerden etkilenmeden çözüm elde etmeyi amaçlamaktadır.

Ma vd. [7], önerilen algoritmanın diğer sağlam sınıflandırma yöntemlerinden daha hızlı çalıştığını belirtmişlerdir. Ayrıca öğrenme kümesindeki aykırı değerlerin, uygulamada doğru sınıflandırma kurallarının belirlenmesini engellediği gösterilmiştir.

Song vd. [8], öğrenme kümesindeki veri noktalarına verilecek ağırlıkların merkezden uzağa gittikçe azaltılacak şekilde olmasını önermişlerdir. Ancak veri dağılımı için model varsayımları gereklidir. DVM, ramp-kayıp ile ayrımları ihlal eden tek nokta için maksimum cezalandırıcıya sınır konularak sağlamlaştırılmıştır. Bu durum, iki paralel hinge kayıp fonksiyonunu çıkararak ve bazı ikili doğrusal fonksiyonlarla sağlanmıştır [9,10]. Bu yöntemlerin en önemli eksileri konveksliğin sağlanmamasıdır.

Debruyne [11] çalışmasında, çekirdek fonksiyonlar üzerinde yapılan çalışma verilmiş, elde edilen sağlam DVM yöntemi tartışılmış, Stahel Donoho aykırılık belirleme yönteminin faydası açıklanmış ve çok boyutlu veri kümesinde görselleştirme üzerinde uygulamalar yapılmıştır.

Mai [12] çalışmasında, değişken sayısının gözlem sayısından fazla olduğu durumlarda seçim yöntemlerinin sınıflandırma yöntemleriyle birlikte kullanılabileceği gösterilmiştir. Etkin sınıflandırıcılar oluşturulurken değişken sayısının gözlem sayısını geçmesi, aykırı değerlerin bulunması ve bu etkilerden dolayı hesaplama zamanının çok uzun süreli olması büyük problemdir. Önerilen yöntem, değişken seçimi için kullanılan LASSO'nun cezalandırma yöntemi üzerinden seyreklik (sparsity) analizi yapmasıdır.

Buckstein [13] çalışmasında, model seçimi, sınıflandırma ve boyut indirgeme için kullanılan istatistiksel yöntemlerde aykırı değerlerden kurtulabilmek için genel seyreklik kontrolleri uygulanmıştır. LASSO'nun cezalandırıcı parametresi üzerinden seyreklik kontrolü yapılmaktadır. Ayrıca seyreklik analizi yapılırken aslında sınıflandırma yöntemleri için aykırı değerlerin belirlenmesi sağlanmaktadır. RANSAC yöntemi, altkümelerde elde edilen modellerin tutarlı olana kadar incelenmesi üzerine kurulmuştur. Seyrekliğin incelenmesi, karışıklığın azaltılmasında kullanılmaktadır.

Lim [14] çalışmasında, gruplama yöntemi önerilmiştir. Çalışmanın en önemli özelliği altküme seçimleri üzerinden sınıflandırma yaparak bu sınıfların birleştirilmesidir. Ancak altküme seçimleri farklı altkümeler üzerinden yapılmakta ve değişken seçim yöntemleri bu sürece dâhil edilmemektedir.

Oh [15] çalışmasında, boosting yöntemlerin sınıflandırma üzerindeki tahmin kesinliğini geliştirmek amacıyla öneride bulunulmuştur. Boosting yöntemlerde en önemli problem, aşırı uyuma yatkın olmalarıdır. GA-Boosting yöntemi, zayıf öğrenicilerin ve ağırlıklarının, genetik algoritma üzerinden geliştirilmesi üzerine kurulmuş ve önerilmiştir. Çalışmada lojistik

regresyon ve ayrıştırma analizi, varsayımlarından dolayı eleştirilmiştir. Karar ağaçları açıklama ve görsellik bakımından tercih edilebilir ve ilişkileri açıklamada oldukça başarılı olarak betimlenirken katıştırmalı modeller kadar etkin olmadıkları belirtilmiştir. DVM ve ayrım sınıflandırıcı yöntemler, hinge kaybını ya da benzer yapıdaki kayıp fonksiyonları optimize etmeye çalışmaktadır. Aykırı değerlere karşı dayanıklı olmak bu çalışmanın konusu olmuştur. DVM gibi modern yöntemlerle yeni bir sınıflandırıcıyı birleştirmek yerine, kayıpların gerçek durumunu ortaya koyabilen bir kayıp fonksiyonu üretmenin analitik olarak daha kolay olduğu açıklanmıştır.

Verinin boyutlarına göre küçük örneklemlerde çekirdek fonksiyonun karmaşıklığı ve çok boyutlu uzayda projeksiyon çalışması aşırı uyum problemine sebep olabilmektedir[16, 17]. Özellikle biyolojideki deneysel tasarımlarda mikro-dizi verilerin gözlem sayılarının az olması boyut azaltma yöntemlerine sebep olmaktadır. Roweis ve Saul [18] çalışmasındaki gibi az boyutlu uzaylarda çalışılan komşu katıştırma yöntemleri için boyut azaltma çalışmaları önemlidir.

Ren ve Dai [19] çalışmasında, regresyon tabanlı bir sınıflandırma yöntemi önerilmiştir. Önerilen yöntem, büyük varyanslarla dağılan sınıflandırma problemlerinde aykırı değerlere karşı sağlam bir yöntemdir.

Chi [20], LASSO'nun örneklem sayısının küçük, değişken sayısının çok olduğu durumda yaptığı çözümlemelerin başarılı olmasından yola çıkmıştır. Değişken sayısı ve dolayısıyla verinin boyutu arttıkça verideki bozulumları incelemek, aykırı değerleri belirlemek daha zor olmaktadır. Bu çalışmada aykırı değerler değişken seçim yöntemini problemli hale getirmektedir. Sorunu giderebilmek için iki düzeyli cevap değişkeninde minimum uzaklık kestiricisi önerilmiştir.

Nudurupati [21] çalışmasında, iyi bir sınıflandırıcı önerilirken düşük bir sınıflandırma hatası, aynı zamanda grup sayısı, gözlem sayısı, grup yapısı ve şekli gibi çeşitli durumların göz önünde bulundurulması gerektiği belirtilmiştir. Projeksiyon izleme (projection pirsuit) yöntemi üzerinden verilen öneriler gözden geçirilmiştir. Önerilen yöntemlerin tamamı sağlam istatistikler yardımıyla geliştirilmiştir.

Literatürde sağlam yaklaşımların önerildiği önemli bir yöntemler topluluğu ise boosting algoritmalardır. Gerek klasik gerekse sağlam yöntemler kullanılarak zayıf öğrenicilerin (weak

learners) elde edilmesi, sürecin sonunda zayıf öğrenicilerin birleştirilmesiyle güçlü bir sınıflandırıcı bulunması amaçlanmaktadır. AdaBoost [22] algoritması üstel hatayı minimize etmeye çalışır ve zayıf öğreniciler ile güçlü bir sınıflandırıcı yaratmaya çalışan, hızlı ve etkili yöntemlerden en çok bilinenidir. LojitBoost [23], binom sapmayı minimize ederek çalışmaktadır. Zayıf öğreniciler, regresyon modeli üzerinden olasılık değerleri elde etmektedir. Sınıflar ise bu olasılık değerleri üzerinden belirlenmektedir. GentleBoost [23] aynı zamanda Gentle AdaBoost olarak da bilinen, AdaBoost ve LojitBoost yöntemlerini birleştiren bir boosting algoritmasıdır. AdaBoost gibi üstel kaybı kullanırken aynı zamanda LojitBoost gibi regresyon modeli üzerinden zayıf öğrenicileri elde etmeye çalışmaktadır. RobustBoost [24] algoritması, AdaBoost mantığı ile çalışır. Farkı ise aşırı yanlış sınıflandırmaların bulunduğu durumlara ağırlık vermeyerek aykırı değerlere karşı duyarsızlığı sağlamasıdır. RealBoost [23] ya da Real AdaBoost yöntemi ise sınıflara ait olma olasılıkları üzerinden hareket ederek, analitik olarak minimizasyonu sağlamaktadır. Elde edilen olasılık ve minimizasyon problemi ile sınıf etiketleri tahmin edilmektedir. BrownBoost [25] yöntemi konveks olmayan bir fonksiyon kullanmaktadır ve denklem sistemi çözümünden hareket etmektedir. Zayıf öğrenicilerin sürekli olarak yanlış sınıflandırıldıkları durumlarda ilgili gözlemi sınıflandırma dışında tutarak AdaBoost yönteminde iyileştirme denenmiştir.

Masnadi-Shirazi [26] çalışmasında, kayıp fonksiyonlarının hem Bayes tutarlı özelliğinden hem de ayrım tabanlı sınıflandırma yöntemlerinden bahsedilmektedir. Bayes tutarlı kayıp fonksiyonu başarısı üzerinde durulmuştur. Ayrıca kayıp fonksiyonlarına savage kayıp ve tanjant kayıp önerileri getirilmiş ve bu önerilerin sağlam özellikleri açıklanmıştır. Tanjant kayıp fonksiyonu için LojitBoost algoritmasına benzer bir algoritma önerilmiştir. Konveks olmayan bir kayıp fonksiyonu, bilinen risk fonksiyonu üzerinden bu algoritma ile minimize edilmektedir. Ancak önerilen TanjantBoost algoritması olasılık değerlerini 0-1 arasında bulamamaktadır. Kobetski ve Sullivan [27] olasılık değerleri üzerindeki sorunu görüp w ağırlıkları üzerinden çözüm üretmişlerdir. Bu çalışmada, LojitBoost mantığı değiştirilmeden çözüm için algoritmadaki olasılık değerleri alt ve üst sınır değerleri yardımıyla 0-1 aralığına alınmıştır.

Ayrım tabanlı sınıflandırıcıların amaç fonksiyonunda kayıp fonksiyonun önemli bir yer tutması ve sağlam önerilerin kayıp fonksiyonlarla incelenmesi nedeniyle bundan sonraki bölümde kayıp fonksiyon hakkında bilgi verilecek, sınıflandırma yöntemlerinde kullanılan kayıp fonksiyonlar ve özellikler incelenecektir.

2. KAYIP FONKSİYONU

2.1. Kayıp Fonksiyonu

Risk miktarı, $L: X \times Y \times R \rightarrow [0, \infty)$ olmak üzere $f(x)$ tahmin değerinin beklenen kaybı kullanılarak ölçülebilir. Bu durumda $f(x)$'in riski,

$$R_{L,P}(f) = \int L\big(x, y, f(x)\big)\, dP(x, y) = \iint L\big(x, y, f(x)\big) dP(y|x) \tag{2.1}$$

şeklinde elde edilir. $x_{n+1}, x_{n+2}, \dots, x_m$ yeni verileri için ortalama deneysel kaybın da mümkün olduğunca küçük olması gerekmektedir. Büyük sayılar yasası gereğince $m \rightarrow \infty$ durumunda ortalama deneysel kayıp $R_{L,P}(f)$'ye yakınsar. Yani risk, f fonksiyonu için iyi bir ölçümdür. Sınıflandırma durumlarında $f(x)$'in gerçek tahmini $f^*(x)$'e mümkün olduğunca yakınsaması gerekmektedir. Örnek olarak bir kayıp fonksiyonu, $p > 0$ olmak üzere, $L_p\big(x, y, f(x)\big) = |f^*(x) - f(x)|^p$ şeklinde ifade edilebilir. Bu durumda ortalama kayıp yani risk,

$$R_{L_p,P}(f) = \int |f^*(x) - f(x)|^p \, dP_X(x) \tag{2.2}$$

şeklinde ifade edilebilir. Bu durumda bütün mümkün fonksiyonların infimumu en küçük risktir:

$$R^*_{L,P}(f) = \inf_{f: X \rightarrow R} R_{L,P}(f) \tag{2.3}$$

Özetlenmesi gerekirse amaç, $f: X \rightarrow R$ fonksiyonu ile $R_{L,P}(f)$ riskini minimize etmektir [28].

$$R_{L,P}(f_D) \underset{n \rightarrow \infty}{\rightarrow} R^*_{L,P}(f) \tag{2.4}$$

Ayrım tabanlı sınıflandırma yöntemlerinde beklenen kaybı, riski, bulmak için dağılımla ilgili gerekli ek bilgi P dağılımının bilinmediği varsayımı altında $D: \{(x_1, y_1), \dots, (x_n, y_n)\}$ veri kümesinden elde edilmektedir. Bu durumda P bilinmediğinden $R_{L,P}(f)$ bilinemez ve f^* doğrudan bulunamaz. D sonlu örneklem çiftleri için $R_{L,P}(f)$, deneysel karşılığı $R_{L,D}(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))$ ile yer değişir. Ancak her bir f için deneysel risk kuramsal riske yakınsa da yaklaşık minimumu vermez. $R^*_{L,P}(f) = \inf_{f: X \rightarrow R} R_{L,P}(f)$ minimizasyonunun sağlanmadığı durumda D veri kümesinde sınıflandırma başarılı olabilir ancak yaklaşık minimum için minimizasyonu sağlamadığından daha sonra elde edilecek veriler için gerekli olan sınıflandırma tahminlerinde başarısız olur ve bu durum aşırı öğrenme (overfitting) olarak adlandırılır. Aşırı

öğrenme durumundan kurtulmak için öncelikle $f: X \to R$ fonksiyonlarının küçük bir F kümesi seçilir. Seçilen fonksiyonlar minimum riske yakınsama sağlayanlardır. Dolayısıyla bütün fonksiyonlar üzerinden $R_{L,D}(f)$'yi minimize etmek yerine sadece F üzerinden minimizasyon sağlanacak şekilde $\inf_{f \epsilon F} R_{L,D}(f)$ çözümü yapılır. Bu yaklaşım deneysel risk minimizasyonu (empirical risk minimization-DRM) olarak adlandırılır ve $\inf_{f \epsilon \mathcal{F}} R_{L,D}(f)$'in sonlu örnekleme karşılık gelen yaklaşık sonuçlarını üretmektedir ($R^*_{L,P,F}(f) = \inf_{f \epsilon F} R_{L,D}(f)$).

DRM'nin iki olumsuz yönü bulunmaktadır. İlk olarak F'yi tanımlayacak zenginliğe sahip bir dağılım (P) bilgisi bulunmamaktadır. Yani model hatasının veya yakınsama hatasının yeterli derecede küçük olduğunu garanti edemez. Bu problemi giderebilmek için örneklem büyüklüğü n ile f küme büyüklüğü arttırılarak yakınsama hatası azaltılır. İkinci olumsuzluk ise $\inf_{f \epsilon F} R_{L,D}(f)$ çözümsüz olabilir. Bu durumda $R_{L,D}()$ riski çözüme daha uygun olan temsili riskle değiştirilebilir. Kayıp fonksiyonlar, ayrım tabanlı sınıflandırıcılarla bu noktada kullanılabilir hale gelmektedir.

2.2. Sınıflandırma Yöntemlerinde Kullanılan ve Önerilen Kayıplar

Bu bölümde sınıflandırma yöntemlerinde kullanılan kayıp fonksiyonlar ele alınmıştır. $L: X \times Y \times R \to [0, \infty)$ ölçülebilir olduğu durumda kayıp fonksiyonu ya da kısaca kayıp olarak adlandırılmaktadır. Sınıflandırma yöntemlerine en uygun olan kayıp fonksiyonu 0-1 kayıp fonksiyonudur. Ancak, fonksiyon konveks olmayan ve türevlenemeyen yapısından dolayı çok uygulanabilir değildir [1]. Dolayısıyla temsili konveks kayıp fonksiyonlar önerilmiştir. Konveks yapının genelde tercih edilen yapı olmasının sebepleri tek optimum değerin olması; kullanım kolaylığı sağlaması ve konveks optimizasyon araçları ile çözümlenebilmesi olarak sıralanabilir. Konveks yapıda karşılaşılan en büyük sorun ise yanlış sınıflandırılmış olan gözlemlere çok fazla ceza verilmesidir. Üstelik karesel olarak artan kayıp fonksiyon değerlerinin monoton cezadan daha fazla ceza veriyor olması, sınıflandırıcının aykırı değerlerden etkilenmesine ve tek bir gözlemde bile sınıflandırıcının farklılaşmasına (kararlı olmamasına) neden olmaktadır. Dolayısıyla, aykırı değerlerin sınıflandırma üzerindeki etkisini azaltmak için kesilme (truncated) uygulanan bazı kayıp fonksiyonların ister istemez konveks yapısı bozulmaktadır. Bu

yüzden son dönemde yapılan çalışmalarda konveks olmayan kayıp fonksiyonlar önerilmektedir. Her kayıp fonksiyonun özellikleri genel olarak kendi başlıkları altında incelenecektir.

2.1.1. Sıfır-Bir, Ağırlıklı Sıfır-Bir Kayıp ve Düzleştirilmiş Sıfır Bir Kayıp

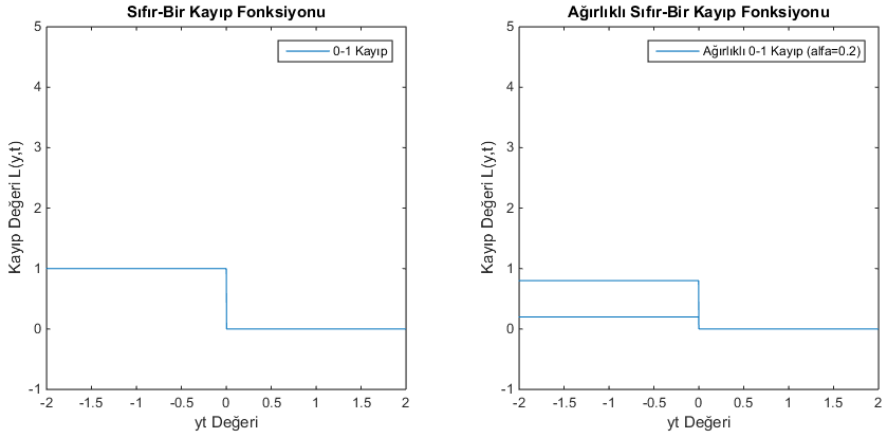
$Y = \{-1, +1\}$, P ise $X \times Y$ üzerinde tanımlı bilinmeyen dağılımdır. İkili sınıflandırmadaki amaç, P dağılımından gelen (x, y) çiftlerini kullanarak sadece x'i gözlenmiş durumların y etiket tahminini yapmaktır. Bu durum için $L_{0-1}: Y \times R \rightarrow [0, \infty)$ olmak üzere $y \epsilon Y$ ve $t \epsilon R$ iken sınıflandırma kaybı aşağıdaki gibi tanımlanır:

$$L_{0-1}(y, t) = 1_{(-\infty,0]}\big(ysign(t)\big) \tag{2.5}$$

Tahmin ile gerçek değerin işaretleri uyuşmadığı durumda bir cezalandırma yapan bu kayıp fonksiyon aslında öğrenme amacını basit bir şekilde açıklamaktadır. İkili sınıflandırmanın gerçek kayıp fonksiyonu olarak ifade edilir. Ağırlıklı ikili sınıflandırma durumunda ise $Y = \{-1, +1\}$ ve $\alpha \epsilon (0,1)$'dir. Bu durumda α-ağırlıklı sınıflandırma kaybı $L_{\alpha(0-1)}: Y \times \mathcal{R} \rightarrow [0, \infty)$ olmak üzere aşağıdaki gibi tanımlanır [28]:

$$L_{\alpha(0-1)}(y, t) = \begin{cases} 1 - \alpha, & y = 1 \; ve \; t < 0 \\ \alpha, & y = -1 \; ve \; t \geq 0 \\ 0, & \text{ö.} d. \end{cases} \tag{2.6}$$

Eşitlik (2.6) ve Şekil 2.1'den de anlaşılacağı gibi α-ağırlıklı sınıflandırma kaybı sınıflara göre farklı ağırlık vermektedir.

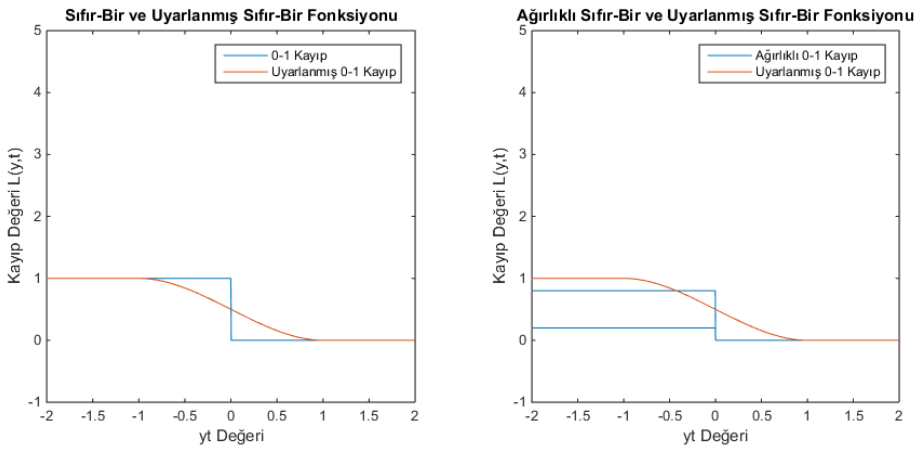


Şekil 2.1. Sıfır-Bir Kayıp Fonksiyonları 1

Sıfır-bir kaybına düzeltme işlemi uygulanarak limit ile ilgili sorun çözülmek istenmektedir. Düzleştirme işleminin yapılmasının sebebi ise türevlenebilir sıfır-bir kaybının oluşturulmasıdır.

Zhao vd. [29] çalışmasındaki uyarlanmış sıfır bir kaybı Şekil 2.1'den takip edilebilir ve $Y = \{-1, +1\}$, $L_{d(0-1)}: YX\mathcal{R} \rightarrow [0, \infty)$ olmak üzere aşağıdaki gibi tanımlanır:

$$L_{d(0-1)}(y,t) = \begin{cases} 0, & yt > 1 \\ \frac{1}{4}(yt)^3 - \frac{3}{4}yt + \frac{1}{2} & -1 \leq yt \leq 1 \\ 1, & yt \leq -1 \end{cases} \tag{2.7}$$



Şekil 2.2. Sıfır-Bir Kayıp Fonksiyonları 2

2.1.2. Karesel, Kesilmiş Karesel ve Uyarlanmış Karesel Kayıp

Karesel kaybın sıfır-bir kayıplarından ayırt edici en önemli özelliği doğru sınıflandırmaya da cezalandırma yöntemi uygulamasıdır. Eşitlik (2.8)'de verilen karesel kayıp, sınıflandırmanın doğru ya da yanlışlığına bakmaksızın sınıflandırıcıya uzak gözlemlere karesel şekilde artan bir ceza değeri vermektedir:

$$L_{Karesel} = (1 - yt)^2 \quad y = \pm 1, t \epsilon R \tag{2.8}$$

Karesel kayıp ile elde edilen optimizasyon probleminin elde edeceği sınıflandırma fonksiyonu çok değişken olabilmektedir. Bu durum aykırı değerlere karşı duyarlı olmanın haricinde sınıflandırma fonksiyonunun bir değerde bile kararlı davranamamasına sebep olmaktadır. Bu durumu engellemek için Eşitlik (2.9)'da verilen kesilmiş karesel kayıp önerisi getirilmiştir:

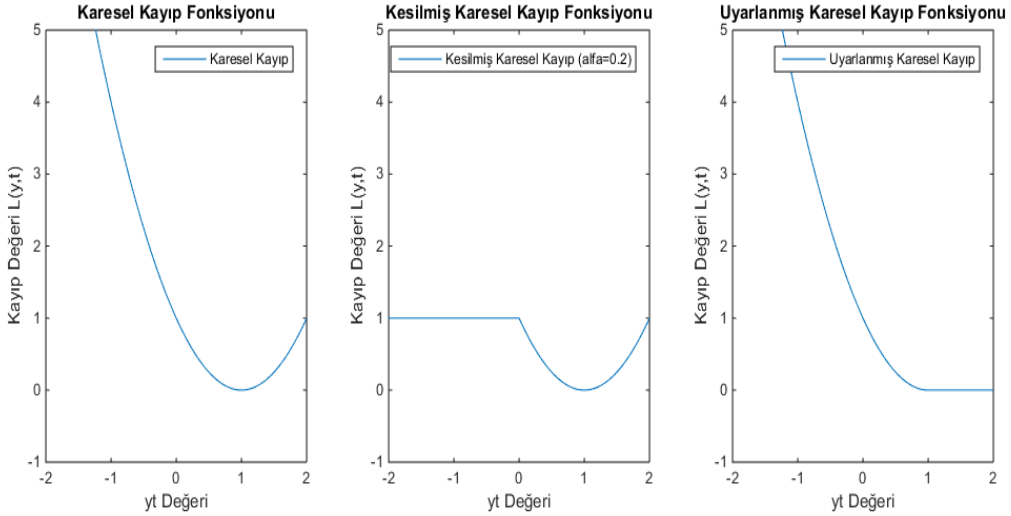
$$L_{K.\ Karesel} = \begin{cases} (1 - yt)^2, & (1 - yt) \leq 1 \\ 1, & (1 - yt) > 1 \end{cases} \quad y = \pm 1, t \epsilon R \tag{2.9}$$

Karesel kayıp fonksiyonun kesilmesi ile yanlış sınıflandırılmış aykırı değerlerin sınıflandırıcılara olan etkisi giderilmiştir. Ancak karesel kayıptaki diğer bir dezavantaj ise doğru

sınıflandırmanın ve yanlış sınıflandırmanın da eşit ceza almasıdır. Doğru sınıflandırmanın ceza değeri almaması gerektiği düşünülerek Eşitlik (2.10)'da verildiği gibi karesel fonksiyona bir uyarlama yapılmıştır:

$$L_{U.Karesel} = mak.(0,(1-yt))^2 \quad y = \pm 1, t \epsilon R \tag{2.10}$$

Kayıpların ilgili değerlere göre vermiş olduğu ceza Şekil 2.3'ten takip edilebilir.



Şekil 2.3. Karesel Kayıp Fonksiyonlar

Uyarlanmış karesel fonksiyon hinge karesel fonksiyon olarak da bilinmektedir. Doğru sınıflandırmada herhangi bir ceza verilmemekte ancak yanlış sınıflandırılmış bir gözlemin cezası karesel şekilde artırılarak devam etmektedir. Her ne kadar doğru sınıflandırmada cezanın sıfıra indirgenmesi tartışılabilir bir nokta olsa da gerçekte dikkat edilmesi gereken nokta yanlış sınıflandırılmış aykırı değerlerin kayıp fonksiyonu üzerinde baskın olarak sınıflandırıcıda etkin olmasıdır.

2.1.3. Hinge, Kesilmiş Hinge, Düzleştirilmiş Hinge ve Genelleştirilmiş Düzleştirilmiş Hinge Kaybı

Hinge, klasik DVM yönteminin temel almış olduğu kayıp fonksiyondur ve Şekil (2.11)'de verilmiştir. $L_{hinge}: Y \times R \rightarrow [0,\infty)$ için $yt \leq 1$ ile her tahmin t'yi doğrusal olarak cezalandırmaktadır.

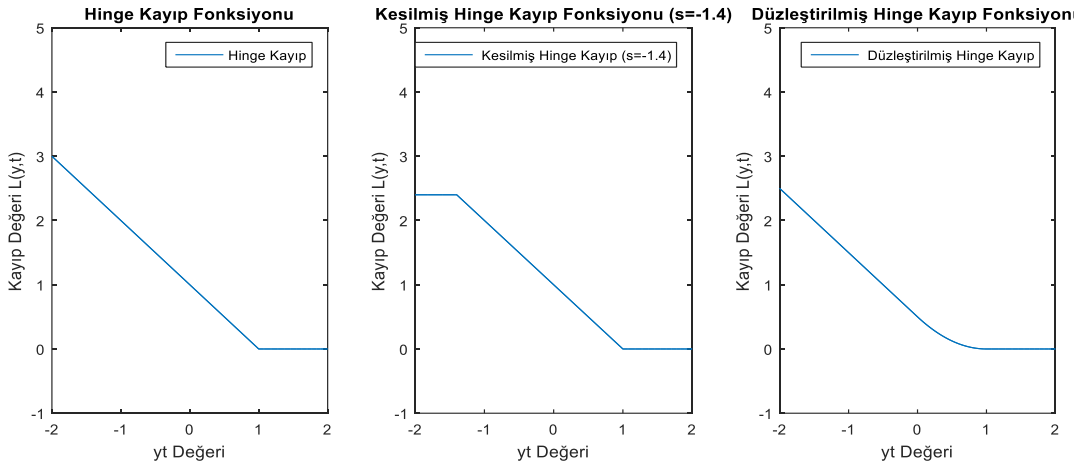
$$L_{hinge}(y,t) = max\{0, 1-yt\} \quad y = \pm 1, t \epsilon R \tag{2.11}$$

Hinge kaybı ile elde edilen sınıflandırıcının, doğrusal çekirdek fonksiyonu kullanımında aykırılıklara duyarlı olması, veri kümesindeki küçük değişimlerde bile kararlı davranmamasına sebep olduğu görülmüştür [30]. Rennie [31], hinge kaybında doğru sınıflandırmaya ceza verilmemesinin sınıflandırıcıya etkisinin olduğunu belirtmiştir. Zhang [32], çalışmasında hinge kaybının yumuşak (soft hinge) DVM yöntemindeki sınıflandırma kaybının en iyi temsilcisi olduğunu belirtmiştir. Ayrıca Wu ve Liu [2], çalışmalarında destek vektör sayısını azaltabilmek ve aykırı değer etkisini en aza indirgeyebilmek için kesilmiş hinge kaybını önermişlerdir. Kesilmiş hinge kaybı bazı kaynaklarda ramp-kaybı olarak geçmektedir [30].

Kesilmiş hinge fonksiyonu, $y = \pm 1, t \epsilon R, s < 0$ değerleri için $L_{H1}(y,t) = \max\{0, 1 - yt\}$ ve $L_{HS}(y,t,s) = \max\{0, s - yt\}$ olan iki fonksiyonun farkı olarak tanımlanmıştır ve Eşitlik (2.12)'deki gibi verilmiştir:

$$L_{K\_Hinge}(y,t,s) = L_{H1}(y,t) - L_{HS}(y,t,s) \quad y = \pm 1, t \epsilon R, s < 0 \qquad (2.12)$$

Kesilme ile belirlenen s $(s < 0)$ değerinden daha küçük tüm yt değerlerine aynı ceza verilir. Fonksiyonların ilgili değerleri Şekil 2.4'ten takip edilebilir.



Şekil 2.4. Hinge Kayıp Fonksiyonları 1

Bu sayede optimizasyon problemi yanlış sınıflandırılmış ve sınıflandırıcıdan çok uzakta elde edilmiş gözlemlerin etkisinden kurtulacaktır. Wu ve Liu [2], aykırı değer etkisinden arındırılmış sınıflandırıcıyı belirlemektedir. Ayrıca, değişken sayısının çok olduğu durumlarda sınıflandırıcıları belirleyen destek vektörlerin sayısının azaldığını göstermişlerdir. Çalışmada, kesilmiş hinge ile elde edilen destek vektörlerin, hinge kaybı ile elde edilen destek vektörlerin altkümesi olduğu da belirtilmiştir.

Hinge kaybı ile ilgili literatürde karesel hinge olarak isimlendirilen fonksiyon bu çalışmada uyarlanmış karesel kayıp olarak alınmıştır.

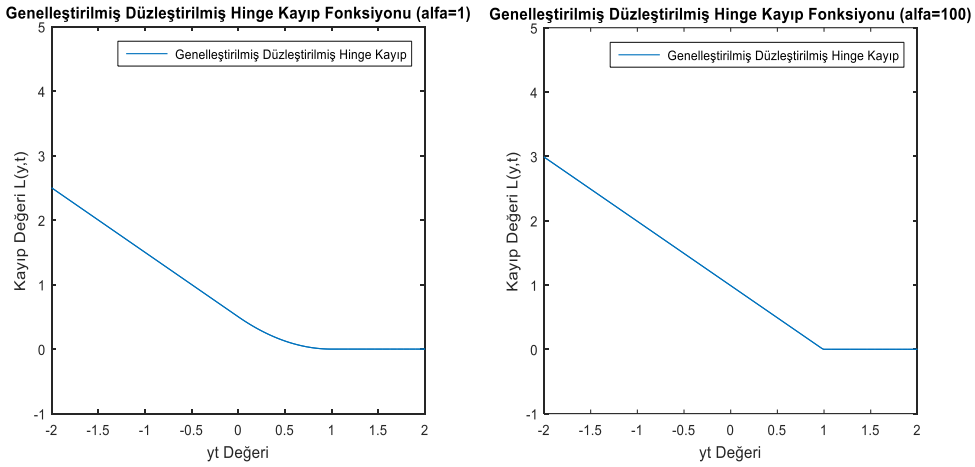
Düzleştirilmiş Hinge kaybı, hinge kaybının $yt = 1$'deki türevlenemez yapısının optimizasyon yöntemine olan etkisini arındırmak için önerilmiştir[31]:

$$L_{D\_Hinge}(y,t) = \begin{cases} 0.5 - yt, & yt \leq 0 \\ 0.5(1-yt)^2, & 0 < yt < 1 \\ 0, & yt \geq 1 \end{cases} \quad (2.13)$$

Düzleştirilmiş Hinge kaybı sadece türevlenebilmeyi değil aynı zamanda kanonik yapıyı sağlayarak gradyen tabanlı çözümün de mümkün olmasına yardımcı olmaktadır. Düzleştirilmiş hinge kaybı bir parametreye bağlanarak genelleştirilmiş düzleştirilmiş hinge kaybı ismiyle Eşitlik (2.14)'deki gibi önerilmiştir [31]:

$$L_{Gen\_D\_Hinge}(\alpha,y,t) = \begin{cases} {}^{\alpha}/_{(\alpha+1)} - yt, & yt \leq 0 \\ {}^{1}/_{(\alpha+1)}\,(yt)^{\alpha+1} - yt + {}^{\alpha}/_{(\alpha+1)}, & 0 < yt < 1 \\ 0, & yt \geq 1 \end{cases} \quad (2.14)$$

Şekil 2.5'ten de görülebileceği gibi α parametresi kayıp değerin büyüklüğüne göre cezayı belirlemektedir. α'nın büyük bir sayı olması hinge kaybına benzer bir kaybın oluşmasına sebep olmaktadır.



Şekil 2.5. Hinge Kayıp Fonksiyonları 2

2.1.4. Lojistik, Kesilmiş Lojistik Kayıp

Lojistik kayıp, lojistik regresyonun kayıp fonksiyonu olmakla birlikte en önemli özelliği şekil olarak hinge kaybına yakın olmasıdır. Ancak hinge kaybının aksine sonsuz sayıda türevlenebilmektedir. Lojistik kayıp Eşitlik (2.15)'teki gibi verilebilir:

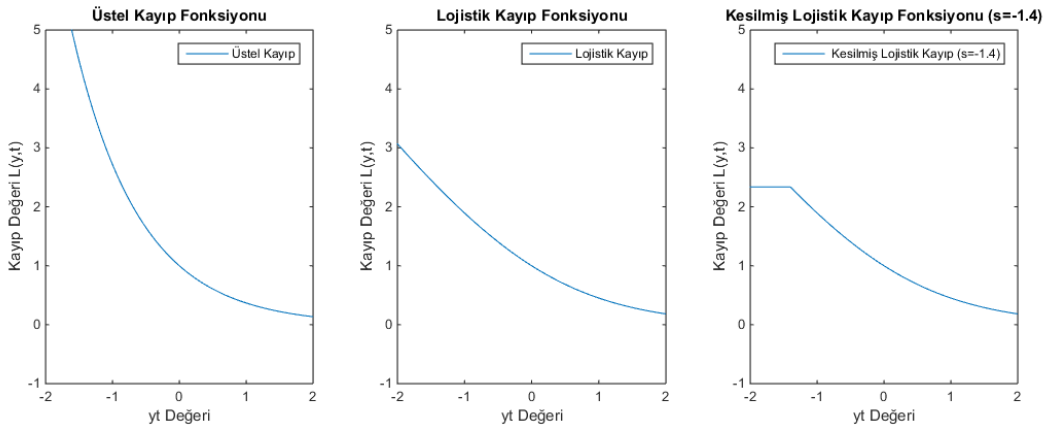
$$L_{loj}(y,t) = log_2(1 + exp(-yt)) \quad y = \pm 1, t \epsilon R \tag{2.15}$$

Sınıfa atama tahminlerinde olasılık vermesi ve bu olasılıklar üzerinden yeni sınıflandırma yapmak isteyen uygulamacıların çok olması, lojistik regresyon yöntemini çekici hale getirmektedir. Cezalandırılmış lojistik regresyon (penalized logistic regression) yönteminde de mantık ceza ve kayıp olarak iki aşamanın optimizasyonudur. Kullanılacak kayıp fonksiyon değerinin yani cezanın sınırsız olması üzerine Park [5] kesilmiş lojistik kayıp kullanılmasını önermiştir:

$y = \pm 1, t \epsilon R, s < 0$ değerleri için $L_{KL1}(y,t) = log_2(1 + exp(-yt))$ ve $L_{KLS}(y,t,s) = log_2(s + exp(-yt))$ olan iki fonksiyonun farkı kesilmiş lojistik fonksiyonu olarak tanımlanmıştır ve Eşitlik (2.16)'daki gibi verilmiştir:

$$L_{K_{Loj.}}(y,t,s) = L_{KL1}(y,t) - L_{KLS}(y,t,s) \quad y = \pm 1, t \epsilon R, s < 0 \tag{2.16}$$

Cezalandırılmış lojistik regresyonda verilen bu önerinin klasik yönteme göre daha sağlam olduğu gösterilmekle birlikte, Fisher tutarlılığı sağladığı, aykırı değerlere daha duyarsız bir sınıflandırma yöntemi olduğu ve sonuçta elde edilen olasılık değerlerinin de aykırı değerlerden etkilenmediği gözlenmiştir [5]. Lojistik ve kesilmiş lojistik fonksiyonlarının farkları Şekil 2.6'dan takip edilebilir.



Şekil 2.6. Üstel ve Lojistik Kayıp Fonksiyonları

2.1.5. Üstel Kayıp

Adaboost yönteminin kayıp fonksiyonu üstel kayıptır ve Eşitlik (2.17)'deki gibi gösterilir. Üstel kayıp fonksiyonun boosting yöntemlerde kullanılma sebebi yanlış sınıflandırma oldukça cezanın hızlı bir biçimde artıyor olması olarak açıklanmıştır. Üstel kaybın kademeli olarak türevlenebilmesi ve konveks olması Newton yöntemi ile çözümü kolaylaştırmaktadır. Ancak yöntemin aykırı değerlere karşı sınırsız ceza veriyor olması, aşırı öğrenme durumunu ortaya çıkartabilecektir [23].

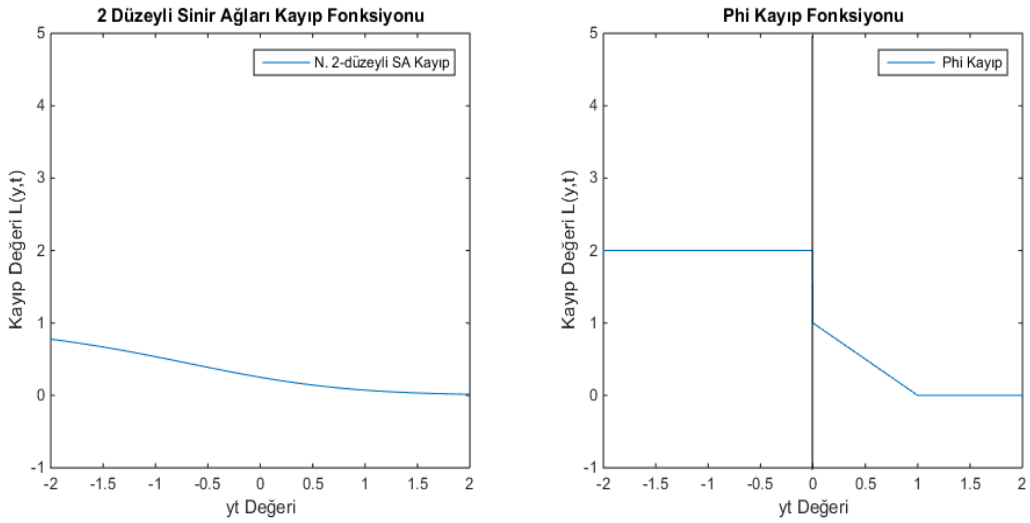
$$L_{\text{üstel}}(y,t) = exp(-yt) \quad y = \pm 1, t \epsilon R \tag{2.17}$$

2.1.6. 2-Düzeyli Yapay Sinir Ağları Kaybı

Ceza ve kayıp terimleri için kayıp fonksiyona sinir ağları üzerinden bir fonksiyon yazılarak çözümleme yapılabileceği de bilinmektedir. Cezalandırılmış lojistik regresyona oldukça benzer şekilde kullanılan bu kayıp, diğer kayıp fonksiyonları ile de sıkça karşılaştırılmaktadır. Li ve Yang [33] gizli tabaka olmadan iki tabakadaki sınıflandırma için yapay sinir ağlarını aşağıdaki gibi tanımlamıştır:

$$L_{2\,YSA}(y,t) = \left(1 - \frac{1}{1 + exp(-yt)}\right)^2 \quad y = \pm 1, t \epsilon R \tag{2.18}$$

İki düzeyli yapay sinir ağları kayıp fonksiyonu Şekil 2.7'deki gibidir.



Şekil 2.7. İki Düzeyli Sinir Ağları ve Psi Kayıp Fonksiyonları

2.1.7. Psi(Ψ) Kaybı

Sınıflandırmada deneysel genelleştirme hatasını en aza indirgemeye çalışmanın aşırı öğrenmeye sebep olacağından yola çıkan Shen vd. [34], yeni kayıp fonksiyonu eşitlikteki gibi önermişlerdir:

$$L_{\Psi}(y,t) = \begin{cases} 2, & yt < 0 \\ 1 - yt & 0 \leq yt \leq 1 \\ 0, & yt > 1 \end{cases} \qquad y = \pm 1, t \epsilon R \tag{2.19}$$

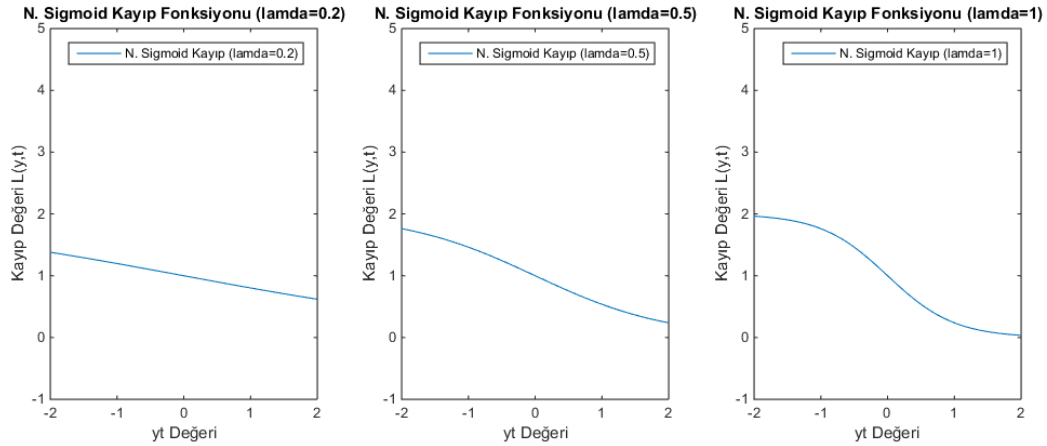
Yöntem aslında hinge fonksiyonuna benzemektedir. Farkı ise, doğru sınıflandırılmış ancak sınıflandırıcıya yakın olan kayıplara doğrusal artan bir ceza verirken, yanlış sınıflandırmaya sabit ve 2 değerinde bir ceza vermektedir. Cezanın iki ile sabit olması aykırı değerlere duyarsız olmasına ve yeniden örnekleme durumunda kararlı davranmasına sebep olacaktır.

2.1.8. Sigmoid Kayıp

Mason vd. [35], ayrım tabanlı kayıp fonksiyonların aşırı öğrenmeden kurtulmanın bir yolu olduğunu açıklamışlardır. Adaboost yönteminde üstel kayıp ile aşırı öğrenme durumu meydana geldiğinden sigmoid kayıp fonksiyonu Eşitlik (2.20)'deki gibi önerilmiştir:

$$L_{Sigmoid}(y,t) = 1 - tanh(\lambda yt) \;\; y = \pm 1, t \epsilon R \tag{2.20}$$

Fonksiyondaki λ parametresi, ayrıma göre dikliğin miktarını belirlemektedir. Şekil 2.8'den de görüleceği üzere λ parametresi arttıkça, kayıp fonksiyonu doğrusal bir halden eğrisel bir hale geçmektedir.



Şekil 2.8. Sigmoid Kayıp Fonksiyonları

Düzeltme yapan λ parametresi, çok küçük kayıp değerlerine artan bir hızda; daha sonra doğrusala yakın olacak şekilde ceza verir. Büyük kayıp değerine sahip olanlara ise azalan ve belli bir noktadan sonra sabitleşen ceza vermektedir.

2.1.9. Huber ve Uyarlanmış Huber Kaybı

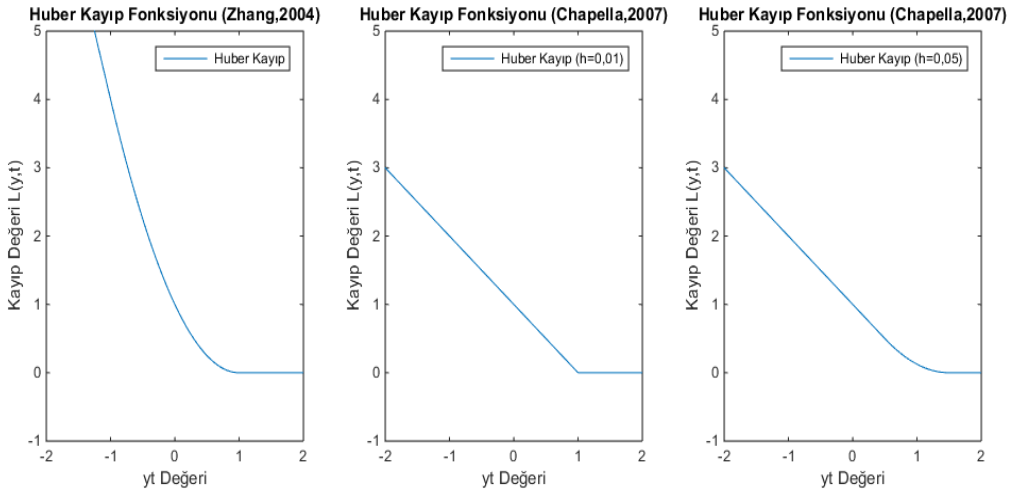
Zhang [32] tarafından önerilen Huber kaybı, uyarlanmış karesel kaybın (karesel hinge kaybın) yt değeri -1 ve daha düşük değer aldığı durumda cezayı monoton hale getirmektedir. Kaybın aldığı ceza değerleri Eşitlik (2.21)'den takip edilebilir:

$$L_{\boldsymbol{Huber}}(y,t) = \begin{cases} max\{0,(1-yt)\}^2, & yt \geq -1 \\ -4yt, & \text{ö.d.} \end{cases} \qquad y = \pm 1, t \epsilon R \qquad (2.21)$$

Ancak Chapelle [36], DVM'nin primal çözümünde denemeler yaparken uyarlanmış modifiye kaybını Eşitlik (2.22)'deki gibi önermiştir:

$$L_{\boldsymbol{M\_Huber}}(y,t) = \begin{cases} 0, & yt > 1+h \\ \dfrac{(1+h-yt)^2}{4h}, & |1-yt| \leq h \\ 1-yt, & yt < 1-h \end{cases} \qquad y = \pm 1, t \epsilon R \qquad (2.22)$$

Zhang [32] çalışmasında, sınıflandırma için kullanılan Huber kaybının belli bir değerden sonra monoton hale getirilmesi doğrusal bir fonksiyon ile sağlanırken, Chapelle [36]'de önerilen uyarlanmış Huber kaybı h parametresine bağlı olacak şekilde değer almaktadır. h değerinin 0,01 ile 0,5 arasında olması önerilmektedir ve sıfıra yakın değer alması hinge kaybına benzer fonksiyonu ortaya çıkarmaktadır. Kayıpların değerleri Şekil 2.9'dan takip edilebilir.



Şekil 2.9. Huber Kayıp Fonksiyonları

2.1.10. Savage Kayıp ve Tanjant Kayıp

Aykırı değerlere karşı sağlam olan yapıya sahip kayıp fonksiyonlardır. Koşullu risk fonksiyonunun sağlaması gereken tutarlılık tartışılmış ve Masnadi-Shirazi ve Vasconcelos [37,38] tarafından Eşitlik (2.23)'te verilen Savage kayıp önerilmiştir:

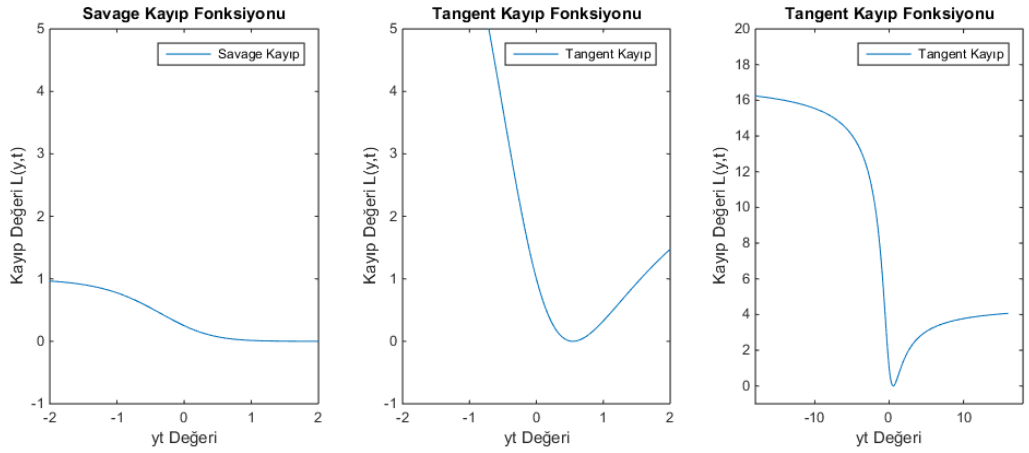
$$L_{Savage}(y,t) = \frac{1}{(1+exp(2yt))^2} \qquad y = \pm 1, t \epsilon R \qquad (2.23)$$

Savage kayıp fonksiyonu, sınıflandırmadaki yt değerinin küçük olduğu durumda karesel olacak şekilde bir kayıp değeri verirken yt değerinin artmasıyla birlikte sabitleşmeye başlamaktadır. Bu özellik ile aykırı değerlere karşı sağlam bir kayıp fonksiyon önerilmiştir. Kayıp fonksiyonu konveks olmamasına rağmen yöntemde minimizasyon işlemi karesel risk üzerinden yapılmaktadır.

Tanjant kayıp, savage kayıp özelliğini sağlamakla birlikte doğru sınıflandırmaya da kayıp değeri tanımlamaktadır. Kayıp fonksiyonu Eşitlik (2.24)'deki gibi önerilmiştir:

$$L_{Tangent}(y,t) = (2\,arctan(yt) - 1)^2 \qquad y = \pm 1, t \epsilon R \qquad (2.24)$$

Üstelik doğru sınıflandırmanın ceza değeri aynı durumda olan yanlış sınıflandırmadan daha küçük ve yine sınırlı bir şekilde elde edilmiştir. Kayıp değerleri Şekil 2.10'dan takip edilebilir.



Şekil 2.10. Savage ve Tanjant Kayıp Fonksiyonları

3. BOOSTING YÖNTEMLERDE SAĞLAM KAYIP FONKSİYONU

3.1. Boosting

Boosting yöntemlerde en temel mantık, zayıf sınıflandırıcıları bir araya getirerek güçlü bir sınıflandırıcı ortaya çıkartmaya çalışmaktır. Boosting yöntemler, birçok zayıf sınıflandırıcının birlikte vereceği çıktı sonucundaki kararın, bir sınıflandırıcının kararından daha mantıklı olacağından hareketle önerilmiştir. 1980'lerin sonunda sınıflandırıcıların birleştirilmesiyle daha güçlü bir sınıflandırma elde edilebilir mi sorusu ile tartışmalar başlamıştır [39]. Günümüze kadar çeşitli algoritma yöntemleri önerilerek boosting metodolojisi geliştirilmiştir. Başlıca algoritmalar, Adaboost, LojitBoost, Realboost, Brownboost olarak isimlendirilebilse de temel mantık aynıdır. Özellikle araştırmacıların kullanmış olduğu boosting algoritmaları ile ilgili birçok çalışma bulunmaktadır. Son yıllarda yapılan çalışmalar incelendiğinde Wang ve Yang [40], meme kanseri görüntülerini tanılamada AdaBoost algoritması ile bir uygulama yaparak gerçek veri kümesinde boosting yöntemlerin başarılarına değinmiştir. Dubossarsky vd. [41] bir uygulama olarak dalga tabanlı gradyan boosting algoritmasını önermiştir. Yöntem R programına eklenmiştir ve başarılı bir şekilde regresyon uyumu ve sınıflandırıcı üretebilmektedir. Sen vd. [42] boosting ve aşırı örnekleme ile etkili bir sınıflandırma yöntemi önermişlerdir. Pan vd. [43] çalışmasında, ceza duyarlı grafik sınıflandırma algoritması, CogBoost, yanlış sınıflandırma cezalarını minimize edecek şekilde çalışmaktadır. Yöntem ayrıca büyük ölçekli verilerde hızlı öğrenme mantığını geliştirmiştir. Yöntemi yanlış sınıflandırmaların minimize edilmesi için iteratif olarak en önemli ayrışma grafiğini seçer ve doğrusal programlama problemini her iterasyon için optimal kayıp fonksiyon tabanlı Bayes karar kuralı ile çözümler. Mund vd. [44], dengesiz sınıf dağılımları ve aykırılıkların bulunduğu durumlar için bir sınıflandırma yöntemi önermişlerdir. Zeng vd. [45], esnetilebilir konveks kabuk ile maksimum ayrım sınıflandırması için bir öneri vermiştir. Nie vd. [46] sınıflandırma problemlerinde AdaBoost yönteminin olasılıksal sınıflandırmaları üzerine öneriler getirmişlerdir. Özellikle sınıflara atama için gerekli olan olasılık değerlerinin aykırı değerlere karşı duyarsız olması için öneri geliştirmişlerdir. De Menezes vd. [47] en çok olabilirlik modeli ile lojistik regresyon ve ikili boosting algoritması ile lojistik regresyon yöntemlerini karşılaştırmak amacıyla çalışma yapmışlar ve boosting tabanlı olan sınıflandırmada daha başarılı sonuçlar elde etmişlerdir. Kim vd. [48], sınıflar arasında dengeli oranların bulunmadığı veri kümeleri için geometrik ortalama tabanlı bir boosting algoritması önermişlerdir. Zhai vd.

[49] dengesiz veri kümeleri için dengeli birden çok veri kümesi yaratarak çözüm bulmaya çalışan bir yöntem önermişlerdir. Önerilen algoritma, hızlı, ölçeklenebilir ve karşılaştırıldığı yöntemlerden daha başarılı olarak tanıtılmıştır. Lin vd. [50], RBoosting isimli, tekrar ölçeklenebilir ve boosting algoritmasının öğrenme sürecini geliştiren yöntemi önermişlerdir. Yöntemin genelleştirme bakımından karşılaştırılan diğer yöntemlerden iyi sonuçlar verdiği görülmüştür. Wang ve Pineau [51], dengesiz sınıfların bulunduğu durumda anlık sınıflandırma yapabilecek olan bagging ve boosting yöntemler üzerinde çalışmışlar ve birçok algoritma vermişler, uygulama için birçok veri kullanmışlardır. Nikolaou vd. [52], araştırdıkları boosting yöntemler üzerindeki uygulamanın sonucunda AdaBoost yönteminin karar eşik değeri ve kalibre edilmiş olasılık değerleri ile kullanılabilir olduğunu belirtmişlerdir. Ghosal vd. [53], DVM sınıflandırması ile seyrek cezalandırmalı ileri seçim yöntemini birleştirerek çok boyutlu verilerde değişken seçimi ve ikili sınıflandırma yöntemi önermişlerdir. Yöntem aslında ileri seçim yöntemi ile cezalandırılmış DVM ve onun çeşitlerinin bir araya getirilmesidir. Martinez ve Gray [54], boosting algoritmalarında aykırı değerlerin etkisini arındırmak için yöntemin, ayrım soyma (margin peeling) yöntemi ile birleştirilmesini önermişlerdir. Conroy vd. [55], kayıp verilerin olduğu durumda sağlam sınıflandırma için katıştırma modeller kullanılan bir yöntem önermişlerdir. Birçok sınıflandırma yöntemi ve kayıp veri yükleme yöntemleri ile elde edilen önerileri bir araya getirerek gerçek veri kümesi üzerinde sonuçları karşılaştırmışlardır. Xiong vd. [56], görsel tanılama için sağlam değişken havuzu yöntemi ile kümeleme yöntemi önermişlerdir. Önerilen yöntemler gerçek veri uygulamasına sahip güncel çalışmalarda kullanılmaktadır [57-60].

Bundan sonraki bölümlerde boosting yöntemlerde kayıp fonksiyonun önemi, sağlam kayıp fonksiyon özellikleri, Masnadi-Shirazi vd. [38] tarafından önerilen sağlam kayıp yöntemi, tanjant kayıp fonksiyonu, bu çalışmada önerilen Gudermannian kayıp fonksiyonu ve sağlam özellikleri, tanjant kayıp için gradyan azalış (gradient descent) üzerinden uyarlanan LojitBoost mantığıyla GudermannianBoost yöntemi aktarılmıştır.

3.2. Boosting Yöntemlerde Sağlam Kayıp Fonksiyonlar ve Özellikleri

Boosting yöntemlerde kayıp fonksiyon önemli bir yer tutmaktadır. Özellikle kayıp fonksiyon üzerinden riski minimize eden boosting yöntemlerde seçilen kayıp fonksiyonun önemi büyüktür. Sınıflandırmada kullanılan kayıp fonksiyonlar ile ilgili bilgiler detaylı olarak Bölüm 2'de anlatılmıştır. Klasik boosting yöntemlerde en temel mantık, basit yanlış sınıflandırmaya

verilen cezaların sınırsız olmasıdır. Bu mantık çeşitli araştırmacılar tarafından tartışılmıştır. Aynı şekilde doğru sınıflandırılmış olan gözlemler için sadece sınıflandırıcıya yakın olanlara az miktar ceza uygulanması (margin-enforcing) öğrenme veri kümesindeki aykırı değerlerin ve bozulmaların sınıflandırıcı üzerinde etkili olmasına sebep olmaktadır [30]. Bu problemin giderilmesi için Masnadi-Shirazi vd. [38] tarafından TanjantBoost algoritması ile sağlam olan bir yöntem önerilmiştir. Sağlam kayıp fonksiyonlar ile sınıflandırıcıların elde edilmesi, veri kümesindeki aykırı değerler ve bozulumlardan etkilenmeyen kestirimler elde etmek için büyük önem taşımaktadır. Hem yanlış hem de doğru sınıflandırma için ceza verilmesi aykırı değerlerin etkisini ortadan kaldırmayı amaçlamaktadır. Masnadi-Shirazi vd. [38] bilgisayar görüntülemesinde sağlam sınıflandırıcılar üzerine yapmış oldukları önerilerde sağlam kayıp için çeşitli özellikler sıralamışlardır.

3.3. Sağlam Kayıp Fonksiyon Özellikleri

Sınıflandırma yöntemlerinde yanlış sınıflandırma için sınırsız cezaların bulunması, doğru sınıflandırmalara ceza verilmemesi, sınıflandırıcının aykırı değerlerden etkilenmesine ve örneklemlere göre kararsız sınıflandırıcılara sebep olmaktadır. Birçok kayıp fonksiyon, sınıflandırıcıya yakın ve doğru sınıflandırılmış olan gözlemler için çok az ceza uygulamaktadır. Ayrıca sınıflandırıcıdan uzak ve yanlış sınıflandırılmış gözlemler için ceza sınırlandırılmasının yapılmaması, yine klasik kayıplarda sağlam olmayan sınıflandırıcıların ortaya çıkmasına sebep olmaktadır.

Bu noktadan hareketle sağlam kayıp fonksiyonlarının hangi özellikleri taşıması gerektiği Masnadi-Shirazi vd. [38] tarafından aşağıdaki maddeler ile verilmiştir ($L_\phi(y,t) = \phi(yt)$):

- Sınıflandırıcıya uzak olan gözlemler için belli noktadan sonra sabitleşmiş kayıp değeri verilmelidir:

$$\phi'(\infty) = \phi'(-\infty) = 0 \tag{3.1}$$

- Yanlış sınıflandırma için sınırlandırılmış ceza uygulanmalıdır:

$$\phi(-\infty) = k_1 < \infty \tag{3.2}$$

- Doğru sınıflandırma için yanlış sınıflandırmadan daha az olacak şekilde sınırlandırılmış ceza uygulanmalıdır:

$$0 < \phi(\infty) = k_2 < k_1 \tag{3.3}$$

- Ayrım sınır cezalandırması olmalıdır (margin enforcing):

$$\phi(0) > 0 \tag{3.4}$$

Bu koşulların sağlanması için sınıflandırıcı ve sınıflandırıcıya ait olan risk fonksiyonu, $\gamma(v) = f^{-1}(-v)xJ'[f^{-1}(-v)]$ ve $-J[v] = C_{\phi}^{*}[v]$ eşitlikleri altında aşağıdaki koşulları sağlamalıdır:

$$[f^{-1}]'(\infty) = [f^{-1}]'(-\infty) = 0 \tag{3.5}$$

$$f^{-1}(\infty) > \frac{1}{2} = f^{-1}(0) \tag{3.6}$$

$$C_{\phi}^{*}(0{,}5) > 0 \tag{3.7}$$

$$C_{\phi}^{*}[f^{-1}(\infty)] + \gamma(\infty) > 0 \tag{3.8}$$

$$C_{\phi}^{*}[f^{-1}(\infty)] + \gamma(-\infty) < \infty \tag{3.9}$$

Boosting algoritmalar için önerilen sağlam kayıp fonksiyon mantığı, temel olarak sınıflandırıcıya olan uzaklığa göre verilecek cezaların miktarlarını sınırlandırarak sınıflandırıcının daha tutarlı çözümler vermesini amaçlamaktadır. Bu noktadan hareket edildiğinde Masnadi-Shirazi vd. [38] tarafından tanjant kayıp fonksiyonu ve bu kayıp fonksiyona bağlı boosting algoritması önerilmiştir.

3.4. Tanjant Kayıp Fonksiyonu

Koşullu risk minimizasyonu sınıfsal olasılıkların elde edilmesiyle ilişkilidir. Aslında kayıp fonksiyonu minimize ederken buradaki amaç, aşağıdaki beklenen kazancı maksimum yapan olasılık kestiricisi $\hat{\eta}$ değerini bulmaktır:

$$I(\eta, \hat{\eta}) = \eta I_1(\hat{\eta}) + (1 - \eta) I_{-1}(\hat{\eta}) \tag{3.10}$$

Eşitlikteki $I_1(\hat{\eta})$, $y = 1$ olayında $\hat{\eta}$ tahmini için elde edilen kazanım, $I_{-1}(\hat{\eta})$, $y = -1$ olayında $\hat{\eta}$ tahmini için elde edilen kazanımdır. $I_1(\hat{\eta})$ ve $I_{-1}(\hat{\eta})$ fonksiyonlarının beklenen kazancın maksimum olması $\hat{\eta} = \eta$ eşitliği ile sağlanmaktadır. Bu durumda,

$$I(\eta, \hat{\eta}) \leq I(\eta, \eta) = J(\eta) \tag{3.11}$$

denklemi için eşitlik ancak $\hat{\eta} = \eta$ durumunda sağlanmaktadır. Eşitliğin sağlanması için maksimum kazanım fonksiyonu $J(\eta)$'in tam konveks olması ve aşağıdaki eşitliklerin sağlanması gerekmektedir [61]:

$$I_1(\eta) = J(\eta) + (1-\eta)J'(\eta) \tag{3.12}$$

$$I_{-1}(\eta) = J(\eta) - \eta J'(\eta) \tag{3.13}$$

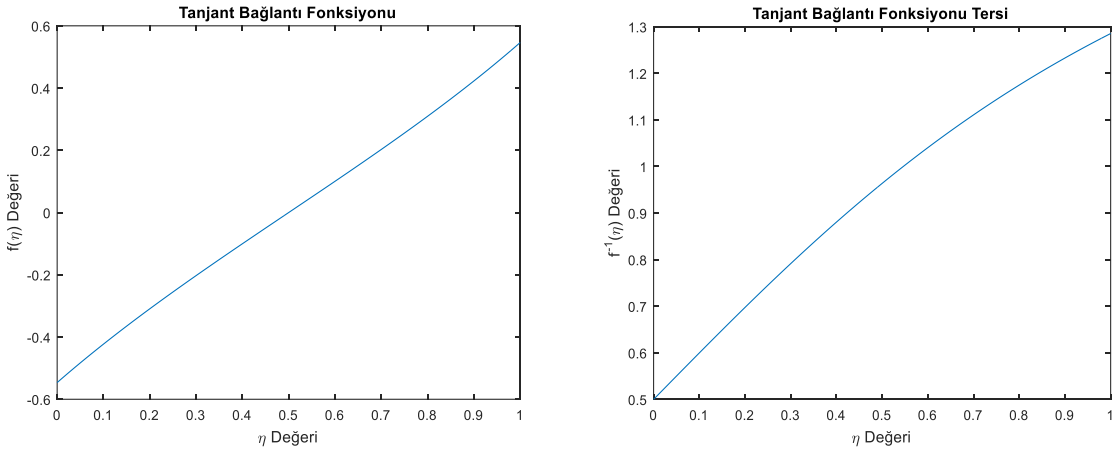
Riskin minimize edilmesi ve beklenen kazanç ve olasılık ile ilgili çıkarımlar üzerinden yola çıkan Masnadi-Shirazi vd. [38] tanjant kaybını önermişlerdir. $J(\eta) = J(1-\eta)$ eşitliğinin ve f tersinirken $f^{-1}(-v) = 1 - f^{-1}(v)$ simetrisinin sağlanması durumunda yukarıda eşitlikleri verilen $I_1(.)$ ve $I_{-1}(.)$ fonksiyonları $I_1(\eta) = -\phi(f(\eta))$ ve $I_{-1}(\eta) = -\phi\big(-f(\eta)\big)$ Eşitlik (3.14)'teki kayıp fonksiyon için sağlanır:

$$\phi(v) = -J[f^{-1}(v)] - \big(1 - f^{-1}(v)\big)J'[f^{-1}(v)] \tag{3.14}$$

Bu durumda beklenen kazanımın maksimum edilmesi ile riskin minimize edilmesi arasındaki ilişki $C_\phi^*(\eta) = -J(\eta)$ eşitliği üzerinden ifade edilebilmektedir. Yani, $J(\eta) = -C_\phi^*(\eta)$ eşitliği düşünüldüğünde $C_\phi^*(\eta)$ ve f_ϕ^* fonksiyonları tanımlanarak kayıp fonksiyon Eşitlik (3.14) kullanılarak elde edilebilir. Burada koşul olarak $f^{-1}(-v) = 1 - f^{-1}(v)$ olmalı ve $C_\phi^*(\eta)$ tam konkav olmalıdır.

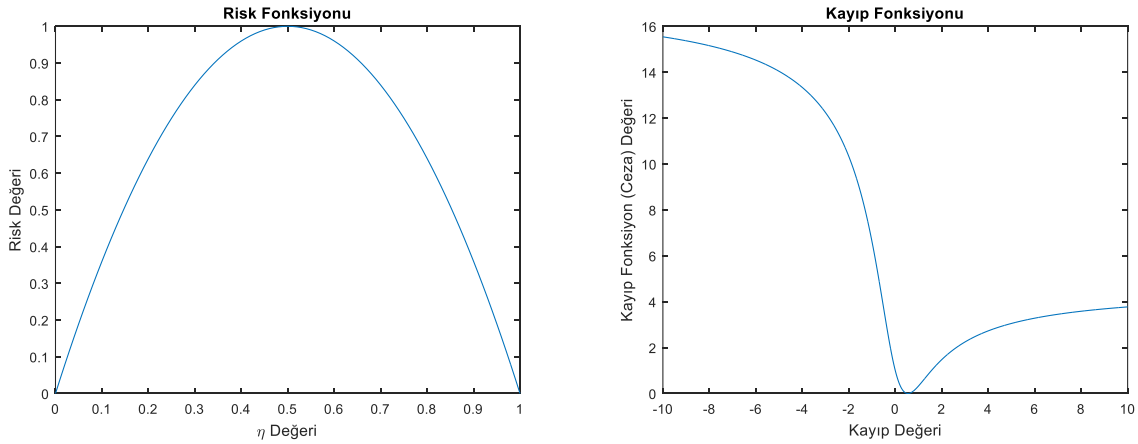
Bu özelliklerden hareket ederek $C_\phi^*(\eta) = 4\eta(1-\eta)$ ve Şekil 3.1'de verilen bağlantı fonksiyonu $f(\eta) = \tan(\eta - 0{,}5)$ alındığında kayıp fonksiyonu formülü üzerinden Eşitlik (3.15)'teki gibi bir öneri getirilmiştir:

$$\phi(v) = [2\,arctan(\eta) - 1]^2 \tag{3.15}$$



Şekil 3.1. Tanjant Fonksiyonu ve Tersi

Önerilen fonksiyonun grafiği Şekil 3.2'deki gibidir.



Şekil 3.2. Tanjant Risk Fonksiyonu ve Kayıp Fonksiyonu

Karesel bir risk olduğu ve $C_{\phi}^{*}(\eta)$'nin konveks olduğu bilinmektedir. Ayrıca f^{-1}'in var olduğu ($f^{-1}(v) = \arctan(v) + 0.5$) ve simetrik olma özelliğini sağladığı görülmektedir. Bu noktadan hareketle $f^{-1}(v)$'in S-tipi bir fonksiyon olması ve sınırlı olması gerektiği tartışılmıştır. Bu özelliğe sahip olan $f^{-1}(v)$ fonksiyonunun tersi alınarak $f(\eta)$ elde edilebiliyorsa ve riskin minimizasyonu için gerekli olan şartları sağlıyorsa kayıp fonksiyon önerilebilir.

3.5. Öneri: Gudermannian Kayıp Fonksiyonu

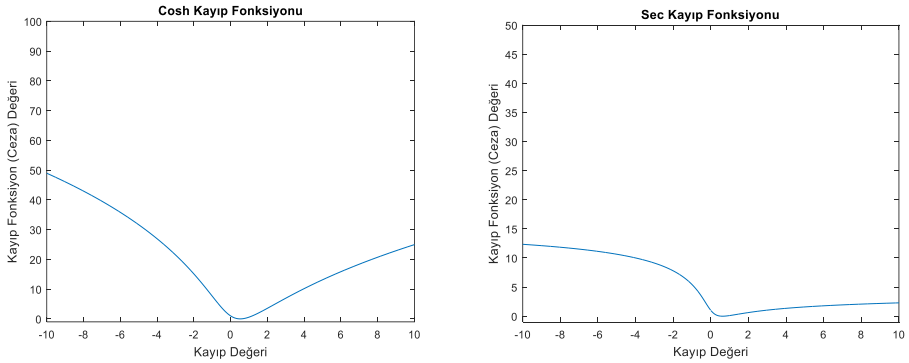
Bir önceki alt bölümde belli özellikler üzerinden kayıp fonksiyon elde edilebildiği ve bu kayıp fonksiyonlardan biri olan tanjant kayıp fonksiyonu açıklanmıştır. Ters fonksiyonun S-tipi fonksiyon olacağı düşünüldüğünde olasılık dağılım fonksiyonları kullanılabilir. Ancak yanlış sınıflandırma ve doğru sınıflandırmada farklı cezalandırma değerleri için trigonometrik fonksiyonların kullanılması önerilebilir.

Masnadi-Shirazi [26], ters fonksiyonun $cosh$ ve sec olduğu ve risk fonksiyonun en küçük kareler alındığı durumda hem bayes tutarlı hem de sağlam olan kayıp fonksiyonların elde edildiğini göstermiştir. Bu yöntemlerle elde edilen iki kayıp fonksiyon Eşitlik (3.16)'da ve Eşitlik (3.17)'de verilmiştir:

$$\phi_{cosh}(v) = 1 + \frac{4}{a} arcsinh\left(\frac{-v}{a}\right) + \frac{4}{a^2}\left(arcsinh\left(\frac{-v}{a}\right)\right)^2 \tag{3.16}$$

$$\phi_{sec}(v) = 1 + \frac{4}{a} arcsin\left(\frac{a - \sqrt{a^2 + 4v^2}}{2v}\right) + \frac{4}{a^2} arcsin\left(\frac{a - \sqrt{a^2 + 4v^2}}{2v}\right)^2 \tag{3.17}$$

İki kayıp fonksiyonun ceza değerleri Şekil 3.3'ten takip edilebilir.



Şekil 3.3. Cosh ve Sec Kayıp Fonksiyonları

Yukarıdaki kayıp fonksiyonların en büyük özelliği yanlış sınıflandırmaya olduğu gibi doğru sınıflandırmaya da ceza vermesi ve ceza miktarının yanlış sınıflandırma cezasından daha az olmasıdır. Aynı şekilde tanjant kayıp fonksiyonu da benzer özelliği taşımaktadır. Tanjant kayıp fonksiyonunun Şekil 3.3'te verilen fonksiyonlardan farkı daha hızlı şekilde ceza vermesi ve daha hızlı sabitleşmesidir. Sınıflandırıcıya yakın ve yanlış sınıflandırılmış gözlemlerin sınıflandırıcı üzerindeki etkisini azaltmak için Tanjant kayıp fonksiyonundan daha hızlı ceza veren ve daha hızlı sabitleşen bir kayıp fonksiyona ihtiyaç duyulmaktadır. Trigonometrik fonksiyonlar gibi ters fonksiyon özelliğini taşıyan fonksiyonlardan biri de Eşitlik (3.18)'de verilen Gudermannian fonksiyonudur:

$$arcsin(tanh(n)) \tag{3.18}$$

Gudermannian fonksiyonu trigonometrik fonksiyonlar arasında geçiş sağlayan bir fonksiyondur ve genel yazım şekli Eşitlik (3.19)'da verilmiştir:

$$gd(n) = \int_0^n \frac{1}{cosh(t)}\,dt\,, -\infty < n < \infty \tag{3.19}$$

Eşitlik (3.19) haricinde Gudermannian fonksiyonu birden çok şekilde yazılabilir ve bu eşitlikler Eşitlik (3.20)'de verilmiştir:

$$gd(n) = arcsin(tanh(n)) = arctan(sinh(n)) = arccsc(coth(n))$$
$$= sgn(n)\,arccos(sech(n)) = arcsec(cosh(n)) = 2\,arctan\left(tanh\left(\frac{1}{2}n\right)\right)$$
$$= 2\,arctan(exp(n)) - \frac{1}{2}\pi \tag{3.20}$$

Tersi alınabilen bir fonksiyon olması özelliği de kayıp fonksiyon için kullanımını uygun kılmaktadır. Gudermannian fonksiyonun tersi Eşitlik (3.21)'de ve farklı yazılımları Eşitlik (3.22)'de verilmiştir:

$$gd^{-1}(n) = \int_0^n \frac{1}{cos(t)}\,dt\,, -\frac{\pi}{2} < n < \frac{\pi}{2} \tag{3.21}$$

$$gd^{-1}(n) = ln\left|\frac{1+sin(n)}{cos(n)}\right| = \frac{1}{2}ln\left|\frac{1+sin(n)}{1-sin(n)}\right| = ln|tan(n)+sec(n)|$$

$$= ln\left|tan\left(\frac{1}{4}\pi + \frac{1}{2}n\right)\right| = arctanh(sin(n)) = arcsinh(tan(n)) = arccoth(csc(n))$$

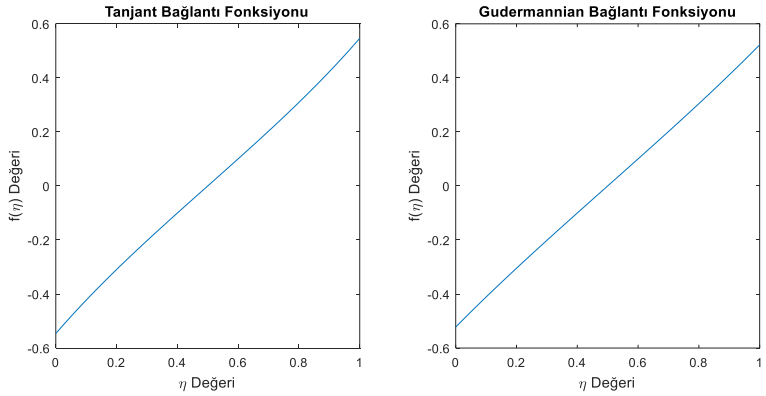
$$= arccsch(cot(n)) = sgn(n)\,arccosh(sec(n))$$

$$= sgn(n)\,arcsech(cos(n)) \tag{3.22}$$

Bu durum için Gudermannian fonksiyonunun tersi bağlantı fonksiyonu olarak ele alınsın:

$$f(\eta) = arcsinh(tan(\eta - 0{,}5)) \tag{3.23}$$

Tanjant bağlantı fonksiyonu ve Gudermannian bağlantı fonksiyonu arasındaki farklılık Şekil 3.4'ten takip edilebilir.

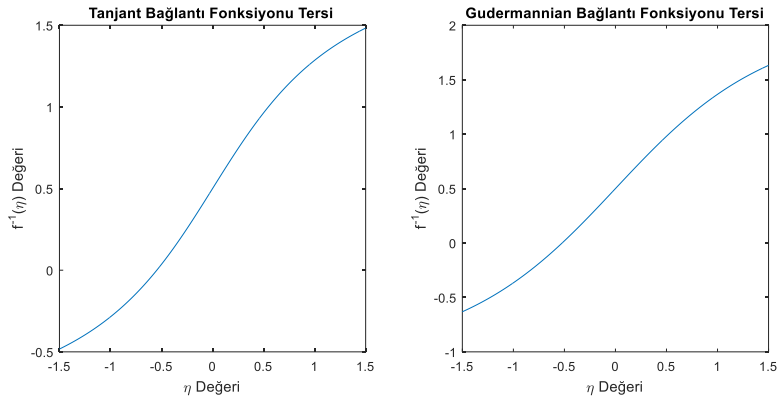


Şekil 3.4. Tanjant ve Gudermannian Fonksiyonları

Gudermannian fonksiyonu $f^{-1}(v)$ olarak alındığında aşağıdaki gibidir:

$$f^{-1}(v) = arcsin(tanh(v)) + 0{,}5 \tag{3.24}$$

Eşitlik (3.24)'teki 0,5 değeri, ayrım sınırlarında cezalandırma uygulanması ve bayes tutarlılığın sağlanabilmesi ($\eta \geq 0{,}5$ durumunda $f(\eta) \geq 0$) için yapılan düzenlemedir. Tanjant ve Gudermannian bağlantı fonksiyonlarının tersi Şekil 3.5'ten takip edilebilir.



Şekil 3.5. Tanjant ve Gudermannian Fonksiyonları Tersi

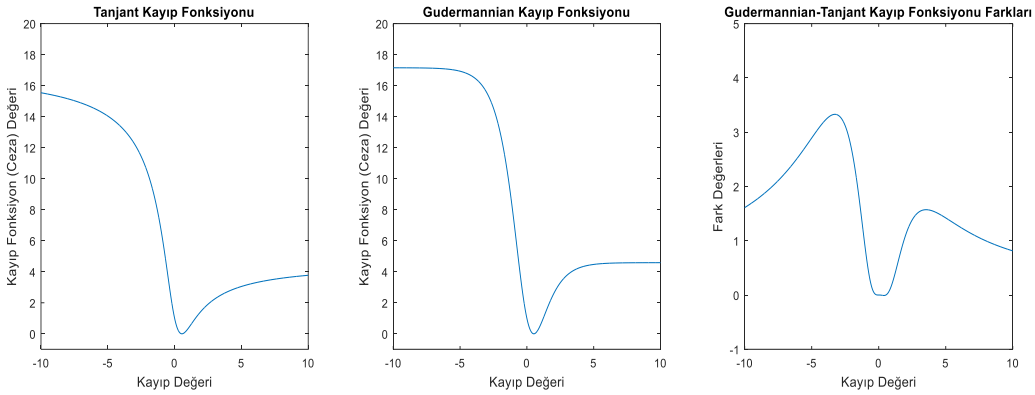
$f^{-1}(v) = \arcsin(\tanh(v)) + 0{,}5$ ve risk $C_\phi^*(\eta) = -J(\eta) = 4\eta(1-\eta)$ şeklinde kabul edildiğinde $\phi(v) = -J[f^{-1}(v)] - \big(1 - f^{-1}(v)\big)J'[f^{-1}(v)]$ eşitliğinden ve $J'(\eta) = -4(1-2\eta)$ eşitliğinden kayıp fonksiyon aşağıdaki gibi elde edilir:

$$\begin{aligned}\phi(v) &= 4[\arcsin(\tanh(v)) + 0{,}5][0{,}5 - \arcsin(\tanh(v))] \\ &\quad + 4[0{,}5 - \arcsin(\tanh(v))][-2\arcsin(\tanh(v))] \\ &= 4[\arcsin(\tanh(v)) + 0{,}5][0{,}5 - \arcsin(\tanh(v))] \\ &\quad - 8[0{,}5\arcsin(\tanh(v)) - \arcsin(\tanh(v))^2] \\ &= [2\arcsin(\tanh(v)) - 1]^2\end{aligned}$$

$$\phi(v) = [2\arcsin(\tanh(v)) - 1]^2 \tag{3.25}$$

Kayıp fonksiyon genel olarak sağlam kayıp fonksiyon özelliklerini taşımasıyla birlikte diğer trigonometrik fonksiyonlardan elde edilen yöntemlere benzemektedir. Diğer trigonometrik fonksiyonlardan farkı, Gudermannian kayıp fonksiyonunda cezalandırma daha hızlı şekilde artar ve sabitleşir.

Şekil 3.6'dan da gözlenebileceği gibi tanjant fonksiyonunun artışına nazaran Gudermannian fonksiyonu hızla ceza arttırmakta; daha sonra tanjant fonksiyonunda olduğu gibi sabitleşmektedir. Cezalar sınırlandırılmaktadır ve doğru sınıflandırma, yanlış sınıflandırmadan daha az şekilde ceza almaktadır. Kaybın değeri, yani sınıflandırıcıya uzaklık arttıkça kayıp fonksiyonların vermiş olduğu ceza aynıdır. Bu kayıp fonksiyonun sınır değerlerine fazla ceza vermesi sayesinde sınıflandırıcıya yakın yanlış sınıflandırılmış gözlemlerin sınıflandırıcının kararlılığını etkilemesi önlenebilir.

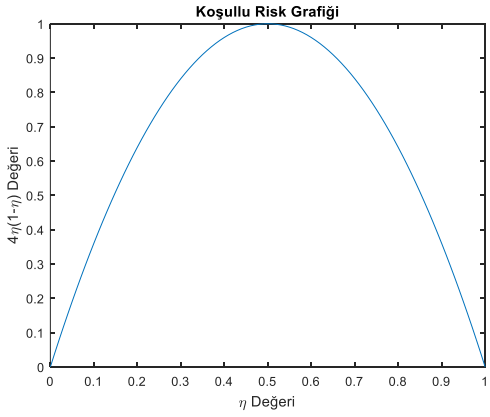


Şekil 3.6. Tanjant ve Gudermannian Kayıp Fonksiyonları ve Farkları

3.6. Gudermannian Kayıp Fonksiyon Özellikleri

Riskin minimize edilmesi için risk fonksiyonu $C_\phi^*(\eta) = -J(\eta) = 4\eta(1-\eta)$ ve koşullu risk $C_\phi(\eta) = \eta\phi(f) + (1-\eta)\phi(-f)$ olmak üzere aşağıdaki Eşitlik (3.26) sağlanmalıdır. Şekil 3.7'de verilen $C_\phi^*(\eta)$ önceden belirlenerek formülle kayıp fonksiyonu elde edildiğinden koşul sağlanacaktır:

$$
\begin{aligned}
C_\phi^*(\eta) \; &= \eta\phi(f^*) + (1-\eta)\phi(-f^*) \\
&= \eta(2\,arcsin(tanh(arcsinh(tan(\eta - 0.5))))) - 1)^2 \\
&+ (1-\eta)(2\,arcsin(tanh(-arcsinh(tan(\eta - 0.5))))) - 1)^2 \\
&= 4\eta(1-\eta) \qquad (3.26)
\end{aligned}
$$



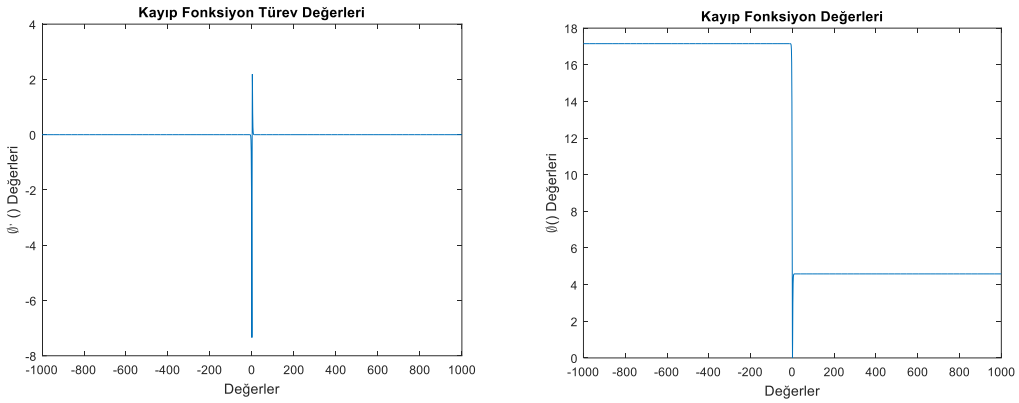
Şekil 3.7. Gudermannian Risk Fonksiyonu

$J(\eta) = J(1-\eta)$ özelliği de $J(\eta) = -4\eta(1-\eta)$ olduğundan sağlanmaktadır.

$f^{-1}(-v) = 1 - f^{-1}(v)$ eşitliğinin sağlandığı aşağıdaki gibi görülebilir:

$$f^{-1}(-v) = arcsin(tanh(-v)) + 0{,}5 = -arcsin(tanh(v)) + 0.5$$
$$= 1 - (arcsin(tanh(v)) + 0{,}5)$$
$$= 1 - f^{-1}(v) \tag{3.27}$$

Bu koşullar bayes tutarlılık ve riskin minimizasyonu için gerekli olan ve formülle elde edilen kayıp fonksiyonun sağladığı özelliklerdir ancak gösterimlerle yeniden açıklanmıştır. Sağlam özellikler için Bölüm 3.3'te verilmiş olan maddelerin geçerliliği incelenmek istendiğinde kayıp fonksiyonun türevi için Şekil 3.8 çizilerek $\phi'(\infty) = \phi'(-\infty) = 0$ olduğu görülmüştür.



Şekil 3.8. Kayıp Fonksiyon ve Türev Değerleri

Şekil 3.8'de görüldüğü gibi yanlış sınıflandırma için sınırlandırılmış ceza uygulanmakta ($\phi(-\infty) = k_1 < \infty$); doğru sınıflandırma için yanlış sınıflandırmadan daha az olacak şekilde sınırlandırılmış ceza uygulanmaktadır. ($0 < \phi(\infty) = k_2 < k_1$). Gudermannian kayıp fonksiyonu ayrım sınır cezalandırması koşulu $\phi(v) = [2\arcsin(\tanh(v)) - 1]^2$ fonksiyonu için $\phi(v) = 0{,}5 > 0$'dır. Fonksiyonun sağlam kayıp için gerekli özellikleri sağladığı görülmüştür. Gudermannian kayıp fonksiyonu için $\gamma(v) = f^{-1}(-v)xJ'[f^{-1}(-v)]$ ve $-J[v] = C_\phi^*[v]$ olmak üzere gerekli koşullar Eşitlik (3.28)- Eşitlik (3.33)'te incelenmiştir:

$$[f^{-1}]'(\infty) = \sqrt{1 - tanh(\infty)^2} = 0 \tag{3.28}$$

$$[f^{-1}]'(-\infty) = \sqrt{1 - tanh(-\infty)^2} = 0 \tag{3.29}$$

$$f^{-1}(\infty) = arcsinh(tan(\infty)) + 0{,}5 = 2.5708 > \frac{1}{2} = f^{-1}(0) \tag{3.30}$$

$$C_\phi^*(0.5) = 4(0{,}5)(0{,}5) = 1 > 0 \tag{3.31}$$

$$C_{\phi}^{*}[f^{-1}(\infty)] + \gamma(\infty)$$

$$= 4[arcsin(tanh(\infty)) + 0{,}5]\left[0{,}5 - arcsin\left(tanh\binom{\infty}{z}\right)\right]$$

$$+ [arcsin(tanh(-\infty)) + 0{,}5]4[2(arcsin(tanh(-\infty)) + 0{,}5) - 1]$$

$$= 4\left[\frac{\pi+1}{2}\right]\left[\frac{1-\pi}{2}\right] + 4\left[\frac{1-\pi}{2}\right][-\pi] = 1 - \pi^2 + 2\pi^2 - 2\pi = (\pi-1)^2$$

$$> 0 \tag{3.32}$$

$$C_{\phi}^{*}[f^{-1}(\infty)] + \gamma(-\infty)$$

$$= 4[arcsin(tanh(\infty)) + 0{,}5][0{,}5 - arcsin(tanh(\infty))]$$

$$+ [arcsin(tanh(\infty)) + 0{,}5]4[2(arcsin(tanh(\infty)) + 0{,}5) - 1]$$

$$= 4\left[\frac{\pi+1}{2}\right]\left[\frac{1-\pi}{2}\right] + 4\left[\frac{1+\pi}{2}\right][\pi] = 1 - \pi^2 + 2\pi^2 + 2\pi = (\pi+1)^2$$

$$< \infty \tag{3.33}$$

Tüm koşulların sağlandığı eşitliklerle de gösterilerek Masnadi-Shirazi [26]'de sağlam kayıp fonksiyon özellikleri olarak tanımlanan tüm maddeler incelenmiştir. Gudermannian kayıp fonksiyonu kullanılarak GudermannianBoost algoritması yazılabilir. Ancak öncesinde TanjantBoost algoritmasında istatistiksel tutarlılık için olasılık değerlerinde bir düzeltme yapılmıştır.

3.7. TanjantBoost Algoritmasında Bir Düzeltme

Masnadi-Shirazi vd. [38]'nin önerdikleri algoritmada başlangıç olasılık değerleri sınıflandırıcı fonksiyon kullanılarak yeniden hesaplanmaktadır. Ancak verilen algoritmanın olasılık değerleri 0-1 arasında olmadığından hem başlangıç olasılık değerleri hem de bu değerler üzerinden hesaplanan ağırlık değerleri hesaplanamamaktadır. Kobetski ve Sullivan [27], olasılık değerleri üzerindeki sorunu görüp w ağırlıkları üzerinden gradyan azalış yöntemi ile çözüm üretmişlerdir. Ancak yapılan benzetim çalışmasında TanjantBoost sonuçları, diğer yöntemlere göre iyi sonuçlar vermemiştir. Çalışmanın bu bölümünde LojitBoost algoritma mantığı [62] kullanılarak R programında TanjantBoost kodlaması yazılmış ve p değerini 0-1 aralığına indirgeyen gerekli düzeltme yapılmıştır. Düzenleme sonucunda sınıflandırıcı fonksiyon değeri negatif olan gözlemin olasılığı 0,5 altında ve sınıflandırıcı fonksiyon değeri pozitif olan gözlemin olasılığı 0,5 üstünde olacak şekilde düzenlenmiş, ayrıca ağırlık değerleri de olasılık değerleri üzerinden bulunduğu için sınıflandırma, istatistiksel sınıflandırma yöntemine benzetilmiştir. Algoritmalar

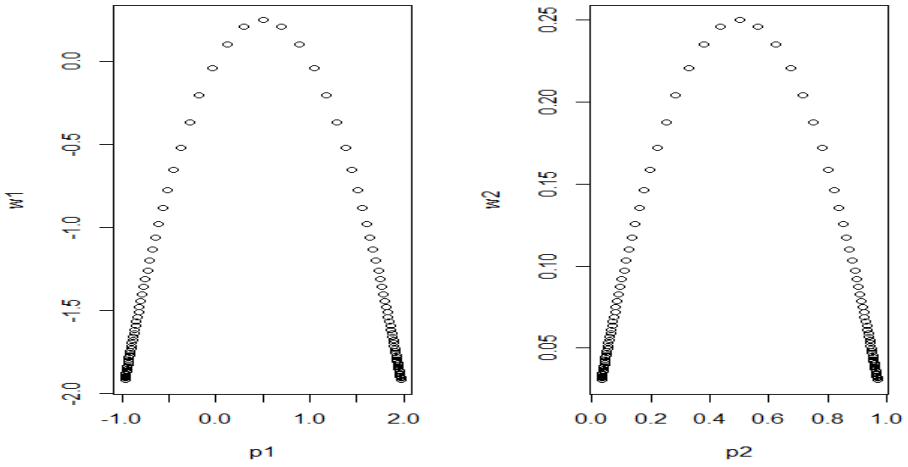
Çizelge 3.1'de verilmiştir. Algoritmada değiştirilen olasılık değeri p ve yeniden formüle edilmiş p değeri Eşitlik (3.34)'teki gibidir:

$$p_{Eski} = arctan(f) + .5$$

$$p_{Yeni} = \frac{arctan(f) - arctan(-\infty)}{arctan(\infty) - arctan(-\infty)} \tag{3.34}$$

Belli aralıktaki sınıflandırıcı fonksiyon değeri için olasılık ve ağırlık değerlerini veren grafikler çizdirilmiştir. Elde edilen grafikler incelendiğinde p değerinin doğru atanmasının w ağırlık değerlerindeki doğruluğu sağladığı görülmektedir.

w değerinin p olasılık değeriyle ilişkisi Şekil 3.9 üzerinden incelenebilir.



Şekil 3.9. TanjantBoost Algoritmaları Ağırlık Değerleri

p değerinin 0,5 'ten küçük olması -1 sınıfında yer alması; 0,5'ten büyük olması ise +1 sınıfında yer alması durumudur. 0,5 civarındaki p değerine sahip gözlemlerin sınıflarının belirlenmesi zordur. Ayrıca bu p değerlerinin w değerleri en büyük olanlardır. Bu değerlere göre minimizasyon yapılması yöntemin ilk adımıdır.

TanjantBoost yönteminde olasılıklar ve ağırlıklar için elde edilebilecek değerler Şekil 3.10'da çizdirilmiştir. Şekilden de izlenebileceği gibi 0,5 olasılık değerine yakın gözlemlerin ağırlıkları artmaktadır. Bu durum sayesinde sınıflandırıcıya yakın olanlara daha fazla ağırlık vererek ayrım tabanlı bir çözüm sağlamaya çalışılmaktadır.

Çizelge 3.1. TanjantBoost ve Düzeltilmiş TanjantBoost Algoritması

TanjantBoost Algoritması

Girdiler: $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ veri kümesi olmak üzere, $y \epsilon \{1, -1\}$ sınıfları belirtirken M, zayıf sınıflandırıcıların sayısını belirlemektedir.

Başlangıç Değerleri: Her i için $\eta^{(1)}(x_i) = 0{,}5$'dir. $\hat{f}^{(1)}(x) = 0$

Döngü 1.

$m = \{1,2, \dots, M\}$ için aşağıdaki işlemler devam ettirilir.

$y = 1$ sınıfı için $z_i^{(m)}$ aşağıdaki eşitlikteki gibi alınır:

$$z_i^{(m)} = -(\eta - 1)(1 + tan^2(\eta - .5))$$

$y = -1$ sınıfı için $z_i^{(m)}$ aşağıdaki eşitlikteki gibi alınır:

$$z_i^{(m)} = -\eta(1 + tan^2(\eta - .5))$$

Ağırlıklar ise aşağıdaki formülle elde edilir:

$$w_i^{(m)} = \eta^{(m)}(x_i)\left(1 - \eta^{(m)}(x_i)\right)$$

$$w_i^{(m)} = w_i^{(m)} / toplam(w_i^{(m)})$$

Döngü 2.

$k = \{1,2, \dots, K\}$ için $\langle q(x_i)\rangle_m = \sum_i w_i^{(m)} q(x_i)$ olmak üzere aşağıdaki en küçük kareler işlemi çözümlenir:

$$a_{\phi_k} = \frac{\langle 1\rangle_w \langle \phi_k(x_i) z_i\rangle_w - \langle \phi_k(x_i)\rangle_w \langle z_i\rangle_w}{\langle 1\rangle_w \langle \phi_k^2(x_i)\rangle_w - \langle \phi_k(x_i)\rangle_w^2}$$

$$b_{\phi_k} = \frac{\langle \phi_k(x_i)^2\rangle_w \langle z_i\rangle_w - \langle \phi_k(x_i)\rangle_w \langle \phi_k(x_i) z_i\rangle_w}{\langle 1\rangle_w \langle \phi_k^2(x_i)\rangle_w - \langle \phi_k(x_i)\rangle_w^2}$$

Döngü 2 Sonu

$$k^* = arg\min_k \sum_i w_i^{(m)} \left(z_i - a_{\phi_k}\phi_k(x_i) - b_{\phi_k}\right)^2$$

İlgili k^* değeri üzerinden riskin minimizasyonunu sağlayan önemli değişken üzerinden sınıflandırıcı fonksiyon elde edilir.

$$\hat{f}^{(m+1)}(x_i) = \hat{f}^{(m)}(x_i) + (a_{\phi_k}\phi_k(x_i) + b_{\phi_k})$$

$$\eta^{(m+1)}(x_i) = \arctan(\hat{f}^{(m+1)}(x_i)) + .5$$

Döngü 1 Sonu

Çıktı: $h(x) = sgn\left[\hat{f}^{(M)}(x)\right]$

Algoritma Sonu.

TanjantBoost (Düzeltilmiş) Algoritması

Girdiler: $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ veri kümesi olmak üzere, $y \epsilon \{1, -1\}$ sınıfları belirtirken M, zayıf sınıflandırıcıların sayısını belirlemektedir.

Başlangıç Değerleri: Her i için $\eta^{(1)}(x_i) = 0{,}5$'dir. $\hat{f}^{(1)}(x) = 0$

Döngü 1.

$m = \{1,2, \dots, M\}$ için aşağıdaki işlemler devam ettirilir.

$y = 1$ sınıfı için $z_i^{(m)}$ aşağıdaki eşitlikteki gibi alınır:

$$z_i^{(m)} = -(\eta - 1)\left(1 + tan^2(\eta - .5)\right)$$

$y = -1$ sınıfı için $z_i^{(m)}$ aşağıdaki eşitlikteki gibi alınır:

$$z_i^{(m)} = -\eta(1 + tan^2(\eta - .5))$$

Ağırlıklar ise aşağıdaki formülle elde edilir:

$$w_i^{(m)} = \eta^{(m)}(x_i)(1 - \eta^{(m)}(x_i))$$

$$w_i^{(m)} = w_i^{(m)} / toplam(w_i^{(m)})$$

Döngü 2.

$k = \{1,2, \dots, K\}$ için $\langle q(x_i)\rangle_m = \sum_i w_i^{(m)} q(x_i)$ olmak üzere aşağıdaki en küçük kareler işlemi çözümlenir:

$$a_{\phi_k} = \frac{\langle 1\rangle_w \langle \phi_k(x_i) z_i\rangle_w - \langle \phi_k(x_i)\rangle_w \langle z_i\rangle_w}{\langle 1\rangle_w \langle \phi_k^2(x_i)\rangle_w - \langle \phi_k(x_i)\rangle_w^2}$$

$$b_{\phi_k} = \frac{\langle \phi_k(x_i)^2\rangle_w \langle z_i\rangle_w - \langle \phi_k(x_i)\rangle_w \langle \phi_k(x_i) z_i\rangle_w}{\langle 1\rangle_w \langle \phi_k^2(x_i)\rangle_w - \langle \phi_k(x_i)\rangle_w^2}$$

Döngü 2 Sonu

$$k^* = arg\min_k \sum_i w_i^{(m)} \left(z_i - a_{\phi_k}\phi_k(x_i) - b_{\phi_k}\right)^2$$

İlgili k^* değeri üzerinden riskin minimizasyonunu sağlayan önemli değişken üzerinden sınıflandırıcı fonksiyon elde edilir.

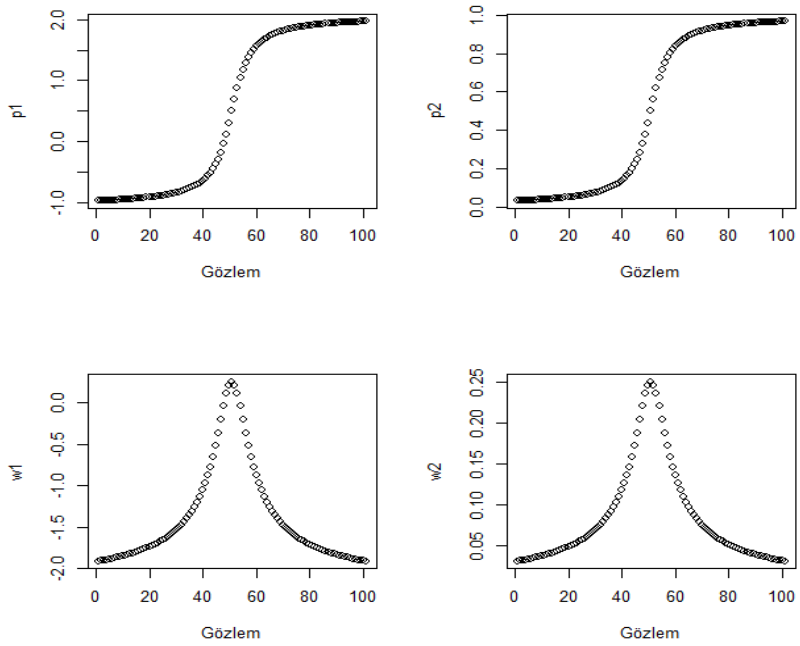
$$\hat{f}^{(m+1)}(x_i) = \hat{f}^{(m)}(x_i) + (a_{\phi_k}\phi_k(x_i) + b_{\phi_k})$$

$$\eta^{(m+1)}(x_i) = \frac{\arctan(\hat{f}^{(m+1)}(x_i)) - \arctan(-\infty)}{\arctan(\infty) - \arctan(-\infty)}$$

Döngü 1 Sonu

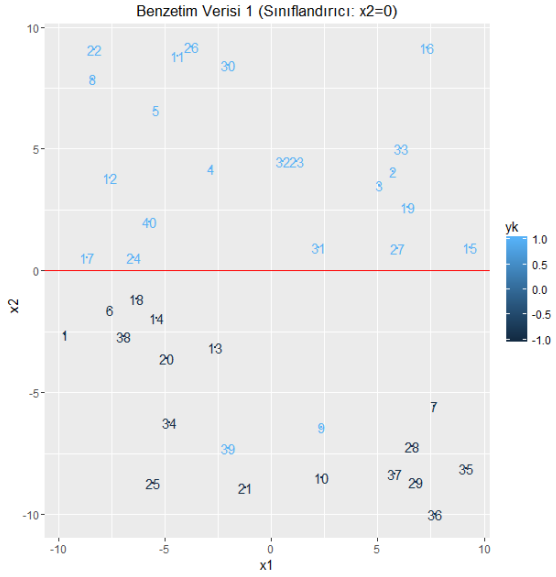
Çıktı: $h(x) = sgn\left[\hat{f}^{(M)}(x)\right]$

Algoritma Sonu.



Şekil 3.10. TanjantBoost Algoritmaları Olasılık ve Ağırlık Değerleri

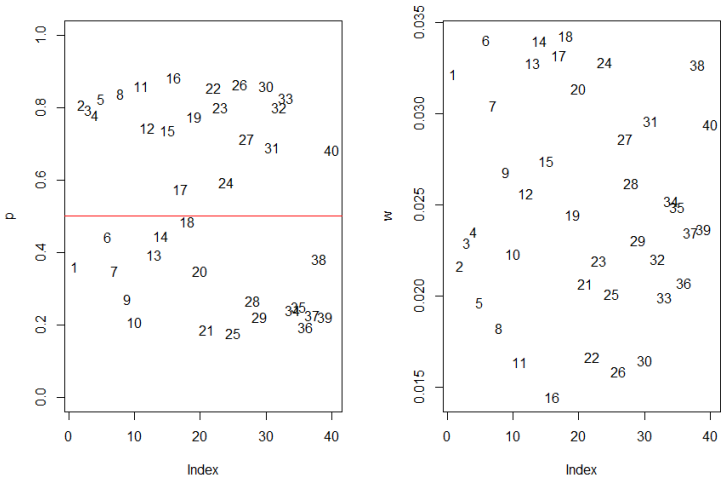
Ağırlıkların TanjantBoost yönteminde önemini görmek için basit bir veri kümesi türetilerek bir sınıflandırma yapılıp sonuçlar incelenebilir. Üretilen veri kümesi, gözlem numaraları üzerinde olacak şekilde gözlemler, gerçek sınıflandırıcı fonksiyonu ve sınıflandırma sonuçları farklı renklerle Şekil 3.11'den incelenebilir:



Şekil 3.11. TanjantBoost Algoritması için Örnek Veri Saçılımı

$x_2 = 0$ doğrusunun üstü 1, altı -1 sınıfı olarak ayrılmıştır. Yanlış sınıflandırma sağlamak için -1 sınıfının içinde belli bir alanda +1 sınıfları oluşturulmuştur. Ağırlıklar ve sınıflar için olasılık değerleri takip edebilmek amacıyla üretilmiş olan 40 gözlem, gözlem numaralarıyla takip edilmiştir. Algoritmanın her adımında en anlamlı bulunan değişkenin üzerinden sınıflandırma yapılmaktadır. Her döngü sonucunda elde edilen fonksiyon değerlerinin toplamı sonucunda gözlemlerin sınıfları belirlenmektedir.

Bir algoritma adımı sonrasında elde edilen olasılık değerleri ve olasılık değerleri üzerinden elde edilen ağırlık değerleri Şekil 3.12'de verilmiştir:



Şekil 3.12. TanjantBoost Algoritmasında Örnek Verinin Olasılık ve Ağırlık Değerleri

Olasılık değerleri incelendiğinde 18. gözlem, -1 sınıfında 0,5'e en yakın gözlem olarak bulunmuştur. 16. gözlem ise +1 sınıfında yer alıp sınıflandırıcıya en uzak pozisyondadır. Ağırlıklardan bu durum incelendiğinde en büyük değeri 18. gözlem alırken 16. gözlem en düşük değeri almıştır. Burada olasılık ve ilgili ağırlık değerleri sınıflandırıcıya uzaklığa göre elde edilirken z değerleri üzerinden de yanlış sınıflandırmanın minimizasyonu yapılmaktadır.

Yöntemin LojitBoost'tan temel farkı, tanjant kayıp fonksiyonu kullanması ve doğru sınıflandırmaya da bir ceza değeri vermesidir. BrownBoost yöntemi de aynı şekilde kayıp fonksiyon ve algoritma değişimi ile daha sağlam bir yapıyı ve kararlı bir sınıflandırıcı oluşturmaya çalışmaktadır.

3.8. GudermannianBoost Yöntemi

Boosting yöntemlerin temel amacı, zayıf öğrenicilerin doğrusal birleşim uzayında gradyan azalış kullanarak deneyimsel riski ($R = \sum_i \phi\big(yf(x)\big)$) minimize etmesidir. $S(x) = \sum_{i=1}^{N} r_i^2(x)$ karesel toplamı için güncelleme adımı $x^{n+1} = x^n + \frac{-r(x)}{\frac{\partial r}{\partial x}}$ şeklindedir. LojitBoost [23], x_i gözlemleri yerine olasılık kestirim değerlerini ($\hat{\eta}(x_i)$) kullanmanın daha uygun olacağı noktasından hareket etmektedir [26]. Gudermannian kayıp fonksiyonu Bölüm 3.5 ve Bölüm 3.6'da verilmiştir. Gudermannian kayıp fonksiyonu için LojitBoost mantığı kullanıldığında $r(\eta) = 2\arcsin\big(\tanh\big(\mathrm{yf}(\eta)\big)\big) - 1$ olur ve bu durumda optimizasyon problemi aşağıdaki gibidir:

$$f^* = arg\min_f \sum_{i=1}^{N} (2\,arcsin\big(tanh\big(yf(\eta)\big)\big) - 1\,)^2 \tag{3.35}$$

Gauss adım güncellemesi için eşitliğin son hali aşağıdaki gibidir:

$$f(\eta)^{n+1} = f(\eta)^n + \Delta f(\eta) = f(\eta)^n - \frac{r(\eta)}{\frac{\partial r}{\partial \eta}} \tag{3.36}$$

$$= f(\eta)^n - \frac{2arcsin(tanh\big(yf(\eta)\big)) - 1}{2y\sqrt{1 - (tanh(f(\eta)y))^2}} \tag{3.37}$$

$$= f(\eta)^n - \frac{arcsin(tanh\big(yf(\eta)\big)) - 0{,}5}{y\sqrt{1 - (tanh(f(\eta)y))^2}} \tag{3.38}$$

Doğrusal regresyon modeli, $z(\eta)$'i yakınsatmak için kullanılabilir. Sınıflar için $z(\eta) = \Delta f(\eta)$ aşağıdaki gibidir:

$$y = 1 \text{ sınıfı için } \mathrm{z}(\eta)_1 = -\left[\frac{\arcsin(\tanh(\mathrm{arcsinh}(\tan(\eta - 0{,}5)))) - 0{,}5}{\sqrt{1-(\tanh(\mathrm{arcsinh}(\tan(\eta-0{,}5))))^2}}\right] \tag{3.39}$$

$$y = -1 \text{ sınıfı için } \mathrm{z}(\eta)_{-1} = -\left[\frac{\arcsin(\tanh(\mathrm{arcsinh}(\tan(\eta - 0{,}5)))) + 0{,}5}{\sqrt{1-(\tanh(\mathrm{arcsinh}(\tan(\eta-0{,}5))))^2}}\right] \tag{3.40}$$

Başka bir formda yazılmak istenirse aşağıdaki gibi elde edilebilir:

$$y = 1 \text{ sınıfı için } \mathrm{z}(\eta)_1 = -\left[\frac{\arcsin\left(\frac{\tan(\eta-0,5)}{\sqrt{1+(\tan(\eta-0,5))^2}}\right)-0,5}{\sqrt{\left[\frac{1-(\tan(\eta-0,5))^2}{1+(\tan(\eta-0,5))^2}\right]}}\right] \tag{3.41}$$

$$y = -1 \text{ sınıfı için } \mathrm{z}(\eta)_{-1} = -\left[\frac{\arcsin\left(\frac{\tan(\eta-0,5)}{\sqrt{1+(\tan(\eta-0,5))^2}}\right)+0,5}{\sqrt{\left[\frac{1-(\tan(\eta-0,5))^2}{1+(\tan(\eta-0,5))^2}\right]}}\right] \tag{3.42}$$

LojitBoost mantığıyla hareket eden TanjantBoost algoritması örnek alındığında GudermannianBoost yöntemi de ağırlıklar üzerinden benzer bir sınıflandırma uygular. GudermannianBoost algoritması Çizelge 3.2'deki gibidir:

Çizelge 3.2. GudermannianBoost Algoritması

Girdiler: $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ veri kümesi olmak üzere, $y\epsilon\{1, -1\}$ sınıfları belirtirken M, zayıf sınıflandırıcıların sayısını belirlemektedir.

Başlangıç Değerleri: Her i için $\eta^{(1)}(x_i) = 0{,}5$'dir. $\hat{f}^{(1)}(x) = 0$

Döngü 1.

$m = \{1,2, \dots, M\}$ için aşağıdaki işlemler devam ettirilir.

$y = 1$ sınıfı için $z_i^{(m)}$ aşağıdaki eşitlikteki gibi alınır:

$$z_i^{(m)} = -\left[\frac{\arcsin\left(\frac{\tan(\eta-0,5)}{\sqrt{1+(\tan(\eta-0,5))^2}}\right)-0,5}{\sqrt{\left[\frac{1-(\tan(\eta-0,5))^2}{1+(\tan(\eta-0,5))^2}\right]}}\right]$$

$y = -1$ sınıfı için $z_i^{(m)}$ aşağıdaki eşitlikteki gibi alınır:

$$z_i^{(m)} = -\left[\frac{\arcsin\left(\frac{\tan(\eta-0,5)}{\sqrt{1+(\tan(\eta-0,5))^2}}\right)+0,5}{\sqrt{\left[\frac{1-(\tan(\eta-0,5))^2}{1+(\tan(\eta-0,5))^2}\right]}}\right]$$

Ağırlıklar ise aşağıdaki formülle elde edilir:

$$w_i^{(m)} = \eta^{(m)}(x_i)\left(1-\eta^{(m)}(x_i)\right)$$

$$w_i^{(m)} = w_i^{(m)}/toplam(w_i^{(m)})$$

Döngü 2.

$k = \{1,2, \dots, K\}$ için $\langle q(x_i)\rangle_m = \sum_i w_i^{(m)} q(x_i)$ olmak üzere aşağıdaki en küçük kareler işlemi çözümlenir:

$$a_{\phi_k} = \frac{\langle 1\rangle_w \langle \phi_k(x_i) z_i\rangle_w - \langle \phi_k(x_i)\rangle_w \langle z_i\rangle_w}{\langle 1\rangle_w \langle \phi_k^2(x_i)\rangle_w - \langle \phi_k(x_i)\rangle_w^2}$$

$$b_{\phi_k} = \frac{\langle \phi_k(x_i)^2\rangle_w \langle z_i\rangle_w - \langle \phi_k(x_i)\rangle_w \langle \phi_k(x_i) z_i\rangle_w}{\langle 1\rangle_w \langle \phi_k^2(x_i)\rangle_w - \langle \phi_k(x_i)\rangle_w^2}$$

Döngü 2 Sonu

$$k^* = arg\min_k \sum_i w_i^{(m)} \left(z_i - a_{\phi_k}\phi_k(x_i) - b_{\phi_k}\right)^2$$

İlgili k^* değeri üzerinden riskin minimizasyonunu sağlayan önemli değişken üzerinden sınıflandırıcı fonksiyon elde edilir.

$$\hat{f}^{(m+1)}(x_i) = \hat{f}^{(m)}(x_i) + (a_{\phi_k}\phi_k(x_i) + b_{\phi_k})$$

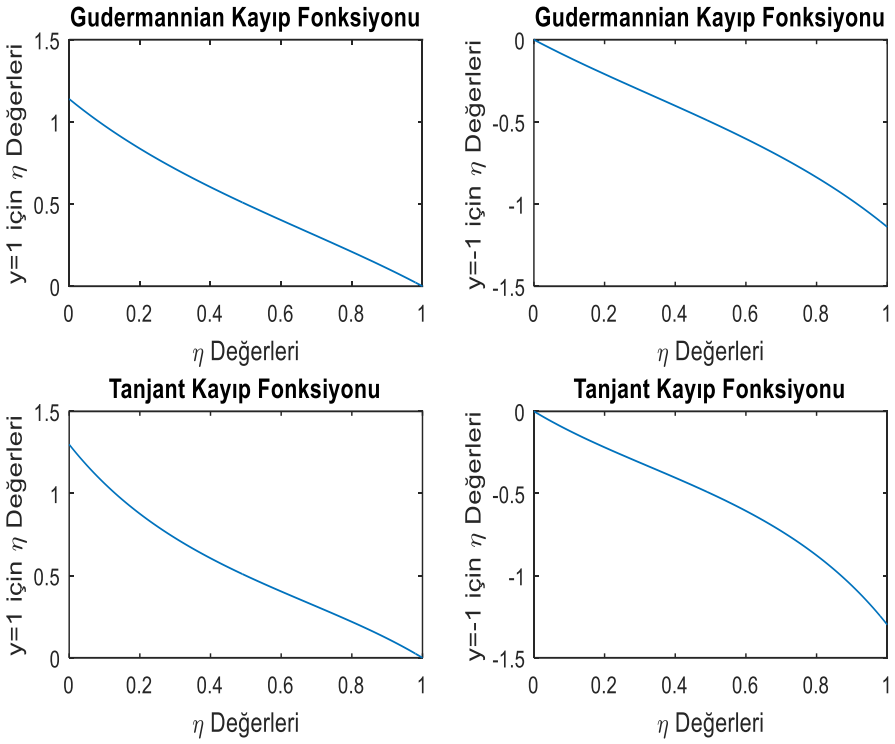
$$\eta^{(m+1)}(x_i) = \frac{\arcsin(\tanh\left(\hat{f}^{(m+1)}(x_i)\right) - \arcsin(\tanh\left(\hat{f}^{(m+1)}(x_i)\right)}{\arcsin(\tanh(\infty)) - \arcsin(\tanh(-\infty))}$$

Döngü 1 Sonu

Çıktı: $h(x) = sgn\left[\hat{f}^{(M)}(x)\right]$

Algoritma Sonu.

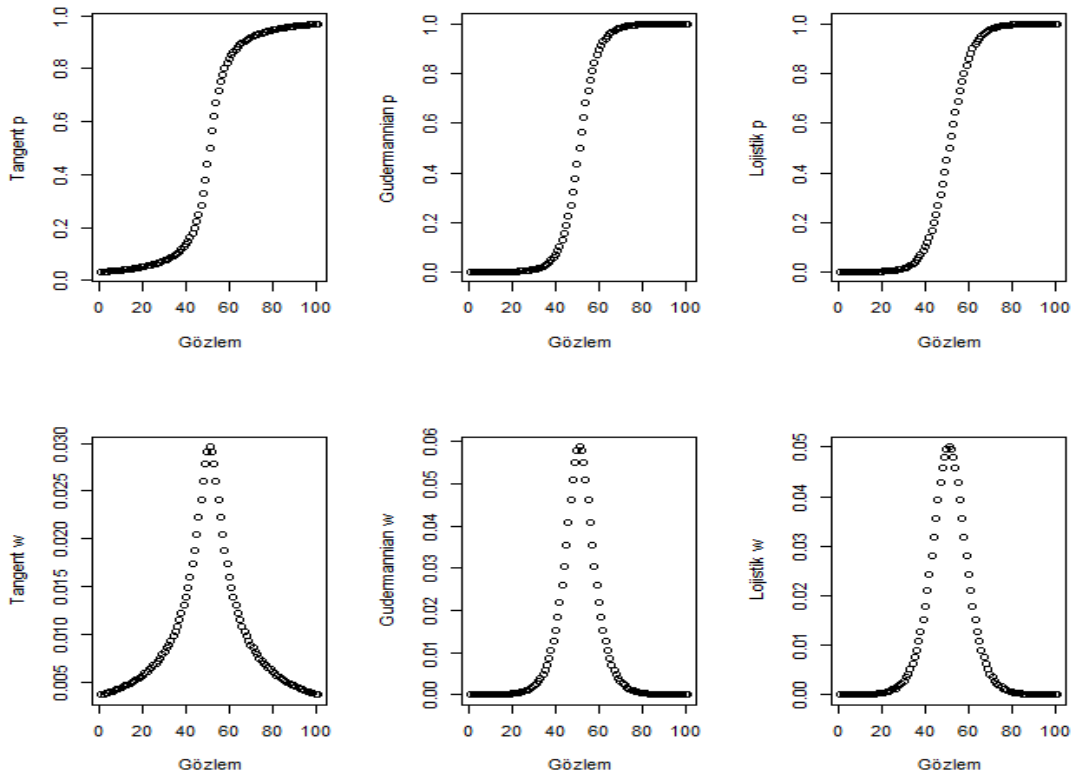
Algoritma aynı olsa da iki yöntemde ortaya çıkan farklılıklar, Gauss adımından, kullanılan Gudermannian fonksiyonundan ve bağlantı fonksiyonundan oluşmaktadır. Ağırlıklandırılmış en küçük kareler çözümlemesi sonucunda elde edilen değişkene ait fonksiyon değeri üzerinden sınıflandırma yapılabilir. Fonksiyonların sınıf değerlerine göre almış olduğu η değerleri Şekil 3.13'ten takip edilebilir:



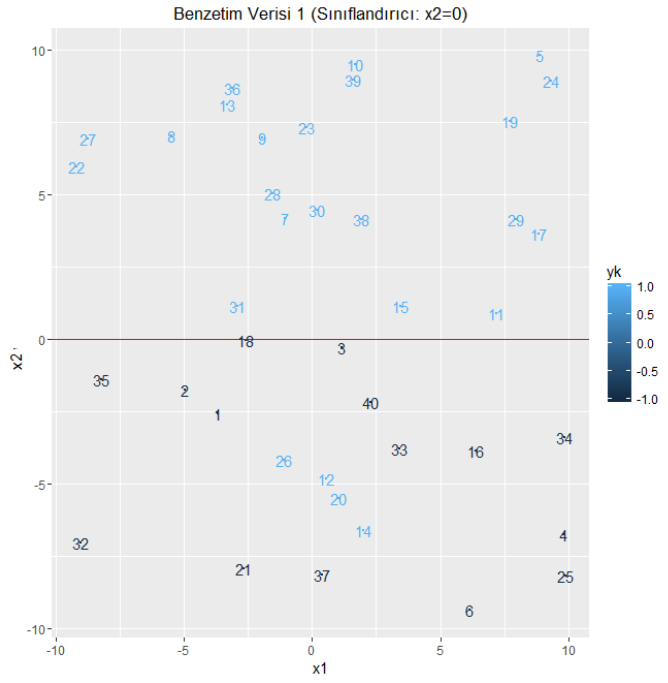
Şekil 3.13. Farklı Sınıflardaki Olasılık Değerleri için Tanjant ve Gudermannian Kayıpları

Gudermannian fonksiyonu, tanjant ve lojistik ile ilgili olasılık ve ağırlık değerleri Şekil 3.14'ten takip edilebilir. TanjantBoost algoritmasının p değerleri, GudermannianBoost ve LojitBoost algoritmasından elde edilen p değerlerinden daha hassas bir yapıya sahiptir. GudermannianBoost algoritması ise sınıflandırıcılara en yakın gözlemlere en büyük ağırlık değerlerini vermektedir. Olasılık değerleri 0-1 değerlerine yaklaştıkça ağırlıklar, LojitBoost algoritmasında olduğu gibi hızlıca düşmektedir. Bu düşüş TanjantBoost algoritmasında daha yavaştır.

TanjantBoost ve GudermannianBoost olasılık ve ağırlık değerlerini incelemek için yine 40 gözlem rasgele üretilmiş ve Şekil 3.15'te verilmiştir.

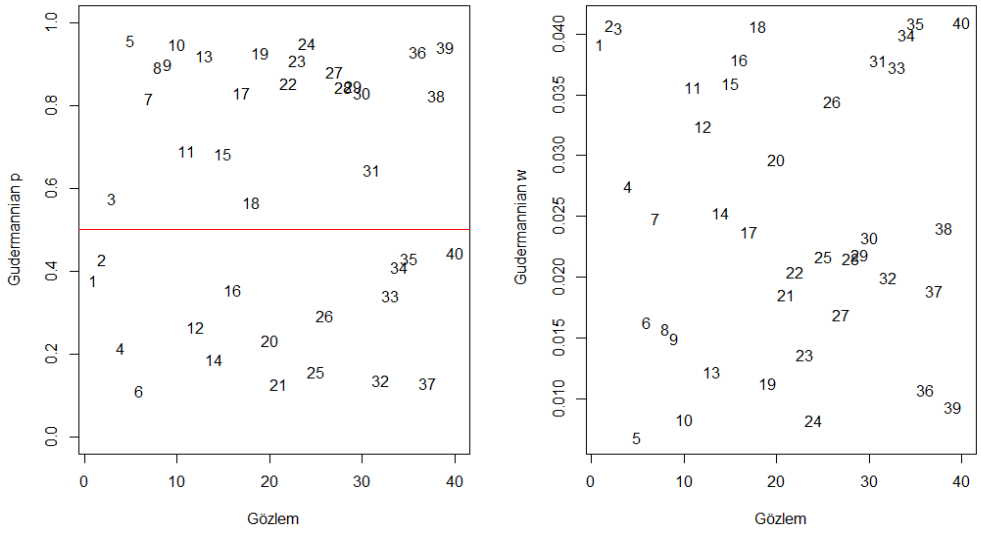


Şekil 3.14. Lojistik, Tanjant ve Gudermannian Olasılık ve Ağırlık Değerleri

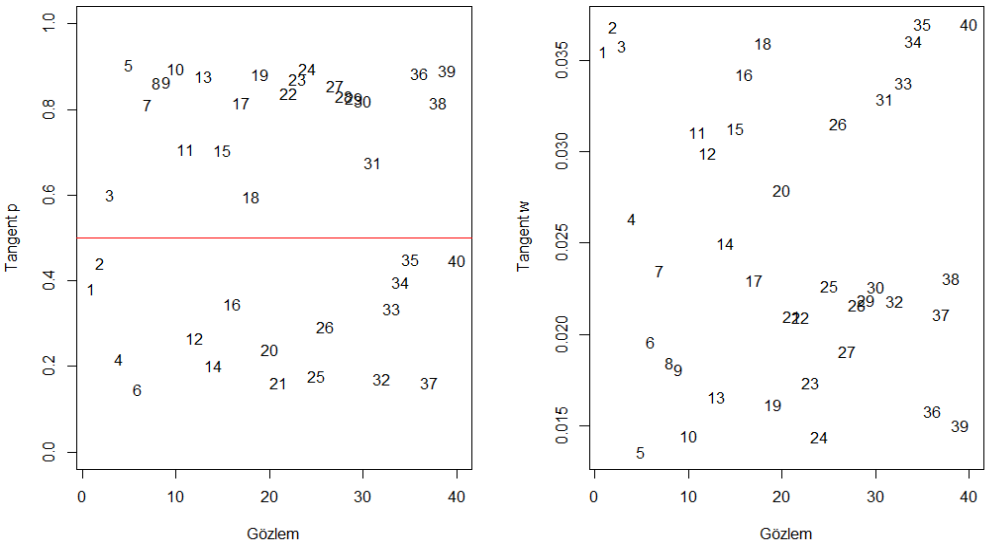


Şekil 3.15. Örnek Bir Veri Sınıflandırması

İlgili kayıp fonksiyonlarının karşılaştırılabilmesi için olasılık değerleri ve olasılık değerleri üzerinden elde edilen ağırlık değerleri Şekil 3.16 ve Şekil 3.17'de verilmiştir.



Şekil 3.16. Örnek veri için GudermannianBoost Olasılık ve Ağırlık Değerleri



Şekil 3.17. Örnek Veri için TanjantBoost Olasılık ve Ağırlık Değerleri

Tanjant olasılık değerleri incelendiğinde 18. gözlem, -1 sınıfında 0,5'e en yakın gözlem olarak bulunmuştur. 5. gözlem ise +1 sınıfında yer alıp sınıflandırıcıya en uzak pozisyonda yer

almaktadır. Ağırlıklardan bu durum incelendiğinde en büyük değeri 18. gözlem alırken 5. gözlem en düşük değeri almıştır. 18. gözlemin yanı sıra +1 sınıfında yer alan 35. ve 40. gözlemlerin de diğer gözlemlere göre yüksek ağırlık değerleri aldığı görülebilir. 35. ve 40. gözlemler yanlış sınıfta bulunan 12, 14, 20, 26 gözlemlerinin sınıflandırıcıya olan etkisini doğru sınıflandırılmış ve ağırlıkları büyük olan gözlemler olarak engellemektedir. Bu durum her ne kadar öğrenme kümesinde doğru sınıflandırma oranını azaltsa da test kümesinde yanlış sınıflandırmadan etkilenmeden doğru sınıflandırma oranını yüksek tutmayı sağlamaktadır. GudermannianBoost ile TanjantBoost mantık olarak aynı hareket eden algoritmalar olsa da GudermannianBoost ağırlık değerlerinin sınıflandırıcıya yakın olan gözlemler için TanjantBoost ağırlık değerlerinden daha büyük olduğu, sınıflandırıcıdan uzaklaşıldıkça ağırlık değerlerinin daha hızlı azaldığı görülmüştür. Bu durum olasılıkların Tanjantboost algoritmasına göre daha homojen olarak dağılmasını sağlamaktadır. Şekil 3.16 ve Şekil 3.17 olasılık dağılımları tablolarından bu durum takip edilebilir.

Bu bölümde, TanjantBoost ve GudermannianBoost yöntemleri, olasılık ve ağırlık değerleri, kayıp fonksiyonlar gibi özellikler teorik olarak verilmiştir. GudermannianBoost yöntemini diğer boosting yöntemlerle karşılaştırmadan önce TanjantBoost ile kıyaslanabileceği bazı özel benzetim kümeleri üretilmiş ve sonuçlar bir sonraki bölümde açıklanmıştır.

4. UYGULAMA

4.1. TanjantBoost ve GudermannianBoost Benzetim Çalışması

R 3.3.0 programında caTools paketi LojitBoost [62] algoritmasındaki fonksiyonlar ve alt döngüler yenilenerek TanjantBoost ve GudermannianBoost kodları yazılmıştır. TanjantBoost ve GudermannianBoost algoritmalarındaki farklılıklar, bağlantı fonksiyonları, bağlantı fonksiyonlarının tersleri ve yöntemlerin olasılık değerleridir. Olasılık değerlerinin farklı bir fonksiyonla üretilmesi ağırlık değerlerini de değiştirmektedir. İki boosting algoritmasındaki farklılık Bölüm 3.5'te anlatıldığı gibidir. Bu bölümde yöntemler öğrenme ve test kümelerindeki doğru sınıflandırma, duyarlılık ve seçicilik (sensitivity and specificity) bakımından karşılaştırılacaktır. Benzetim çalışması için farklı yapıya sahip dört benzetim verisi üretilmiştir:

Benzetim verisi 1: Sınıflandırıcıya yakın tek bir yönden yanlış sınıflandırılmış verilerin olduğu durumdur.

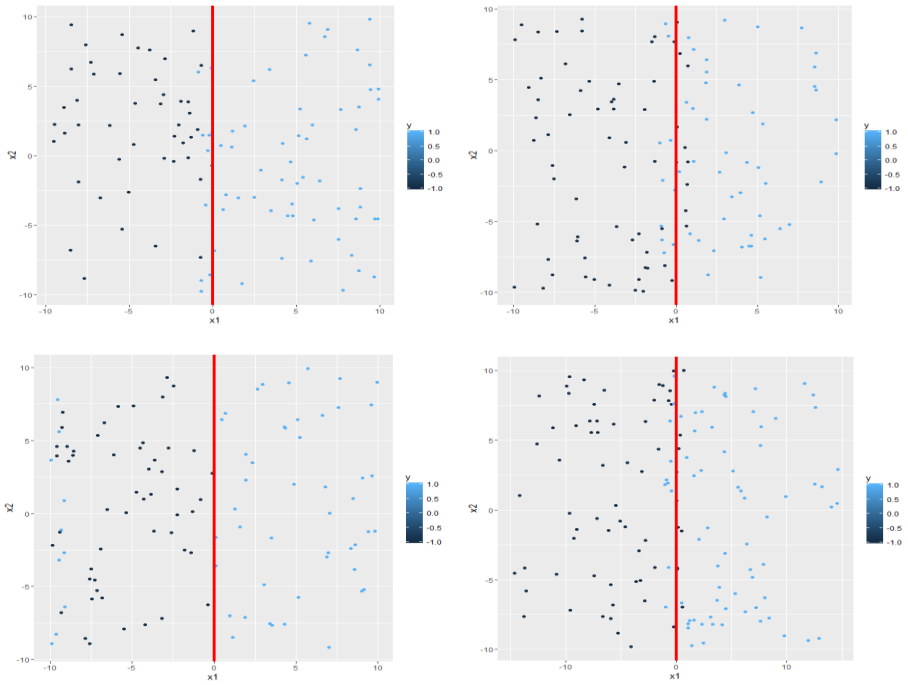
Benzetim verisi 2: Sınıflandırıcıya yakın iki yönden yanlış sınıflandırılmış verilerin olduğu durumdur.

Benzetim verisi 3: Sınıflandırıcıya uzak tek yönden yanlış sınıflandırılmış verilerin olduğu durumdur.

Benzetim verisi 4: Sınıflandırıcıya uzak iki grubun bulunduğu ve sınıflandırıcıya yakın iki yönden de yanlış sınıflandırılmanın olduğu durumdur.

Üretilmiş olan örnek veriler Şekil 4.1'de verilmiştir. Yanlış sınıflandırmalar öğrenme veri kümesi için aykırı değer olarak düşünülmüş ve test kümelerinde sadece (-10,10) arasında üretilmiştir. $x_1 = 0$ doğrusu gerçek sınıflandırıcı olarak kabul edilmiştir. Benzetim çalışması için yukarıda belirtilen özelliklere sahip dört veri kümesi için sırasıyla 100, 200 ve 300 gözleme sahip öğrenme kümeleri oluşturulmuş, her birinde sırasıyla %5, %10, %15, %20 yanlış sınıflandırma ya da sınıflandırıcıya uzak ama doğru sınıflandırma gözlemleri türetilmiştir. TanjantBoost ve GudermannianBoost algoritmaları için kendi döngüleri sırasıyla 5, 10, 15 olarak seçilmiştir. Öğrenme sürecinden sonra her bir süreç için 1000 gözlemlik test kümeleri oluşturulmuştur. İki yöntem için de 1000 tekrar sonucunda elde edilen ortalama genel doğruluk oranı, duyarlılık ve seçicilik değerleri, standart sapmaları ile verilmiştir. Kolay incelenebilmesi

için gözlem sayılarının bulunduğu öğrenme kümelerinin sonuçları, sadece 1. benzetim kümesi için verilmiş, diğer veri kümeleri için verilmemiştir.



Şekil 4.1. Benzetim Veri Kümeleri

Yukarıda verilen benzetim verilerine ilişkin elde edilen sonuçlar Çizelge 4.1-Çizelge 4.6'da verilmiştir. Çizelge 4.1'de 1. benzetim kümesinde gözlem sayısının 100, bozulma için eklenen yanlış sınıflandırma oranlarının %5, %10, %15 ve %20; algoritmanın döngü sayılarının sırasıyla 5, 10 ve 15 (A.A.: Algoritma adımı) olduğu durumlar için elde edilen sonuçlar verilmiştir. Sonuçlar incelendiğinde GudermannianBoost algoritması ile elde edilen sonuçların çok az olsa da TanjantBoost algoritmasından daha fazla doğruluk oranı verdiği görülmüştür. İki algoritmada da yanlış sınıflandırma durumlarında duyarlılık ve seçicilik değerlerinin düştüğü görülmektedir. Algoritma döngü sayısı arttırıldığında doğru sınıflandırma oranlarının arttığı görülmektedir. TanjantBoost algoritmasının kullanıldığı boosting yöntemde seçicilik değerleri bazı benzetim adımlarında daha büyüktür. Ancak genel doğruluk oranında etkisi olmamıştır. Sınıflandırıcının tek tarafına yakın yanlış sınıflandırılmış gözlemler olduğunda GudermannianBoost sonuçları daha büyük doğru sınıflandırma oranı elde etmektedir. 1. benzetim kümesinde Çizelge 4.2'de gözlem sayısının 200 ve Çizelge 4.3'te gözlem sayısının 300; yanlış sınıflandırma oranlarının %5, %10, %15 ve %20; algoritmanın döngü sayılarının sırasıyla 5, 10 ve 15 olduğu durumlar için sonuçlar verilmiştir. Sonuçlar gözlem sayısının artışı bakımından incelendiğinde doğruluk

oranlarının çok az miktarda arttığı, duyarlılık değerinin düştüğü ve seçicilik değerinin arttığı görülmüştür. Yanlış sınıflandırmanın olduğu sınıfta tahminlerin doğruluğunun düştüğü, diğer grupta daha yüksek doğru sınıflandırma olduğu görülmektedir. Ancak yanlış sınıflandırma %5 olduğunda duyarlılığın %5'ten küçük olması, %20 olduğunda duyarlılıktaki oranın %20 kadar etkilenmemesi öğrenme veri kümesindeki sınıflandırıcının aradaki fark kadar etkilendiğini göstermektedir.

Çizelge 4.4'te 2. benzetim kümesinde gözlem sayısının 100, bozulma için eklenen yanlış sınıflandırma oranları %5, %10, %15 ve %20; döngü sayılarının sırasıyla 5, 10 ve 15 olduğu durumlar için sonuçlar verilmiştir. Sonuçlar incelendiğinde GudermannianBoost algoritması ile elde edilen sonuçların TanjantBoost algoritmasından daha az doğruluk oranı verdiği görülmüştür. Sınıflandırıcının iki tarafında da yanlış sınıflandırma olması durumunda öğrenme kümesi için TanjantBoost daha başarılı sonuçlar vermiştir. Öğrenme veri kümelerinin tahmin sonuçları incelendiğinde yanlış sınıflandırma oranının artması öğrenme kümelerinde genel doğruluk oranında azalmalara sebep olmaktadır. Hem duyarlılık hem de seçicilik değerlerinin yanlış sınıflandırma oranı arttıkça düşmesi iki sınıfın da yanlış sınıflandırılan gözlemlerinin öğrenme kümesindeki etkisinden kaynaklanmaktadır. Bu durum hem GudermannianBoost hem de TanjantBoost algoritmalarının yanlış sınıflandırmadan az etkilendiğini göstermektedir. Ancak sınıflandırıcıya yakın yanlış sınıflandırılmış gözlemlerin, öğrenme kümesine olan etkisi TanjantBoost yönteminin yüksek doğruluk oranına sahip olmasına sebep olmuş olabilir. Bu sonucun etkisi test kümesinde daha kolay aktarılabilir. Eğer test kümesinde GudermannianBoost yönteminin daha yüksek doğruluk oranına sahip olduğu görülürse TanjantBoost algoritmasının sınıflandırıcıya yakın yanlış sınıflandırılmış gözlemlere daha duyarlı olduğu sonucu ortaya çıkacaktır. İki yöntemin de önemli dezavantajı vardır: Birbirinden ayrılması çok zor olan veri kümelerinde sınıflandırıcı fonksiyon, gerçekten başka sınıfta olan gözlemi yanlış sınıflandırılmış gözlem kümesi olarak düşünebilir. Yanlış sınıflandırmalara karşı daha duyarsız kayıp fonksiyon kullanılırken amaç, sınıflandırıcıya yakın yanlış sınıflandırma etkisini test kümelerinde azaltarak daha başarılı tahmin yapmaktır. Çok karmaşık yapıya sahip veri kümelerinde ayrılabilir farklı bir uzayda sınıflandırma yapmak dezavantajı engelleyebilir. DVM'deki çekirdek fonksiyonların kullanımı, ayrım tabanlı boosting algoritmasında da sorunlara çözüm getirebilir.

Çizelge 4.1. Benzetim Verisi 1'de 100 Öğrenme Gözlemi için Sonuçlar

Benzetim Verisi 1			**Genel Doğruluk Oranı**				**Duyarlılık**				**Seçicilik**			
			GB		TB		GB		TB		GB		TB	
n	Bozulma	A.A.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.
100	0,05	5	**0,95362**	0,017	0,95337	0,017	**0,95416**	0,042	0,95311	0,043	0,94756	0,042	0,**94785**	0,043
100	0,05	10	**0,95577**	0,017	0,95477	0,017	**0,95571**	0,038	0,95333	0,040	**0,95167**	0,037	0,95148	0,038
100	0,05	15	**0,95758**	0,016	0,95664	0,016	**0,95944**	0,031	0,95849	0,032	**0,95280**	0,033	0,95133	0,035
100	0,1	5	**0,94247**	0,021	0,94231	0,021	**0,91851**	0,050	0,91717	0,051	0,95706	0,049	**0,95769**	0,049
100	0,1	10	**0,94401**	0,022	0,94325	0,022	**0,92068**	0,043	0,91769	0,046	0,95949	0,044	**0,96009**	0,045
100	0,1	15	**0,94676**	0,021	0,94574	0,021	**0,92227**	0,038	0,92172	0,040	**0,96418**	0,038	0,96211	0,040
100	0,15	5	**0,93814**	0,022	0,93759	0,022	**0,87997**	0,053	0,87795	0,054	0,97883	0,037	**0,97927**	0,037
100	0,15	10	**0,94076**	0,020	0,93935	0,020	**0,88489**	0,047	0,88117	0,049	0,98065	0,032	**0,98067**	0,032
100	0,15	15	**0,94346**	0,020	0,94250	0,020	**0,88825**	0,041	0,88751	0,043	**0,98367**	0,029	0,98210	0,030
100	0,2	5	**0,93294**	0,021	0,93209	0,022	**0,84681**	0,058	0,84437	0,060	0,99138	0,022	**0,99152**	0,023
100	0,2	10	**0,93608**	0,018	0,93428	0,018	**0,85325**	0,048	0,84877	0,050	**0,99311**	0,017	0,99289	0,018
100	0,2	15	**0,93838**	0,017	0,93717	0,018	**0,85794**	0,044	0,85602	0,046	**0,99403**	0,015	0,99290	0,018

Çizelge 4.2. Benzetim Verisi 1'de 200 Öğrenme Gözlemi için Sonuçlar

Benzetim Verisi 1			**Genel Doğruluk Oranı**				**Duyarlılık**				**Seçicilik**			
			GB		TB		GB		TB		GB		TB	
n	Bozulma	A.A.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.
200	0,05	5	**0,95420**	0,012	0,95408	0,012	**0,95543**	0,031	0,95461	0,031	0,95045	0,029	**0,95084**	0,030
200	0,05	10	**0,95530**	0,012	0,95475	0,012	**0,95551**	0,028	0,95343	0,029	0,95281	0,028	**0,95337**	0,030
200	0,05	15	**0,95645**	0,012	0,95616	0,012	**0,95594**	0,025	0,95583	0,026	**0,95514**	0,026	0,95447	0,027
200	0,1	5	**0,94398**	0,016	0,94382	0,016	**0,91758**	0,037	0,91612	0,038	0,96311	0,038	**0,96394**	0,038
200	0,1	10	**0,94571**	0,017	0,94505	0,016	**0,91773**	0,032	0,91490	0,033	0,96668	0,035	**0,96759**	0,035
200	0,1	15	**0,94715**	0,016	0,94643	0,016	0,91860	0,027	**0,91867**	0,028	0,96933	0,029	**0,96769**	0,030
200	0,15	5	**0,94220**	0,015	0,94163	0,015	**0,88133**	0,037	0,87918	0,038	0,98709	0,023	**0,98766**	0,022
200	0,15	10	**0,94234**	0,014	0,94100	0,014	**0,87958**	0,033	0,87625	0,034	0,98909	0,020	**0,98911**	0,020
200	0,15	15	**0,94582**	0,013	0,94523	0,014	**0,88416**	0,029	0,88398	0,031	**0,99206**	0,015	0,99096	0,017
200	0,2	5	**0,93569**	0,014	0,93477	0,014	**0,84789**	0,038	0,84557	0,038	0,99758	0,008	**0,99763**	0,008
200	0,2	10	**0,93532**	0,013	0,93369	0,014	**0,84764**	0,036	0,84345	0,037	0,99693	0,010	**0,99698**	0,010
200	0,2	15	**0,93930**	0,012	0,93879	0,012	**0,85581**	0,031	0,85494	0,032	**0,99825**	0,006	0,99787	0,007

Çizelge 4.3. Benzetim Verisi 1'de 300 Öğrenme Gözlemi için Sonuçlar

Benzetim Verisi 1			**Genel Doğruluk Oranı**				**Duyarlılık**				**Seçicilik**			
			GB		TB		GB		TB		GB		TB	
n	Bozulma	A.A.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.
300	0,05	5	**0,95421**	0,010	0,95415	0,010	**0,95502**	0,025	0,95417	0,026	0,95178	0,025	**0,95235**	0,025
300	0,05	10	**0,95497**	0,010	0,95483	0,010	**0,95468**	0,024	0,95308	0,025	0,95354	0,024	**0,95454**	0,025
300	0,05	15	**0,95530**	0,010	0,95502	0,010	0,95525	0,020	0,**95542**	0,021	**0,95433**	0,021	0,95346	0,022
300	0,1	5	**0,94512**	0,014	0,94504	0,014	**0,91550**	0,029	0,91408	0,029	0,96801	0,031	**0,96900**	0,031
300	0,1	10	**0,94635**	0,013	0,94602	0,013	**0,91430**	0,026	0,91187	0,027	0,97145	0,028	**0,97270**	0,028
300	0,1	15	**0,94838**	0,014	0,94772	0,014	**0,91592**	0,022	0,91583	0,023	**0,97421**	0,024	0,97285	0,025
300	0,15	5	**0,94330**	0,012	0,94267	0,012	**0,87790**	0,031	0,87575	0,031	**0,99232**	0,015	0,99282	0,015
300	0,15	10	**0,94400**	0,011	0,94279	0,011	**0,87896**	0,027	0,87557	0,028	0,99318	0,014	**0,99354**	0,014
300	0,15	15	**0,94555**	0,011	0,94518	0,011	0,88202	0,023	**0,88207**	0,024	**0,99392**	0,012	0,99311	0,014
300	0,2	5	**0,93539**	0,011	0,93457	0,011	**0,84499**	0,031	0,84288	0,031	0,99912	0,004	**0,99919**	0,004
300	0,2	10	**0,93536**	0,011	0,93383	0,011	**0,84510**	0,029	0,84129	0,030	0,99910	0,004	**0,99912**	0,004
300	0,2	15	**0,93944**	0,009	0,93898	0,010	**0,85449**	0,024	0,85355	0,026	**0,99930**	0,004	0,99908	0,004

Çizelge 4.4. Benzetim Verisi 2'de 100 Öğrenme Gözlemi için Sonuçlar

Benzetim Verisi 2			**Genel Doğruluk Oranı**				**Duyarlılık**				**Seçicilik**			
			GB		TB		GB		TB		GB		TB	
n	Bozulma	A.A.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.
100	0,05	5	0,91530	0,015	**0,91532**	0,015	**0,91447**	0,044	0,91439	0,045	**0,91017**	0,044	0,91007	0,045
100	0,05	10	**0,91757**	0,016	0,91716	0,016	**0,91547**	0,042	0,91484	0,043	**0,91426**	0,041	0,91356	0,042
100	0,05	15	**0,91699**	0,014	0,91687	0,014	**0,91403**	0,036	0,91359	0,038	**0,91565**	0,034	0,91535	0,036
100	0,1	5	0,85777	0,021	**0,85797**	0,021	0,85518	0,058	**0,85559**	0,059	**0,85227**	0,061	0,85203	0,061
100	0,1	10	0,85776	0,020	**0,85847**	0,020	0,85315	0,054	**0,85411**	0,056	**0,85574**	0,053	0,85570	0,055
100	0,1	15	0,85588	0,020	**0,85596**	0,020	0,85300	0,047	**0,85321**	0,048	**0,85339**	0,048	0,85281	0,049
100	0,15	5	0,80618	0,025	**0,80658**	0,026	0,80231	0,068	**0,80269**	0,069	0,80161	0,070	**0,80183**	0,071
100	0,15	10	0,80732	0,026	**0,80873**	0,026	0,80583	0,063	**0,80712**	0,065	0,80106	0,065	**0,80194**	0,067
100	0,15	15	0,80385	0,025	**0,80445**	0,025	0,80178	0,055	**0,80188**	0,057	0,80008	0,056	**0,80061**	0,057
100	0,2	5	0,76305	0,031	**0,76365**	0,031	0,76104	0,080	**0,76157**	0,081	0,75612	0,077	**0,75659**	0,078
100	0,2	10	0,76536	0,032	**0,76604**	0,032	0,76051	0,071	**0,76084**	0,072	0,76316	0,071	**0,76364**	0,073
100	0,2	15	0,75869	0,031	**0,75894**	0,031	**0,75539**	0,060	0,75537	0,061	0,75645	0,060	**0,75651**	0,061

Çizelge 4.5'te 3. benzetim kümesinde gözlem sayısının 100, bozulma için eklenen yanlış sınıflandırma oranının %5, %10, %15 ve %20; algoritmanın döngü sayılarının sırasıyla 5, 10 ve 15 olduğu durumlar için sonuçlar verilmiştir. Sonuçlar incelendiğinde yanlış sınıflandırma oranı arttıkça öğrenme kümesinde doğru sınıflandırma oranı oldukça düşmektedir. Algoritmada döngü sayısını arttırmak doğruluk oranını arttırsa da öğrenme kümesinde doğru sınıflandırma oranın hızlıca düşmesi, sadece yanlış sınıflandırma ile değil aynı zamanda sınıflandırıcıya uzakta olan yanlış sınıflandırılmış gözlemlerin etkisiyle de açıklanabilmektedir. Yöntemler karşılaştırıldığında, sınıflandırıcının sadece bir tarafında sınıflandırıcıya uzak ve yanlış sınıflandırılmış veriler için GudermannianBoost algoritması ile elde edilen sonuçların TanjantBoost algoritmasından daha yüksek doğruluk oranı verdiği görülmüştür. Hem duyarlılık hem de seçicilik değerleri yanlış sınıflandırma arttıkça düşmekte ancak yanlış sınıflandırmanın bulunduğu sınıfın tahmininden dolayı duyarlılığın daha hızlı azaldığı görülmektedir.

Çizelge 4.6'da 4. benzetim kümesinde gözlem sayısının 100, bozulma için eklenen yanlış sınıflandırma oranının %5, %10, %15 ve %20; algoritmanın döngü sayılarının sırasıyla 5, 10 ve 15 olduğu durumlar için sonuçlar verilmiştir. Sonuçlar incelendiğinde yanlış sınıflandırma oranı arttıkça öğrenme kümesinde doğru sınıflandırma oranı oldukça düşmektedir. Algoritmada döngü sayısını arttırmak doğruluk oranını arttırmaktadır. Ancak 4. benzetim kümesinde test verilerinde olmayan sınıflandırıcıya uzak doğru sınıflandırmanın etkisi öğrenme kümesinde görülememektedir. Yöntemler karşılaştırıldığında, GudermannianBoost algoritması ile elde edilen sonuçların TanjantBoost algoritmasından daha az doğruluk oranı verdiği görülmüştür. Sınıflandırıcı iki tarafta da yanlış sınıflandırma olması durumunda öğrenme kümesinin tahmininde TanjantBoost daha başarılı sonuçlar vermiştir. Ancak bu sonuç benzetim verisi 2'de olduğu gibi, test kümesinde GudermannianBoost sonuçlarının lehine olursa, TanjantBoost algoritmasının sınıflandırıcıya yakın ve yanlış sınıflandırılmış gözlemlerden etkilendiği anlamına gelecektir. Öğrenme veri kümelerinin sonuçları incelendiğinde yanlış sınıflandırma oranının artması genel doğruluk oranında azalmalara sebep olmaktadır. Hem duyarlılık hem de seçicilik değerlerinin yanlış sınıflandırma arttıkça düşmesi iki sınıfında yanlış sınıflandırılan gözlemlerinin öğrenme kümesindeki etkisinden kaynaklanmaktadır.

Çizelge 4.5. Benzetim Verisi 3'te 100 Öğrenme Gözlemi için Sonuçlar

Benzetim Verisi 3			**Genel Doğruluk Oranı**				**Duyarlılık**				**Seçicilik**			
			GB		TB		GB		TB		GB		TB	
n	Bozulma	A.A.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.
100	0,05	5	**0,91441**	0,024	0,91271	0,025	**0,91869**	0,062	0,91493	0,064	**0,90483**	0,016	0,90481	0,016
100	0,05	10	**0,92388**	0,019	0,91897	0,021	**0,94268**	0,049	0,93112	0,053	0,90232	0,021	**0,90300**	0,021
100	0,05	15	**0,93113**	0,016	0,92412	0,019	**0,95827**	0,039	0,94130	0,046	0,90319	0,019	**0,90451**	0,017
100	0,1	5	**0,83172**	0,032	0,82952	0,032	**0,82400**	0,087	0,81896	0,088	0,83124	0,015	**0,83129**	0,015
100	0,1	10	**0,84644**	0,028	0,83925	0,029	**0,85839**	0,075	0,84171	0,080	0,83016	0,019	**0,83039**	0,019
100	0,1	15	**0,86059**	0,024	0,85030	0,025	**0,89089**	0,065	0,86729	0,069	0,82997	0,018	**0,83039**	0,017
100	0,15	5	**0,74254**	0,041	0,73979	0,041	**0,69689**	0,123	0,69016	0,126	0,76825	0,019	**0,76836**	0,019
100	0,15	10	**0,76396**	0,034	0,75615	0,035	**0,75070**	0,101	0,73219	0,105	0,76652	0,019	**0,76661**	0,019
100	0,15	15	**0,77629**	0,032	0,76655	0,033	**0,77787**	0,095	0,75474	0,099	0,76729	0,020	**0,76744**	0,020
100	0,2	5	**0,65463**	0,043	0,65198	0,043	**0,55287**	0,149	0,54541	0,151	0,71684	0,033	**0,71739**	0,034
100	0,2	10	**0,67656**	0,039	0,66929	0,039	**0,61336**	0,127	0,59516	0,130	0,71262	0,022	**0,71293**	0,023
100	0,2	15	**0,68703**	0,037	0,67906	0,037	**0,63705**	0,120	0,61669	0,123	0,71389	0,023	**0,71435**	0,024

Çizelge 4.6. Benzetim Verisi 4'te 100 Öğrenme Gözlemi için Sonuçlar

Benzetim Verisi 4			**Genel Doğruluk Oranı**				**Duyarlılık**				**Seçicilik**			
			GB		TB		GB		TB		GB		TB	
n	Bozulma	A.A.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.
100	0,05	5	0,92264	0,014	**0,92266**	0,014	**0,91976**	0,043	0,91961	0,044	0,91999	0,043	**0,92003**	0,044
100	0,05	10	**0,92344**	0,014	0,92294	0,014	**0,91918**	0,038	0,91810	0,041	**0,92286**	0,039	0,92245	0,041
100	0,05	15	**0,92333**	0,014	0,92328	0,014	**0,92023**	0,034	0,92008	0,036	**0,92273**	0,034	0,92224	0,036
100	0,1	5	0,88001	0,019	**0,88013**	0,019	**0,87408**	0,055	0,87406	0,055	0,87957	0,055	**0,87966**	0,056
100	0,1	10	0,88043	0,019	**0,88071**	0,019	0,87939	0,051	**0,87956**	0,053	**0,87587**	0,049	0,87578	0,051
100	0,1	15	0,87828	0,018	**0,87906**	0,018	0,87632	0,043	**0,87673**	0,045	0,87599	0,043	**0,87661**	0,045
100	0,15	5	0,84759	0,021	**0,84794**	0,021	0,84388	0,065	**0,84400**	0,066	0,84427	0,063	**0,84472**	0,063
100	0,15	10	0,84854	0,023	**0,84928**	0,023	0,84543	0,057	**0,84621**	0,059	0,84609	0,058	**0,84640**	0,060
100	0,15	15	0,84211	0,021	**0,84294**	0,022	0,84050	0,049	**0,84129**	0,051	0,83924	0,049	**0,83961**	0,051
100	0,2	5	0,82341	0,025	**0,82388**	0,025	0,81944	0,072	**0,81963**	0,072	0,82043	0,070	**0,82108**	0,071
100	0,2	10	0,82306	0,025	**0,82384**	0,025	0,81703	0,059	**0,81757**	0,061	0,82387	0,062	**0,82456**	0,064
100	0,2	15	0,81758	0,025	**0,81879**	0,025	0,81341	0,054	**0,81436**	0,055	0,81719	0,055	**0,81826**	0,057

Çizelge 4.7'de tüm benzetim verileri için modelin daha önce görmemiş olduğu 1000 gözlemlik test verileri üretilmiştir. Bu veri kümeleri yanlış sınıflandırma olmadan üretilmiştir. Elde edilmiş modellerin tahminleri üzerinden doğru sınıflandırma, duyarlılık ve seçicilik değerleri verilmiştir. Benzetim verisi 1'de, öğrenme kümesindeki gözlem sayıları değiştiğinde doğruluk oranında, duyarlılık ve seçicilik değerlerinde çok az da olsa artış gözlenmiştir. Test kümelerinde ortaya çıkan başka bir sonuç ise, 2. ve 4. benzetim verilerinde TanjantBoost öğrenme kümesi için daha iyi sonuç vermiş olsa da test kümesinde bu başarıyı gösterememiştir. Yani GudermannianBoost yönteminin sınıflandırıcıya yakın olan yanlış sınıflandırılmış gözlemlere karşı daha duyarsız olduğu söylenebilir. TanjantBoost öğrenme kümesinde daha yüksek doğru sınıflama oranına sahipken test kümesinde bu başarının düşmesi öğrenme kümesindeki yanlış sınıflandırmanın sınıflandırıcıya olan etkilerinden dolayıdır. GudermannianBoost sonuçları tek bir benzetim hariç, hepsinde TanjantBoost algoritmasından daha iyi sonuç vermiştir. Test kümelerinde en düşük doğruluk oranı, öğrenme verisinde olan ancak test verisinde olmayan sınıflandırıcıya uzak, doğru sınıflandırılmış gözlemler ve sınıflandırıcıya yakın, yanlış sınıflandırılmış gözlemlerin olduğu 4. benzetim verisindedir. Yine de %20 oranında yanlış sınıflandırma olmasına rağmen test kümesinde doğru sınıflandırma oranı %90'ın altına düşmemiştir.

Benzetim kümeleri ayrılabilir sınıflardır. Sınıflarda farklı etiketlere sahip gözlemlerin aykırı değer ya da yanlış sınıflandırma olarak kabul edildiği kümelerdir. Benzetim çalışması sonucunda, test kümeleri arasındaki doğruluk oranları karşılaştırıldığında yanlış sınıflandırmalardan ya da aykırı değerlerden etkilenmeden TanjantBoost ve GudermannianBoost yöntemlerinin iyi sonuçlar verdiği, GudermannianBoost yönteminin TanjantBoost'a göre doğruluk oranını arttırdığı görülmektedir. Ayrıca GudermannianBoost yönteminin sınıflandırıcılara yakın yanlış sınıflandırma durumunda öğrenme kümesinden daha az etkilendiği ve test kümesinde de TanjantBoost yönteminden daha iyi sonuç verdiği görülmüştür. Yöntemler basit sınıflandırıcı mantığıyla çalıştığından her değişkende yanlış sınıflandırmadan etkilenmeyecek şekilde sınıflandırıcının seçilmesi temel mantıktır. İki yöntem de ayrılması zor veri kümelerinde doğru sınıfında bulunan bazı gözlemler yanlış sınıflandırılmış olarak kabul etme dezavantajına sahiptir. Bu durumun aşılması için çekirdek fonksiyonlar kullanılabilir ya da algoritmadaki döngü sayısı arttırılabilir. GudermannianBoost yönteminin daha detaylı incelenmesi için farklı veri kümelerinde diğer boosting yöntemleri ile inceleme yapılabilir. Bir sonraki uygulama çalışması bazı gerçek veri kümelerinde yöntemlerin karşılaştırılmasıdır.

Çizelge 4.7. Tüm Benzetim Verileri için Test Veri Kümesi Sonuçları

			Genel Doğruluk Oranı				**Duyarlılık**				**Seçicilik**			
n	Bozulma	A.A.	GB		TB		GB		TB		GB		TB	
BV1			Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.
1000	0	5	**0,97042**	0,018	0,96977	0,018	**0,94814**	0,040	0,94708	0,041	**0,99269**	0,014	0,99245	0,015
1000	0	10	**0,96966**	0,016	0,96818	0,017	**0,94558**	0,036	0,94331	0,038	**0,99374**	0,011	0,99305	0,012
1000	0	15	**0,97204**	0,014	0,97147	0,015	**0,94822**	0,030	0,94792	0,031	**0,99587**	0,009	0,99502	0,010
BV1														
1000	0	5	**0,97466**	0,013	0,97421	0,013	**0,95151**	0,028	0,95070	0,028	**0,99784**	0,005	0,99776	0,005
1000	0	10	**0,97348**	0,013	0,97258	0,013	**0,94933**	0,027	0,94786	0,029	**0,99759**	0,006	0,99727	0,006
1000	0	15	**0,97411**	0,011	0,97389	0,012	**0,94960**	0,024	0,94952	0,024	**0,99858**	0,004	0,99824	0,005
BV1														
1000	0	5	**0,97543**	0,011	0,97499	0,011	**0,95197**	0,023	0,95111	0,023	**0,99883**	0,004	0,99881	0,004
1000	0	10	**0,97454**	0,011	0,97366	0,012	**0,95025**	0,023	0,94864	0,024	**0,99885**	0,003	0,99869	0,004
1000	0	15	0,97505	0,009	**0,97519**	0,009	0,95063	0,019	**0,95102**	0,019	**0,99951**	0,002	0,99939	0,002
BV2														
1000	0	5	**0,98009**	0,012	0,97840	0,012	**0,97895**	0,024	0,97856	0,024	**0,97853**	0,023	0,97821	0,023
1000	0	10	**0,97872**	0,013	0,97786	0,013	**0,97810**	0,023	0,97728	0,024	**0,97931**	0,021	0,97841	0,022
1000	0	15	**0,98194**	0,010	0,98129	0,011	**0,98126**	0,019	0,98081	0,020	**0,98260**	0,018	0,98177	0,019
BV3														
1000	0	5	**0,90681**	0,039	0,90419	0,040	**0,81468**	0,080	0,80939	0,082	0,99870	0,005	**0,99873**	0,005
1000	0	10	**0,92030**	0,034	0,91186	0,036	**0,84442**	0,070	0,82715	0,075	0,99622	0,011	**0,99662**	0,010
1000	0	15	**0,93601**	0,030	0,92474	0,032	**0,87596**	0,061	0,85248	0,064	0,99622	0,010	**0,99718**	0,009
BV4														
1000	0	5	**0,97462**	0,015	0,97412	0,015	**0,97346**	0,029	0,97294	0,030	**0,97579**	0,028	0,97533	0,029
1000	0	10	**0,97504**	0,014	0,97372	0,015	**0,97599**	0,025	0,97467	0,026	**0,97404**	0,027	0,97273	0,028
1000	0	15	**0,97951**	0,012	0,97815	0,012	**0,97914**	0,021	0,97775	0,022	**0,97989**	0,020	0,97856	0,022

4.2. Çeşitli Boosting Yöntemlerde Öğrenme ve Test Kümeleri Doğru Sınıflandırma Oranı Karşılaştırması

Sınıflandırma yöntemleri, gözlenmiş verileri kullanarak öğrenme süreçlerinden geçmekte ve daha sonraki verilerin sınıflarının tahmin edilmesini amaçlamaktadır. Bu bölümde DVM ve boosting yöntemleriyle GudermannianBoost yöntemi karşılaştırılmıştır. Bu karşılaştırma için Kaliforniya Üniversitesi veri tabanındaki dört gerçek veri kullanılmıştır [63]. Veri olarak XOX oyunu verisi (Tic-Toc-Toe) [64], satranç oyunu verisi [65], iflas verisi (qualitative bankruptcy)[66], Avustralya kredi verisi (Credit Approval) [67] kullanılmıştır. Oyun verileri ve iflas verisi klasik karar ağaçları ile %100 doğru sınıflandırılabilecek veri kümeleridir. Bu veri kümelerinde herhangi bir yanlış sınıflandırma ya da aykırılık yoktur. XOX, satranç verilerinde kazanma ve kaybetme durumları belirli şekilde kategorileştirilmiş verilerden oluşurken, iflas verisi de tamamen sistemde gerekli olan durumlarda negatif, orta düzey ve pozitif olarak sınıflandırılmış kategorik verilerden oluşmaktadır. Üç verideki temel farklılık girdi değişken sayılarının çok olması, az olması ve çıktı değişkenlerinde iki sınıfın oranlarının birbirlerinden farklı olmasıdır. Avustralya kredi verisi ise hem nitel hem de nicel verilerden oluşmaktadır. Ayrıca, girdileri ve karar ağaçları ile tam ayrılabilecek bir yapıya sahip değildir. Gerçek veri kümelerindeki doğru sınıflandırma oranları incelenirken seçilmiş olan verilere göre GudermannianBoost yönteminin özellikleri incelenmiştir.

Verilere ve verilerin özellikleri ile ilgili detaylı bilgiye Asuncion and Newman [66]'dan ulaşılabilir. Veri kümeleriyle ilgili genel bir bilgi verilecek olursa, iflas, XOX ve satranç veri kümeleri tam olarak ayrılabilir, yanlış sınıflandırmanın olmadığı ve aslında GudermannianBoost yönteminin en kötü sonuçlar elde edebileceği durumlardır. XOX verisinde %34,7 oranında 1 sınıfı bulunurken, iflas ve satranç veri kümelerinde sırasıyla %57,2 ve %52,2 oranında 1 sınıfı bulunmaktadır. Bu değer Avustralya kredi verisinde %50,40'tır. Gözlem ve değişken sayıları ile sınıf oranları Çizelge 4.8'de verilmiştir.

Çizelge 4.8. Gerçek Veri Kümeleri ile İlgili Bilgiler

Veri kümesi	Değişken Sayısı	Gözlem Sayısı	1 Sınıf Oranı
İflas Verisi	7	250	%57,2
XOX Oyun Verisi	10	958	%34,7
Satranç Oyun Verisi	37	3196	%52,2
Avustralya Kredi Verisi	15	690	%50,4

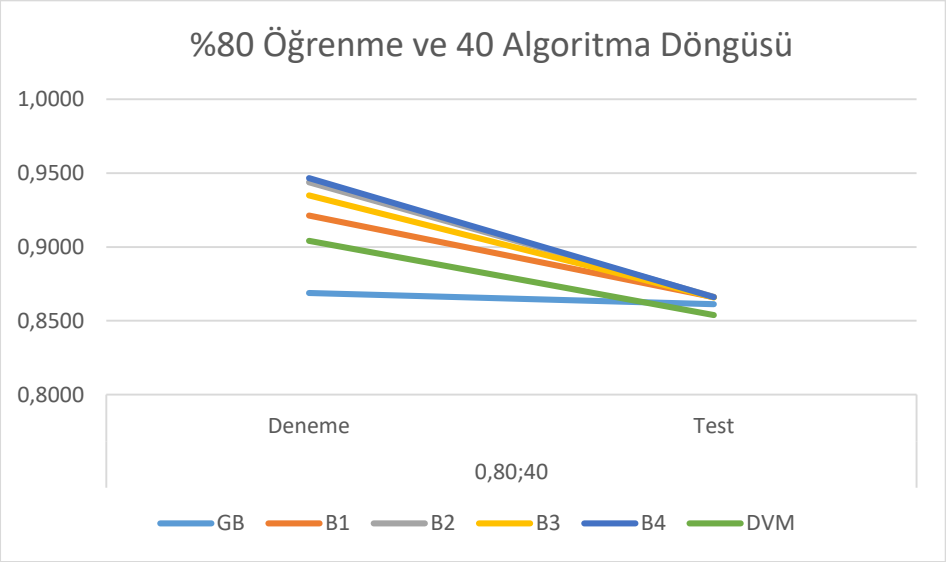
Üç tane tam ayrılabilir veri kümesi hem değişken hem gözlem sayısı hem de sınıf oranları bakımından GudermannianBoost yönteminin doğru sınıflandırma oranına etkisini karşılaştırmak için önemlidir. Avustralya kredi verisi ise günlük hayattandır ve yanlış sınıflandırma (ya da aykırılık, bozulma) olarak düşünülebilecek verileri içinde barındırmaktadır.

Gerçek veri kümelerinden tüm veriler için %60, %70 ve %80 oranında öğrenme kümesi alınarak model kurulmuştur. 40 döngülü algoritma için doğru sınıflandırma ortalamaları ve standart sapmaları Çizelge 4.9'da verilmiştir. Sonuçlar hem öğrenme hem de test kümesi için elde edilmiştir. B1 algoritması üstel kayıp fonksiyonu kullanan RealBoost, B2 algoritması üstel kayıp fonksiyonu kullanan GentleBoost, B3 algoritması lojistik kayıp fonksiyonu kullanan RealBoost, B4 algoritması lojistik kayıp fonksiyonu kullanan GentleBoost algoritmalarıdır. Sonuçlar 250 tekrar için alınmıştır.

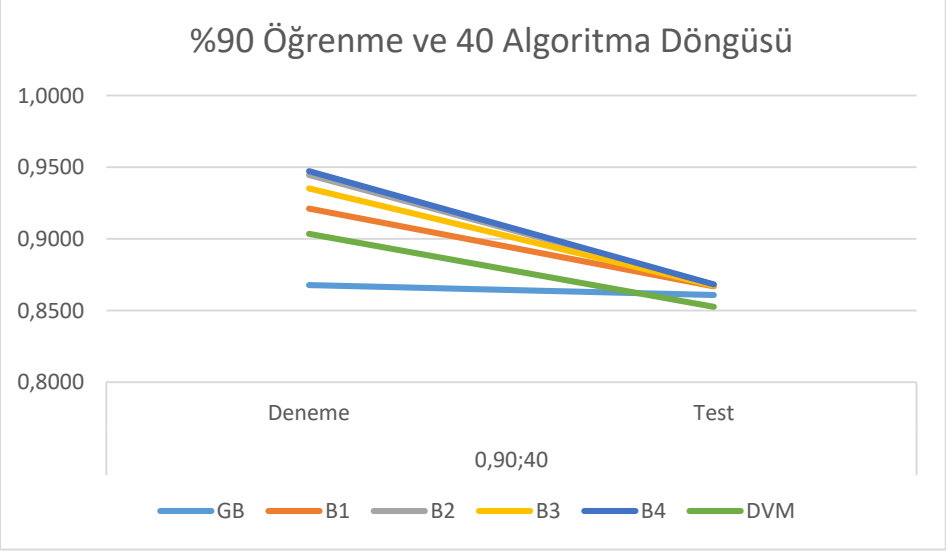
Ayrılabilir veri kümelerinde değişken sayısı ve gözlem sayısı önemli olmaksızın GudermannianBoost yöntemi diğer yöntemlerden daha kötü sınıflandırma sonuçları vermektedir. Ancak XOX kümesinde sınıflandırma sonuçları daha kötüdür. Bu durumun sebebi, sınıf değerlerinin farklı oranlarda olmasından kaynaklanmaktadır. Dengeleme (balancing) yapıldığında sonuçlarda GudermannianBoost yönteminin diğer veri kümelerindeki gibi sonuç vermesi beklenmektedir. Tam ayrılabilir kümelerde GudermannianBoost yöntemi diğer boosting yöntemlerden daha iyi sonuçlar vermemiş olsa da özellikle test kümesinde diğer yöntemlerle benzer doğru sınıflandırma oranları vermiştir. Tam ayrılabilir veri kümeleri yerine karmaşık oranları olan ve aykırılıkların da bulunduğu bir veri kümesinin üzerinden daha detaylı çalışılabilir. Çünkü GudermannianBoost yönteminin diğer boosting yöntemlere göre ayrılabilir veri kümelerinde daha başarılı olması beklenmektedir.

Avustralya kredi verisi Çizelge 4.10 üzerinden incelendiğinde diğer boosting yöntemler ve DVM'nin öğrenme kümesi sonuçları GudermannianBoost yönteminden daha iyidir. Ancak bu aşırı öğrenme probleminden kaynaklanmaktadır. Test kümesinde yapılan incelemede GudermannianBoost yönteminin DVM'den daha iyi sonuç verdiği diğer boosting yöntemlerle doğru sınıflandırma oranlarının birbirlerine yakın olduğu görülmektedir. Yöntemin yanlış sınıflandırma ya da aykırılık diye tanımlanabilecek durumları arttıkça diğer boosting yöntemlerden daha iyi sınıf tahmini yapacağı öngörülmektedir. Çünkü diğer boosting yöntemler veriden öğrenmeyi fazla bir şekilde yaparken yanlış sınıflandırma oranının (öğrenme kümesindeki

aykırılıkların) öğrenme kümesinde artması, yöntemleri fazla etkilemektedir. Ancak GudermannianBoost yönteminde doğru sınıflandırmanın ve yanlış sınıflandırmanın cezalandırılmasının ve belli noktadan sonra cezanın sabitlenmesinin etkisi vardır. TanjantBoost yöntemi de aynı özellikleri taşımaktadır ancak ilk uygulamada görülebileceği gibi GudermannianBoost yöntemi tüm benzetim verilerinde test kümesini doğru sınıflandırmada daha iyi sonuçlar vermiştir. Yöntemlerin öğrenme sürecinden test sürecine geçerken doğru sınıflandırma oranındaki düşüşler Şekil 4.2 ve Şekil 4.3'ten görülebilir.



Şekil 4.2. Yöntemlerin Öğrenme ve Test Kümesi Doğru Sınıflandırma Oranları 1



Şekil 4.3. Yöntemlerin Öğrenme ve Test Kümesi Doğru Sınıflandırma Oranları 2

Çizelge 4.9. Gerçek Veri Kümelerinde Öğrenme ve Test Kümeleri için GudermannianBoost Algoritması Doğru Sınıflandırma Bakımından Diğer Boosting Yöntemler ve DVM ile Karşılaştırılması

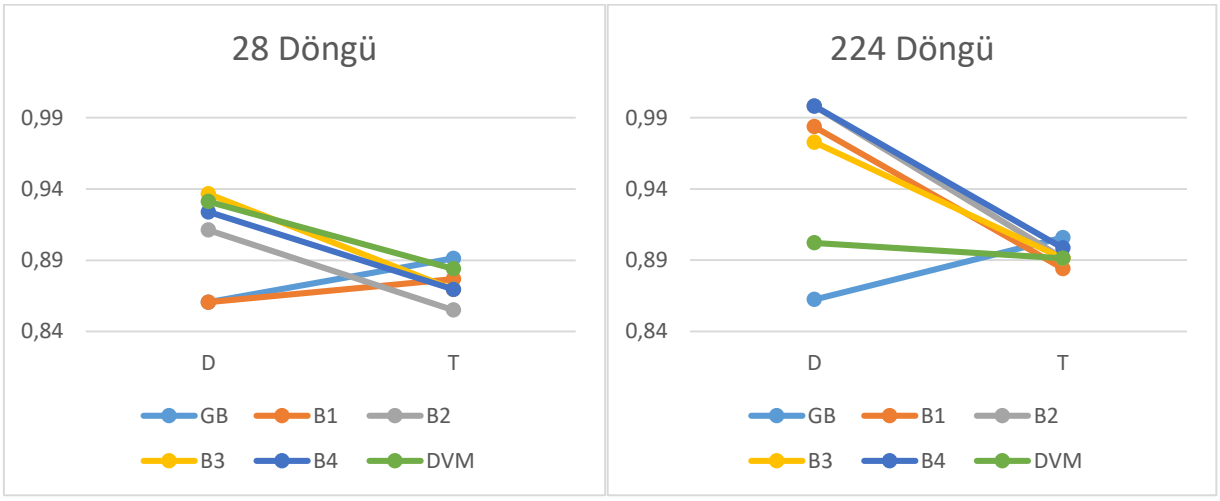
		GB		**B1**		**B2**		**B3**		**B4**		**SVM**	
Veri	Oran	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.
İflas	0,6	0,9999	0,001	0,9998	0,001	0,9998	0,001	1,0000	0,000	1,0000	0,000	1,0000	0,000
Öğrenme	0,7	0,9999	0,001	0,9999	0,001	1,0000	0,000	1,0000	0,000	1,0000	0,000	1,0000	0,000
	0,8	0,9999	0,001	0,9999	0,001	1,0000	0,000	1,0000	0,000	1,0000	0,000	1,0000	0,000
XOX	0,6	0,7135	0,014	0,9862	0,003	0,9882	0,003	0,9874	0,003	0,9890	0,003	0,9570	0,008
Öğrenme	0,7	0,7112	0,011	0,9860	0,003	0,9880	0,003	0,9877	0,003	0,9891	0,003	0,9628	0,006
	0,8	0,7150	0,010	0,9861	0,002	0,9879	0,002	0,9880	0,003	0,9890	0,002	0,9673	0,004
King	0,6	0,9383	0,003	0,9949	0,002	0,9961	0,001	0,9971	0,001	0,9969	0,001	0,9578	0,011
Öğrenme	0,7	0,9386	0,003	0,9954	0,001	0,9964	0,001	0,9972	0,001	0,9972	0,001	0,9625	0,011
	0,8	0,9384	0,002	0,9957	0,001	0,9968	0,001	0,9976	0,001	0,9973	0,001	0,9647	0,010
Kredi	0,6	0,8702	0,010	0,9207	0,010	0,9414	0,009	0,9345	0,010	0,9450	0,009	0,9081	0,010
Öğrenme	0,7	0,8700	0,009	0,9205	0,009	0,9426	0,007	0,9342	0,009	0,9458	0,008	0,9059	0,009
	0,8	0,8687	0,007	0,9212	0,007	0,9438	0,007	0,9349	0,008	0,9467	0,007	0,9041	0,007
İflas	0,4	0,9952	0,006	0,9941	0,008	0,9963	0,007	0,9956	0,008	0,9976	0,007	0,9948	0,006
Test	0,3	0,9956	0,007	0,9935	0,008	0,9954	0,007	0,9945	0,008	0,9968	0,007	0,9956	0,006
	0,2	0,9956	0,007	0,9935	0,008	0,9954	0,007	0,9945	0,008	0,9968	0,007	0,9956	0,006
XOX	0,4	0,6998	0,019	0,9691	0,010	0,9664	0,010	0,9722	0,009	0,9704	0,009	0,8976	0,021
Test	0,3	0,7061	0,025	0,9749	0,008	0,9746	0,008	0,9769	0,007	0,9760	0,007	0,9225	0,018
	0,2	0,7072	0,031	0,9758	0,011	0,9741	0,011	0,9773	0,010	0,9767	0,011	0,9396	0,019
Satranç	0,4	0,9383	0,005	0,9896	0,003	0,9905	0,003	0,9914	0,003	0,9913	0,003	0,9497	0,009
Test	0,3	0,9377	0,007	0,9903	0,003	0,9912	0,003	0,9918	0,003	0,9917	0,003	0,9528	0,009
	0,2	0,9383	0,008	0,9918	0,004	0,9925	0,004	0,9935	0,003	0,9932	0,003	0,9558	0,010
Kredi	0,4	0,8619	0,016	0,8656	0,016	0,8663	0,016	0,8660	0,017	0,8664	0,017	0,8554	0,016
Test	0,3	0,8621	0,022	0,8667	0,021	0,8676	0,021	0,8673	0,021	0,8668	0,021	0,8536	0,022
	0,2	0,8612	0,027	0,8660	0,026	0,8662	0,028	0,8659	0,027	0,8660	0,028	0,8539	0,027

Çizelge 4.10. Avustralya Gerçek Veri Kümesinde Öğrenme ve Test Kümeleri için GudermannianBoost Algoritması Doğru Sınıflandırma Bakımından Diğer Boosting Yöntemler ve DVM ile Karşılaştırılması

			GB		**B1**		**B2**		**B3**		**B4**		**DVM**	
Veri	Oran	A.A	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.	Ortalama	S.S.
Öğrenme	0,6	30	0,8689	0,010	0,9138	0,009	0,9319	0,009	0,9235	0,010	0,9346	0,010	0,9086	0,010
	0,6	40	0,8702	0,010	0,9207	0,010	0,9414	0,009	0,9345	0,010	0,9450	0,009	0,9081	0,010
	0,6	50	0,8709	0,011	0,9274	0,010	0,9495	0,008	0,9432	0,010	0,9542	0,009	0,9075	0,010
	0,7	30	0,8681	0,008	0,9137	0,009	0,9332	0,008	0,9237	0,009	0,9353	0,009	0,9062	0,009
	0,7	40	0,8700	0,009	0,9205	0,009	0,9426	0,007	0,9342	0,009	0,9458	0,008	0,9059	0,009
	0,7	50	0,8711	0,009	0,9283	0,008	0,9500	0,007	0,9439	0,008	0,9546	0,008	0,9054	0,009
	0,8	30	0,8674	0,007	0,9133	0,007	0,9335	0,007	0,9237	0,008	0,9364	0,007	0,9043	0,007
	0,8	40	0,8687	0,007	0,9212	0,007	0,9438	0,007	0,9349	0,008	0,9467	0,007	0,9041	0,007
	0,8	50	0,8699	0,007	0,9283	0,007	0,9516	0,006	0,9451	0,007	0,9551	0,006	0,9044	0,007
	0,9	30	0,8671	0,004	0,9135	0,007	0,9351	0,006	0,9242	0,007	0,9364	0,006	0,9037	0,005
	0,9	40	0,8678	0,005	0,9212	0,007	0,9445	0,006	0,9352	0,007	0,9473	0,006	0,9035	0,004
	0,9	50	0,8688	0,005	0,9280	0,007	0,9519	0,005	0,9448	0,006	0,9554	0,005	0,9033	0,005
Test	0,4	30	0,8604	0,017	0,8659	0,016	0,8661	0,016	0,8658	0,016	0,8657	0,016	0,8545	0,017
	0,4	40	0,8619	0,016	0,8656	0,016	0,8663	0,016	0,8660	0,017	0,8664	0,017	0,8554	0,016
	0,4	50	0,8630	0,018	0,8675	0,017	0,8675	0,017	0,8678	0,017	0,8679	0,017	0,8556	0,017
	0,3	30	0,8601	0,020	0,8658	0,020	0,8658	0,021	0,8664	0,021	0,8663	0,020	0,8538	0,020
	0,3	40	0,8621	0,022	0,8667	0,021	0,8676	0,021	0,8673	0,021	0,8668	0,021	0,8536	0,022
	0,3	50	0,8613	0,021	0,8657	0,021	0,8661	0,020	0,8657	0,021	0,8659	0,021	0,8546	0,021
	0,2	30	0,8595	0,027	0,8650	0,026	0,8655	0,026	0,8662	0,025	0,8658	0,026	0,8521	0,027
	0,2	40	0,8612	0,027	0,8660	0,026	0,8662	0,028	0,8659	0,027	0,8660	0,028	0,8539	0,027
	0,2	50	0,8602	0,027	0,8660	0,025	0,8672	0,026	0,8671	0,025	0,8674	0,026	0,8532	0,026
	0,1	30	0,8596	0,041	0,8662	0,037	0,8661	0,038	0,8654	0,037	0,8674	0,038	0,8522	0,040
	0,1	40	0,8609	0,040	0,8668	0,039	0,8679	0,040	0,8673	0,039	0,8682	0,039	0,8525	0,040
	0,1	50	0,8643	0,038	0,8676	0,040	0,8706	0,038	0,8684	0,038	0,8688	0,038	0,8550	0,039

Diğer boosting yöntemler, her durumda test kümesine bir düşüş ile geçmektedir. Diğer taraftan GudermannianBoost için düşüş gerçekleşmemektedir. Bu durum iki açıdan incelenmelidir. Boosting yöntemlerde aşırı öğrenme sorununun engellenmesi için GudermannianBoost, LojitBoost gibi regresyon tahmin temelli yöntemler kullanılabilir. Veri kümeleri çok karmaşık yapılarda olduğunda, GudermannianBoost çekirdek fonksiyonlar ile birlikte kullanılırsa veya algoritmanın adım sayısı arttırılırsa, yöntem daha başarılı hale getirilebilir. Diğer boosting algoritmalardaki düşüşler incelendiğinde, üstel ve lojistik kayıplı GentleBoost, RealBoost algoritmalarına göre daha fazla düşüş kaydetmektedir. İki yöntemden RealBoost'un aykırı değerlere karşı daha duyarsız olduğu söylenebilir.

Algoritmadaki döngü sayısı değişken sayısının katına göre belirlenirse, GudermannianBoost yöntemi daha başarılı sınıflandırma yapabilmektedir. Yine Avustralya gerçek kredi verisinde 14 bağımsız değişkenin olduğu durum incelendiğinde sırasıyla 14, 28, 42, 224 ve 238 algoritma tekrarı yapılmıştır. Bu şart altında, Çizelge 4.11'de Avustralya verisine ait farklı döngü sayılarında GudermannianBoost yönteminin test kümelerinde daha başarılı olduğu görülmüştür. Diğer boosting yöntemler öğrenme kümesinden test kümesine geçişte doğru sınıflandırma oranlarında düşüş yaşamaktadır. Ancak bu durum hem az döngü hem de çok döngü sayısının bulunduğu durumlarda GudermannianBoost için geçerli değildir. Başarı oranları Çizelge 4.11'de ve Şekil 4.4'te verilmiştir.



Şekil 4.4. Döngü Sayısı Öğrenme ve Test Kümesi Doğru Sınıflandırma Oranları

Çizelge 4.11. Avustralya Gerçek Veri Kümesinde Değişken Sayısı Katında Döngü Sayısına Göre Doğru Sınıflandırma Bakımından Diğer Boosting Yöntemler ve DVM ile Karşılaştırılması

Veri	Oran	A.A.	GB	B1	B2	B3	B4	DVM
Öğrenme	0,8	14	0,849638	0,849638	0,896739	0,905797	0,914855	0,905797
	0,8	28	0,860507	0,860507	0,911232	0,936594	0,923913	0,931159
	0,8	42	0,871377	0,913044	0,951087	0,931159	0,947464	0,905797
	0,8	224	0,862319	0,983696	0,998188	0,972826	0,998188	0,902174
	0,8	238	0,855073	0,987319	0,998188	0,985507	0,998188	0,902174
Test	0,2	14	0,884058	0,876812	0,891304	0,876812	0,884058	0,927536
	0,2	28	0,891304	0,876812	0,855073	0,869565	0,869565	0,884058
	0,2	42	0,847826	0,847826	0,847826	0,84058	0,833333	0,833333
	0,2	224	0,905797	0,884058	0,891304	0,891304	0,898551	0,891304
	0,2	238	0,884058	0,869565	0,862319	0,869565	0,869565	0,869565

Öğrenme ve test kümelerindeki geçişlerde algoritma adım sayısı değişken sayısı kadar olduğunda DVM ve üstel kayıp fonksiyonlu GentleBoost yöntemi GudermannianBoost yönteminden daha iyi sonuç vermiştir. Adım sayısı değişken sayısının iki katına çıkartıldığında GudermannianBoost yöntemi diğer yöntemlerden daha iyi sonuç vermektedir. Diğer tüm algoritma adımlarındaki sonuçlar GudermannianBoost lehinedir.

5. SONUÇ ve TARTIŞMA

İstatistiksel sınıflandırma değişken özelliklerine göre sınıf etiketini belirleyebilme sürecidir. Girdiler ve çıktılar üzerinden modellerin kurulduğu aşama, öğrenme sürecidir ve süreç sonunda elde edilen model ile istatistiksel tahmin elde edilebilir. Girdi verilerinde aykırılıkların ve yanlış sınıflandırmaların olması öğrenme süreci için problemdir. Yanlış modeller ve yanlış tahminler elde edilmesi modelin geçerliliğini ve gözleme göre modelin kararlılığını etkilemektedir. Bu noktadan hareketle sağlam sınıflandırma yöntemleri ile öğrenme sürecindeki aykırılıklara karşı duyarsız modeller elde edilmeye çalışılmaktadır.

Bu çalışmada istatistiksel öğrenme ve sağlam ayrım tabanlı bazı çalışmalar açıklanmıştır. Sınıflandırma yöntemlerinde kullanılan bazı kayıp fonksiyonlar verilmiştir. Klasik ve sağlam sınıflandırma yöntemlerinde kullanılan, istatistiksel özelliğe sahip olan ve metodolojinin içinde sıklıkla kullanılan kayıp fonksiyonlar yapıları itibariyle yüzeysel olarak incelenmiştir. Gudermannian kayıp fonksiyonu olarak isimlendirilen ve Masnadi-Shirazi [26] çalışmasında belirtilen özellikleri taşıyan fonksiyon için regresyon tabanlı boosting algoritması, LojitBoost'a benzer algoritma, GudermannianBoost önerilmiştir. Çalışmada son olarak benzetim ve gerçek veri kümeleri ile uygulamalar yapılmış, GudermannianBoost'un etkin olduğu noktalar açıklanmıştır.

LojitBoost gibi klasik ve regresyon tabanlı yöntemlerde öğrenme ve test kümesindeki doğru sınıflandırma oranları birbirine yakındır. Ancak diğer boosting yöntemlerde aşırı öğrenme problemi, test kümesindeki doğruluk oranını öğrenme kümesindeki doğruluk oranı kadar elde edememektedir. Yöntemlerdeki algoritma sayısı arttırıldıkça öğrenme kümesindeki aykırılıklar sınıflandırıcıyı çok fazla etkilemeye başlamaktadır. GudermannianBoost yöntemi ise öğrenme kümesindeki ayrımları net bir şekilde ifade etmeye çalışmaktadır. Bu durumu regresyon tabanlı tahminleri kullanarak sınıfa ait olma olasılıklarındaki hatayı minimize ederek yapmaya çalışmaktadır. Sınıfa ait olma olasılıkları, yanlış sınıflandırmaların etkisinden arındırılmaya çalışılmaktadır. Bu süreç ise Gudermannian kayıp fonksiyonu ve karesel risk fonksiyonu kullanılarak elde edilmektedir. Uygulamada yapılan sınıflandırma çalışmaları sonucunda, GudermannianBoost yönteminin öğrenme kümesindeki aşırı öğrenmeyi engellediği görülmüştür. Öğrenme kümesindeki bu süreç, tahminin ve test kümesindeki çıktıların sonucunda iyileştirmeler getirmektedir. GudermannianBoost yöntemi, TanjantBoost sürecini

örnek alan ancak sınıflandırıcıya yakın aykırılıkların etkisini TanjantBoost yönteminden daha çabuk arındıran bir yöntemdir. Bu durum benzetim çalışmasıyla gösterilmiştir. Önerilen yöntemin dezavantajı ise uygulamada da açıklandığı gibi, zor ayrılabilir veri kümelerinde doğru sınıflandırılmış gözlemleri yanlış sınıflandırılmış olarak öğrenebilir. Bu durumu engellemek için DVM'de de öneri olarak getirilmiş olan çekirdek fonksiyon kullanımı bir çözümdür. Ayrılabilir hale getirecek şekilde farklı bir uzayda veriyi tanımladıktan sonra GudermannianBoost yönteminin uygulanması daha başarılı sonuçlar getirecektir.

Bu çalışmadan sonra yapılabilecekler, yöntemi çoklu sınıf için genelleştirmek, çekirdek fonksiyon yardımıyla yöntemi daha fazla doğru sınıflandırma oranı verebilir hale getirmek, karar ağacı çizimi ile görselleştirilebilmek, algoritma adımları yardımıyla aykırı değerleri belirlemektir.

KAYNAKLAR

[1] Vapnik, V., An Overview Of Statistical Learning Theory, *IEEE Transactions On Neural Networks*, 10, 5, pp. 988-1000, **1999**.

[2] Wu, Y., Liu, Y., Robust Truncated-Hinge-Loss Support Vector, *Journal of the American Statistical Association*, Vol. 102, No. 479, **2007**.

[3] Xu, H., Caramanis, C., Mannor, S., Robustness and Regularization of Support Vector Machines, *Journal of Machine Learning Research*, Vol. 10, pp. 1485-1510, **2009**.

[4] Carrizosa, E., Morales, D. R., Supervised Clasification and Mathematical Optimization, *Computers and Operations Research*, 40, 150-165, **2013**.

[5] Park, S. Y., Flexible Margin-Based Classification Techniques, *Phd Dissertation, Chapel Hill*, 114 p., **2010**.

[6] Gupta, S., Robust Margin Based Classifiers For Small Sample Data, *Arizona State Unversity*, *Phd Dissertation*, 47 p, **2011**.

[7] Ma, Y., Li, L., Huang, X., Wang, S., Robust Support Vector Machine Using Least Median Loss Penalty, *Proceedings of the 18th IFAC World Congress,* **2011**.

[8] Song, Q., Hu, W., Xie, W., Robust Support Vector Machine With Bullet Hole İmage Classification, *Transactions on Systems, Man and Cybernetics, Part C*, 32(4), 440–448, **2002**.

[9] Ertekin, S., Bottou, L., Giles, C., 2010, Nonconvex Online Support Vector Machines, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2), 368–381, **2010**.

[10] Liu, Y., Shen, X., Doss, H., Multicategory Ψ-Learning And Support Vector Machine, *Journal of Computational and Graphical Statistics*, 14(1), 219–236, **2005**.

[11] Debruyne, M., Robust Support Vector Machine Classification, *Leuven Statistical Day*, **2008**.

[12] Mai, Q., Variable Selection in High-Dimensional Classification, *University of Minnesota, Phd. Dissertation, 115p*, **2013**.

[13] Buckstein, G. M., Sparsity Control for Robustness and Social Data Analysis, Classification, *University of Minnesota, Phd. Dissertation, 138p*, **2012**.

[14] Lim, N., Classification by Ensembles from Random Partitions Using Logistic Regression Models, *Stony Brook University, Phd. Dissertation, 85* p, **2007**.

[15] Oh, D.Y, GA-BOOST: A Genetic Algorithm for Robust Boosting, *The University of Alabama, Phd. Dissertation*, 147 p, **2012**.

[16] Pochet, N., Smet, F.D., Suykens J. A.N. De Moor, B. L. R., Systematic Benchmarking Of Microarray Data Classification: Assessing The Role Of Non-Linearity And Dimensionality Reduction, *Bioinformatics,* Vol. 20 no. 17, 3185–3195, **2004**.

[17] Raudys, S. J., Jain, A. K., Small Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol 13, No: 3, 252-264, **1991**.

[18] Roweis, S. T., Saul, L. K., Nonlinear dimensionality reduction by locally linear embedding, *Science 22*, Vol. 290, Mo. 5500, 2323-2326, **2000**.

[19] Ren, C.X., Dai, D.Q, Robust classification using l2,1-norm based regression model, *Pattern Recognition*, 45, 2708–2718, **2012**.

[20] Chi, E. C., Parametric Classification and Variable Selection by the Minimum Integrated Squared Error Criterion, *Rice University, Phd. Dissertation*, 98 p, **2011**.

[21] Nudurupati, S. V., Robust Nonparametric Discriminant Analysis Procedures, *Auburn University, Phd. Dissertation,* 132 p, **2009**.

[22] Freund, Y., R. E. Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, *Journal of Computer and System Sciences,* Vol. 55, pp. 119–139, **1997**.

[23] Friedman, J., T. Hastie, R. Tibshirani, Additive logistic regression: A statistical view of boosting, *Annals of Statistics,* Vol. 28, No. 2, pp. 337–407, **2000**.

[24] Freund, Y., A more robust boosting algorithm. *arXiv:0905.2138v1,***2009**.

[25] Freund, Y., An adaptive version of the boost by majority algorithm. *Machine Learning*, 43(3):293-318, **2001**.

[26] Masnadi-Shirazi, H., The Design Of Bayes Consistent Loss Functions For Classification, *University of California, Phd Dissertation, San Dieago,* 223 p, **2011**.

[27] Kobetski, M., Sullivan. J., Improved Boosting Performance by Exclusion of Ambiguous Positive Examples. *In ICPRAM,* **2013**.

[28] Steinwart, I., Christmann, A., Support Vector Machines, *Springer, New York*, 600 pg, **2008**.

[29] Zhao, L., Mammadov, M., Yearwood, J., From Convex to Nonconvex: A Loss Function Analysis for Binary Classification, *IEEE International Conference on Data Mining Workshops (ICDMW)*, pp.1281,1288, **2010**.

[30] Huang, X., Shi, L., Suykens, J. A., Ramp Loss Linear Programming Support Vector Machine, *Journal of Machine Learning Research*, 15, 2185-2211, **2014**.

[31] Rennie, J. D., Smooth Hinge Classification. *Proceeding of Massachusetts Institute of Technology*, **2005**.

[32] Zhang, T., Statistical behavior and consistency of classification methods based on convex risk minimization, *Annals of Statistics*, 56-85, **2004**.

[33] Li, F., Yang, Y., A Loss Function Analysis For Classification Methods İn Text Categorization, *In Machine Learning-International Workshop Then Conference,* volume 20, page 472, **2003**.

[34] Shen, X., Tseng, G. C., Zhang, X., Wong W. H., On [Psi]-Learning, *Journal of the American Statistical Association, 98*(463):724–735, **2003**.

[35] Mason L., Bartlett, P.L., Baxter, J., Improved Generalization Through Explicit Optimization Of Margins, *Machine Learning*, 38(3):243–25, **2000**.

[36] Chapella, O., Training a Support Vector Machine in the Primal, *Neural Computation*, 19, 5 (May 2007), 1155-1178, **2007**.

[37] Masnadi-Shirazi, H., Vasconcelos, N., On The Desing Of Loss Funcitons For Classificaiton: Theory, Robustness To Outliers And Savageboost, *Advances in Neural Information Processing Systems 21*, pp. 1049-1056, **2009**.

[38] Masnadi-Shirazi,H., Vasconcelos, N., Mahadevan, V., On the Design of Robust Classifiers for Computer Vision, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2010**.

[39] Schapire, R. E., The Strenght of Weak Learnability, *Machine Learning*, 5, p. 197-227, **1990**.

[40] J. Wang, Yang Y., Boosted classification of breast cancer by retrieval of cases having similar disease likelihood, *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*,908-911, **2016**.

[41] Dubossarsky, E., Friedman, J.H., Ormerod, J.T., Wand, P.M., Wavelet-Based Gradient Boosting, *Stat Computation*, 26: 93, **2016**.

[42] A. Sen, Islam, M. M., Murase K., Yao, X., Binarization with Boosting and Oversampling for Multiclass Classification, *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1078-1091, **2016**.

[43] S. Pan, Wu, J., Zhu, X., CogBoost: Boosting for Fast Cost-Sensitive Graph Classification, *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 2933-2946, **2015**.

[44] D. Mund, Triebel R., Cremers, D., Active online confidence boosting for efficient object classification, *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, 2015, pp. 1367-1373, **2015**.

[45] Zeng, M., Yang, Y., Zheng, J., Cheng, J., Maximum margin classification based on flexible convex hulls. *Neurocomputing*, 149, 957-965, **2015**.

[46] Nie, Q., Jin, L., Fei, S. Probability Estimation For Multi-Class Classification Using Adaboost. Pattern Recognition, 47(12), 3931-3940, **2014**.

[47] De Menezes, F. S., Liska, G. R., Cirillo, M. A., Vivanco, M. J., Data Classification With Binary Response Through The Boosting Algorithm and Logistic Regression, *Expert Systems with Applications*, 69, 62-73, **2017**.

[48] Kim, M. J., Kang, D. K., Kim, H. B., Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction, Expert Systems with Applications, 42(3), 1074-1082, **2015**.

[49] Zhai, J., Zhang, S., Wang, C., The Classification of Imbalanced Large Data Sets Based on MapReduce and Ensemble pf ELM Classifiers, *International Journal of Machine Learning & Cyber*, **2015**.

[50] Lin, S., Wang, Y., Xu, L., Re-Scale Boosting for Regression and Classification, *arxiv.org*, **2015**.

[51] Wang, B., Pineau, J. Online Bagging and Boosting for Imbalanced Data Streams. *IEEE Transactions on Knowledge and Data Engineering*, 28(12), 3353, **2016**.

[52] Nikolaou, N., Edakunni, N., Kull, M., Flach, P., Brown, G. Costsensitive Boosting Algorithms: Do We Really Need Them?, *Machine Learning*, 104(2), 359-384, **2016.**

[53] Ghosal, S., Turnbull, B., Zhang, H. H., Hwang, W. Y., Sparse penalized forward selection for support vector classification. *Journal of Computational and Graphical Statistics*, 25(2), 493-514,**2016.**

[54] Martinez, W., Gray, J. B., Noise peeling methods to improve boosting algorithms. Computational Statistics & Data Analysis, 93, 483-497, **2016**.

[55] Conroy, B., Eshelman, L., Potes, C, Xu-Wilson, M., A Dynamic Ensemble Approach to Robust Classification in the presence of Missing Data, *Machine Learning*, 102: 443,**2015**.

[56] Xiong, W., Zhang, L., Du, B., Tao, D., Combining local and global: Rich and robust feature pooling for visual recognition, *Pattern Recognition*, 62, 225-235, **2017**.

[57] Tempel, S., Zerath, B., Zehraoui, F., Tahi, F. miRBoost: boosting support vector machines for microRNA precursor classification, *RNA*, 21(5), 775-785, **2015**.

[58] Li, G., Wang, S, Oversampling boosting for classification of imbalanced software defect data, *In Control Conference (CCC)*, 2016 35th Chinese (pp. 4149-4154). TCCT, **2016**.

[59] Pan, S., Wu, J., Zhu, X, Long, D., Zhang, C., Boosting for Graph Classification with Universum, *Knowledge of Inference System*, **2016**.

[60] Momparler, A., Carmona, P.,Climent, F., Banking Failure Prediction: A Boosting Classification Tree Approach, *Spanish Journal of Finance and Accounting/Revista Española de Financiación y Contabilidad*, 45(1), 63-91, **2016**.

[61] Savage, L. J., The Elicitation of Personal Probabilities and Expectations, *Journal of American Statistical Association*, 66:783-801, **1971**.

[62] Tuszynski, J., R packages: *Package 'caTools':* LogitBoost, **2015.**

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

[64] Matheus, C.J., Rendell, L.A., Constructive induction on decision trees. *In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence.* pp. 645-650. Detroit, MI: Morgan Kaufmann, **1989**.

[65] Shapiro, A. D., Structured Induction in Expert Systems Addison, Wesley. *This book is based on Shapiro's Ph.D. thesis at the University of Edinburgh entitled "The Role of Structured Induction in Expert Systems"*, **1987**.

[66] Asuncion, A., Newman, D.J., UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, *CA: University of California, School of Information and Computer Science*, **2007**.

[67] Quinlan, R., Simplifying decision trees, *International Journal Man-Machine Studies 27*, pp. 221-234, **1987**.

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı	: Onur TOKA
Doğum Yeri	: Ankara
Medeni Hali	: Evli
E-posta	: onur.toka@hacettepe.edu.tr
Adres	: Hacettepe Üniversitesi Beytepe Yerleşkesi Fen Fakültesi İstatistik Bölümü, 06800, Çankaya/ANKARA

Eğitim

Lise	: 2000-2004 Sincan (Yabancı Dil Ağırlıklı- YDA) Lisesi
Lisans	: 2004-2009 Hacettepe Üniversitesi, İstatistik Bölümü
	: 2006-2011 Anadolu Üniversitesi, İşletme Bölümü
Yüksek Lisans	: 2009-2012 Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, İstatistik Anabilim Dalı
Doktora	: 2012-2016 Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, İstatistik Anabilim Dalı

Yabancı Dil ve Düzeyi

İngilizce : 75 (KPDS)

İş Deneyimi

Araştırma Görevlisi, Hacettepe Üniversitesi İstatistik Bölümü, (2009- Devam Ediyor)

Teknik Uzman, Türk Akreditasyon Kurumu (TÜRKAK), (2014- Devam Ediyor)

Deneyim Alanları

Sınıflandırma, Veri Madenciliği, Değişken Seçimi, Regresyon Modelleri

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Yayınlar

-

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

TOKA, Onur, Meral Çetin, "Loss Functions in Classification: A Comparative Study", 2nd International Researchers, Statisticians and Young Statisticians Congress (IRSYSC2016), ANKARA, May 4-8, 2016.

TOKA, Onur, Meral Çetin, "Robust Properties of Some Loss Functions in Classification", International Conference on Information Complexity and Statistical Modeling in High Dimensions with, NEVŞEHİR, May 18-21, 2016.

TOKA, Onur, Meral Çetin, "Robust Boosting Algorithm: GudermannianBoost", Applied Statistics 2016: International Conference, Ribno(Bled), SLOVENYA, September 18-21, 2016.

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 16/12/2016

Tez Başlığı : Gudermannian Kayıp Fonksiyonu ve GudermannianBoost İlgili Sınıflandırma Yöntemi

Yukarıda başlığı gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç ve e)Kaynakça kısımlarından oluşan toplam 66 sayfalık kısmına ilişkin, 16/12/2016 tarihinde şahsım tarafından *Turnitin* adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 3'tür.

Uygulanan filtrelemeler:
1- Kaynakça hariç
2- Alıntılar dâhil
3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orjinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

16.12.2016
Tarih ve İmza

Adı Soyadı: ONUR TOKA
Öğrenci No: N12143691
Anabilim Dalı: İSTATİSTİK
Programı: İSTATİSTİK DOKTORA PROGRAMI
Statüsü: ☐ Y.Lisans ☒ Doktora ☐ Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

Prof. Dr. Meral ÇETİN
(Unvan, Ad Soyad, İmza)