T. C.
BAHCEŞEHİR ÜNİVERSİTESİ

MAKİNE ÖĞRENMESİ ALGORİTMALARI VE ANOMALİ TESPİTİ

Yüksek Lisans Tezi

MEHMET DENİZ TOPUZ

İSTANBUL, 2014

T. C.
BAHCEŞEHİR ÜNİVERSİTESİ
FEN BİLİMLERİ
UYGULAMALI MATEMATİK

MAKİNE ÖĞRENMESİ ALGORİTMALARI VE ANOMALİ TESPİTİ

Yüksek Lisans Tezi

MEHMET DENİZ TOPUZ

Supervisor: Doç. Dr. Atabey Kaygun

İSTANBUL, 2014

T. C.
BAHCEŞEHİR ÜNİVERSİTESİ
FEN BİLİMLERİ
UYGULAMALI MATEMATİK

Title of the Master's Thesis	:	MAKİNE ÖĞRENMESİ ALGORİTMALARI VE ANOMALİ TESPİTİ
Name/Last Name of the Student	:	MEHMET DENİZ TOPUZ
Date of Thesis Defense	:	13 JULY 2014

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assoc. Prof. Dr. F. Tunç BOZBURA
Acting Director

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Commitee Members:

Doç. Dr. Atabey Kaygun (Supervisor)	:
Prof. Dr. İrini DİMİTRİYADİS	:
Yrd .Doç. Dr. Fatih ECEVİT	:

ÖNSÖZ

Danışmanım Doç. Dr. Atabey Kaygun'a çalışma isteğime ve eğitimime olan büyük katkısı için çok teşekkür ederim. Prof. Dr. İrini Dimitriyadis'e doğru araştırma konusunu seçme sürecindeki yardımları için ve sürekli motivasyonumu artırdığı için çok teşekkür ederim. Yrd. Doç.Dr. Süreyya Akyüz'e ve Yrd. Doç.Dr. Fatih Ecevit'e bana sürekli yol göstermeleri, tez jürimde bulunmaları ve vizyonumu genişletmeleri sebebiyle çok teşekkür ederim. Haftalık seminerlere gelip çalıştığımız konuları birlikte ele alabilmemizi sağlayan değerli arkadaşlarıma teşekkür ederim. Ayrıca bölümümüzde bulunan tüm hocalarımıza olumlu yaklaşımları ve destekleyici tutumları için çok minnettarım.

13 TEMMUZ 2014 MEHMET DENİZ TOPUZ

ABSTRACT

MACHINE LEARNING ALGORITHMS AND ANOMALY DETECTION

TOPUZ, MEHMET DENİZ

APPLIED MATHEMATICS
Supervisor: Doç. Dr. Atabey Kaygun

JULY 2014, 63 Pages

Machine learning is the subfield of the artifical intelligence which finds the significant behaviours or functions from the data for future predictions. Huge amount of data were collected in the last decades and analysis of such a big data requires intelligent systems. Machine learning enables a computer to learn from example data or past experience. According to their learning style, machine learning algorithms can be categorized into two groups: supervised learning algorithms and unsupervised learning algorithms. Training data of supervised learning algorithms includes both the inputs and labels. Unsupervised learning model is not provided with the correct labels during training. A detailed explanation of leading machine learning algorithms is offered in the first part of this thesis.

Anomaly is a pattern in the data that does not conform to expected behaviour. Existence of anomalies in the data is important because they might translate to critical actionable information. Both supervised and unsupervised machine learning techniques are applied to detect anomalies in different domains. Last part of this thesis provides an overview of the relation between anomaly detection problem and machine learning approaches.

Keywords: Supervised and Unsupervised Learning, Learning Algorithms, Anomaly Detection

ÖZET

MAKİNE ÖĞRENMESİ ALGORİTMALARI VE ANOMALİ TESPİTİ

TOPUZ, MEHMET DENİZ

UYGULAMALI MATEMATiK
Tez Danışmanı: Doç. Dr. Atabey KAYGUN

TEMMUZ 2014, 63 Sayfa

Makine öğrenmesi yapay zekanın bir alt çalışma alanıdır ve veriden önemli davranışlar ve kurallar çıkartarak ileriye doğru tahminler yapabilmemizi sağlar. Son 20 yılda değişik çalışma alanlarındaki veri miktarı çok hızlı artmıştır ve bu verinin insan çalışması ile analiz edilmesi zordur. Makine öğrenmesi algoritmalarına dair temelde iki öğrenme şekli vardır : gözeticili öğrenme ve gözeticisiz öğrenme. Gözeticili öğrenmede data önceden bilinen sınıflara ayrılır. Gözeticisiz öğrenme de ise sınıflar önceden bilinmez, öğrenme algoritması veri içindeki ayrık yapıları kendisi keşfeder. Bu tezde çok kullanılan makine öğrenmesi algoritmaları detayları ile açıklanmıştır.

Veri kümesi içinde beklenen davranışları doğrulamayan örüntülere anomali denir. Veri kümesi içinde anomali bulunmasının önemli sonuçları olabilir. Tezin son bölümünde önceki kısımda bahsedilen makine öğrenmesi algoritmalarının ve yaklaşımlarının anomali tespit etme problemine nasıl uyarlandığı açıklanmıştır.

Anahtar Kelimeler: Gözeticili Öğrenme, Gözeticisiz Öğrenme, Makine Öğrenmesi Algoritmaları, Anomali Tespiti

İÇİNDEKİLER

TABLOLAR

ŞEKİLLER

KISALTMALAR

Destek Vektör Makineleri	:	DVM (SVM)
En Yakın k Komşuluk Algoritması	:	k-NN
Naif Bayes Sınıflandırıcı	:	NB
Lineer Ayırtaç Analizi	:	LAA (LDA)
Temel Bileşen Analizi	:	TBA (PCA)
Özdeğer Ayrışımı	:	ÖA (EVD)
Tekil Değer Ayrışımı	:	TDA (SVD)
Yerel Anomali Faktörü	:	YAF (LOF)
Bağlantısallık Temelli Anomali Faktörü	:	BTAF (COF)
Çok Tanecikli Sapma Faktörü	:	ÇTSF (MDEF)
En Büyük Olabilirlik	:	EB0 (MLE)

SEMBOLLER

Anomalilerin dağılımı	:	$\mathbf{A}$
Ağırlık terimi	:	w
Girdi	:	x_i
Giriş verilerine uyan çıkış etiketleri	:	y_i
Yanlılık terimi	:	b
Lagrange fonksiyonu	:	$L(\mathbf{w}, \mathbf{b}, \alpha)$
Lagrange çarpanları	:	α_i
Serbestlik değişkeni	:	ξ
Taşıma fonksiyonu	:	Φ
Kernel fonksiyonu	:	$k()$
Real Sayılar kümesi	:	R
n boyutlu reel vektör uzayı	:	R^n
X matrisinin transpozu	:	X^T
Uzaklık fonksiyonu	:	$d_{i,j}$
X matrisinin tersi	:	X^{-1}
Parametre vektörü	:	θ
Sigmoid Fonksiyonu	:	$g(x)$
Düzenlileştirme Çarpanı	:	λ
İstenen grup sayısı	:	k
İzdüşüm matrisi	:	W
Sınıf içi saçılım	:	S_W
Sınıflar arası saçılım	:	S_B
Merkez Vektörü	:	c
Eşdeğişinti matrisi	:	C
Özvektörler matrisi	:	S
Özdeğerler matrisi	:	Λ
Diagonal matris	:	Σ
Karmaşıklık Fonksiyonu	:	$C(.)$
Yoğunluk fonksiyonu	:	$f(x, \theta)$
Ortalama	:	μ
Standard sapma	:	σ
ki-kare	:	χ^2
Normal noktaların dağılımı	:	$\mathbf{M}$
Geçmiş Deneyim	:	E
Performans Ölçütü	:	P
Yeni Görev	:	T
Hipotez	:	$h_\theta(x)$
Teta vektörünün transpozu	:	θ^T
Maliyet Fonksiyonu	:	$J(\theta)$
Giriş Verilerine Uyan Çıkış Etiketleri	:	y_i

SEMBOLLER

Anomalilerin dağılımı	:	$\mathbf{A}$
Ağırlık terimi	:	w
Girdi	:	x_i
Giriş verilerine uyan çıkış etiketleri	:	y_i
Yanlılık terimi	:	b
Lagrange fonksiyonu	:	$L(\mathbf{w}, \mathbf{b}, \alpha)$
Lagrange çarpanları	:	α_i
Serbestlik değişkeni	:	ξ
Taşıma fonksiyonu	:	Φ
Kernel fonksiyonu	:	$k()$
Real Sayılar kümesi	:	R
n boyutlu reel vektör uzayı	:	R^n
X matrisinin transpozu	:	X^T
Uzaklık fonksiyonu	:	$d_{i,j}$
X matrisinin tersi	:	X^{-1}
Parametre vektörü	:	θ
Sigmoid Fonksiyonu	:	$g(x)$
Düzenlileştirme Çarpanı	:	λ
İstenen grup sayısı	:	k
İzdüşüm matrisi	:	W
Sınıf içi saçılım	:	S_W
Sınıflar arası saçılım	:	S_B
Merkez Vektörü	:	c
Eşdeğişinti matrisi	:	C
Özvektörler matrisi	:	S
Özdeğerler matrisi	:	Λ
Diagonal matris	:	Σ
Karmaşıklık Fonksiyonu	:	$C(.)$
Yoğunluk fonksiyonu	:	$f(x, \theta)$
Ortalama	:	μ
Standard sapma	:	σ
ki-kare	:	χ^2
Normal noktaların dağılımı	:	$\mathbf{M}$
Geçmiş Deneyim	:	E
Performans Ölçütü	:	P
Yeni Görev	:	T
Hipotez	:	$h_\theta(x)$
Teta vektörünün transpozu	:	θ^T
Maliyet Fonksiyonu	:	$J(\theta)$
Giriş Verilerine Uyan Çıkış Etiketleri	:	y_i

1. GİRİŞ

Bu tezin ilk bölümünde temel makine öğrenmesi algoritmaları ve bu algoritmalara dair örnekler sunulacaktır.

1.1 Tanım

Makine öğrenmesi yapay zekanın bir alt çalışma alanıdır ve veriden önemli davranışlar ve kurallar çıkartarak ileriye doğru tahminler yapabilmemizi sağlar. Endüstride ve bilimsel çalışmalarda sıklıkla kullanılır. Makine öğrenmesi algoritmaları kendi kendini modifiye eden proğramlar için kullanılır. Çalıştırılmak istenen uygulama zaman içinde değişiyorsa veya farklı durumlara göre özelleşmesi gerekiyorsa, çevresine uyum gösterebilen genel dizgeler (sistemler) oluşturmak kullanışlı olur [Alpaydin (2004)].

Makine öğrenmesinin kesin ve standart bir tanımı yoktur. Arthur Samuel'in tanımına göre makine öğrenmesi bilgisayarları açıkça proğramlamadan onlara öğrenme yeteneği kazandıran çalışma alanıdır [Simon (2013)].

Tom Mitchell makine öğrenmesinin daha formal bir tanımını şu makine öğrenmesi problemini tanımlayarak vermiştir: Bir bilgisayar proğramının yeni bir görev olan T yi gerçekleştirmesi istenmektedir. Bu bilgisayar proğroğramının, yeni görev T yi yaparken geçmiş deneyimlerden (E) öğrenerek performansını (P) artırmasına makine öğrenmesi denir [Mitchell (1997)]. Ethem Alpaydın makine öğrenmesini bilgisayarların örnek veri yada geçmiş deneyimi kullanarak bir ölçüte göre başarılarını artıracak biçimde proğramlaması olarak tanımlamıştır [Alpaydin (2004)]. Makine öğrenmesinde amaç geçmiş deneyimlerden öğrenmek ve yeni gelecek örnekler için genelleme yapmaktır [Bishop et al. (2006)].

Makine öğrenmesi, veri madenciliği, istatistik, veri analizi, yapay zeka, bioinformatik ve matematik güçlü şekilde birbiriyle ilişkili alanlardır. İstenmeyen e-postaların tanınması (Spam detection), dolandırıcılığın ortaya çıkarılması (fraud detection), basamak tanıma(digit recognition), konuşma tanıma ve işleme (natural language processing), yüz tanıma(face detection), ürün tavsiye etme (product recommendation), medikal teşhis (medical diagnosis), stok ticareti (stock trading), müsteri bölümleme (customer segmentation), şekil

tanıma (shape detection) makine öğrenmesi algoritmalarının meşhur uygulamalarıdır. Aslında makine öğrenmesinin tanımı çözülen problemin yapısına ve ortamına bağlıdır.

Makine öğrenmesi tekniklerinin uygulanabilmesi için öncelikle verinin temizlenmesi ve ön işlemeden geçirilmesi gerekir. İlk önümüze gelen işlenmemiş ham veri genellikle tam değildir, gürültülüdür ve de tutarsızlıklar (farklılıklar) içerir, neticede analizde kullanılması zor bir veridir. Orjinal verinin temsili bir alt kümesini almak (sampling), anomalileri tespit etmek, gürültüyü ayıklamak, verinin eksik ya da kayıp kısmını telafi ve tamir için yollar bulmak, tutarsızlıkları ortadan kaldırmak, normalize etmek ve bazı gereksiz öznitellikleri (feature) çıkartmak sıklıkla uygulanan ön işlemlerdir. Bu ön işlemlerden sonra algoritmaların uygulanabileceği bir veri elde ederiz.

1.1.1 Gözeticili ve Gözeticisiz Öğrenme

Bu tez en çok kullanılan makine öğrenmesi algorithmalarına dair bir çalışmadır. Tezin organizasyonu bu algoritmalarının öğrenme biçimine dayanmaktadır. İki temel öğrenme tipi :

- Gözeticili
- Gözeticisiz

öğrenmedir.

Gözeticili öğrenmede eğitim kümesi sınıf adlarını içerir, yani eğitim kümesindeki her bir nokta için doğru kategoriler verilmiştir. Gözeticili öğrenme algoritmaları önceden sınıflanmış eğitim datasını kullanırlar. İlk safhada algoritma sınıflı eğitim datasını kullanarak tüm kategorileri öğrenir. Değişik kategorileri veya sınıfları öğrenmekten kastımız veriyi oluşturan ayrık parçaları anlamaktır. İkinci safhada algoritma yeni gelen test noktası için öğrenilen sınıflardan bir tanesini tahmin eder. Bu yeni gelen test noktası için ilk safhada öğrenilen sınıflardan birinin tahmin edilme sürecine sınıflandırma adı verilir. Sınıflandırma kategorilerin hazır olarak verildiği bir eğitim verisiyle başlar ve gelen test kümesini bu hazır verilen ayrık parçalara sınıflar. Regresyon problemlerinde veri reel sayılarla etiketlenir (yani çıktı süreklidir). Sonuç olarak sınıflandırma ve regresyon problemleri gözeticili öğrenme problemleridir [Hastie et al. (2009)].

Gözeticisiz öğrenmede sınıf adları yani ayrık parçaların isimleri eğitim kümesinde verilmemiştir. Gözeticisiz öğrenme algoritmasından veri içindeki grupları bulması beklenir. Verinin kendi içinde ortak özellikleri olan parçalara ayrılmasına gruplama denir. Çoğu gruplama problemi eğiticisiz öğrenme problemidir [Hastie et al. (2009)].

2. GÖZETİCİLİ ÖĞRENME ALGORİTMALARI

2.1 Lineer Regresyon

Lineer regresyon bir gözeticili öğrenme algoritmasıdır.Lineer regresyon algoritması her örnek için doğru değerleri de içeren bir eğitim verisiyle çalışmaya başlar. Veriye algoritma düzgün bir lineer modeli yerleştirir, yani bir doğruyu veri üzerine oturtur. Sonrasında gelen yeni test noktası için eğitim aşamasında oluşturan bu doğru kullanılarak karar verilir [Ng (2000)].

Lineer regresyonun hipotezini aşağıdaki gibidir :

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots$$

Burada x_0 'ı 1 olarak kabul edilir. Böylece hipotez şu şekilde yazılabilir :

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

Burada θ parametre vektörüdür ve x girdi vektörüdür. İyi bir yaklaştırma yapmak istenmektedir, başka bir deyişle $h(x)$ ve y arasındaki mesafenin küçük olmasını istenmektedir. Bu beklentiyi formal olarak anlatmanın yolu şu maliyet fonksiyonunu tanımlamaktır :

$$J(\theta) = \frac{1}{2m} \sum_{i=0}^{m} (h_\theta(x^i) - y^{(i)})^2$$

θ yı uygun seçerek $J(\theta)$ yı minimize etmek amaçlanır. θ yı uygun seçmenin yolu, bir tane arama algoritmasına başvurmaktan geçer. Bu arama algoritmasına bir başlangıç tahmini verilir. Sonrasında arama algoritması ardarda θ yı güncelleyerek $J(\theta)$ yı daha küçük yapar. Regresyon problemlerinde en çok kullanılan arama algoritması *Gradient Descent* dir. Güncelleme kuralı :

$$\theta_j := \theta_j - \alpha \frac{d\theta_j}{dj}.$$

Burada sayısal parametre olan α ya gradient descent'in öğrenme hızı denir. Tüm j değerleri için θ_j lerin güncellenmesi aynı anda yapılır. Algoritma θ_j leri belli bir yakınsama

durumuna ulaşılıncaya kadar günceller. Bu öteleme işleminden sonra elde edilen θ değeri maliyet fonksiyonunu minimize eder. Eğer bu minimum yeterince küçükse, hipotez veriyle uyumludur denir [Ng (2000)].

Optimum θ ya ulaşmak için Gradient Descent yerine Normal denklemler metodunu da kullanabilir. Normal Denklemler metodu bazı durumlarda Gradient Decent metodundan çok daha hızlı çalışır. Normal denklemler metodu θ 'yı bir basamakta çözebilir, yani öteleme gerektirmez. Normal denklemler metodu θ 'yı analitik olarak aşağıdaki gibi çözebilir:

$$\theta = (X^T X)^{-1} X^T y.$$

Burada X her eğitim örneğinin transpozunu alarak oluşturulan bir $m \times n + 1$ matrisdir.

$$X = \begin{bmatrix} \cdots & (x^1)^T & \cdots \\ \cdots & (x^2)^T & \cdots \\ \vdots & \vdots & \ddots \quad \vdots \\ \cdots & (x^m)^T & \cdots \end{bmatrix}$$

Normal denklemler metodunda öteleme olmadığı için öğrenme hızı α deneme yoktur. Bu avantajlı tarafıdır. Fakat n in çok büyük değerleri için $(X^T X)$ in tersini almak zor bir iştir ve normal denklemler metodu oldukça yavaş çalışır. Yani küçük n değerleri için ($n \leq 1000$ genellikle) normal denklemler metodu tercih edilmelidir. Daha büyük n değerleri için Gradient Descent tercih edilir. Normal denklemler metodu sadece lineer regresyon için çalışan bir metodur. Diğer öğrenme algoritmalrında geçerli değildir [Ng (2000)].

Çoklu lineer regresyonda hipotez aşağıdaki formattadır :

$$h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n.$$

burada x_0 '1 dir. Maliyet fonksiyonu şöyledir:

$$\frac{1}{2m} \sum_{i=0}^{m} (h_\theta(x^i) - y^i)^2.$$

Bu durumda Gradient Descenti güncelleme kuralı

$$\theta_j = \frac{1}{m}\sum_{i=0}^{m}(h_\theta(x^i) - (y^i))x_j^i.$$

Güncelleme $j = 0, 1, 2, ..., n$ için aynı anda yapılır.

Lineer regresyonu bilgisayara aktarmak için etmek için bazı faydalı öneriler

Özellikleri normalize etme Gradient Descent in performansını artırabilir. Gradient Descent i özellikleri ölçeklendirdikten (feature scaling) ten sonra uygularsak, optimum θ daha hızlı bulunur, yakınsama daha az sayıda ötelemeyle gerçekleşir.

Regresyon problemlerinde öğrenme hızı α dikkatli seçilmelidir. α yeterince küçük seçilmesse yakınsamadan önce sürekli azalmasını istediğimiz $J(\theta)$ artabilir. α çok küçük olursa da yakınsama hızı çok düşer.

Sonuç olarak, Gradient Descent in yakınsayıp yakınsamadığını anlamak için ötelemeler arasında maliyetteki değişimler kontrol edilmelidir. Eğer $J(\theta)$ daki düşüş baştan belirlenen bir tolere etme miktarından azsa, yakınsama noktasına ulaşıldığı anlaşılır [Ng (2000)].

2.2 Logistic Regression

Logistik regresyon bir sınıflandırma algoritmasıdır. Sınıflandırma problemlerinde tahmin edeceğimiz değişken y ayrık bir değişkendir, sürekli değildir. Logistik regresyon uygulanan sınıflandırma problemlerine iyi bir örnek tümör problemleridir [Ng (2000)]. Tümör verisi iki alternatifli (binary) logistik regresyon kullanarak iyi huylular sınıfı ve kötü huylular sınıfı olarak iki ayrı sınıfa bölünür. Bu iki sınıf pozitif ve negatifler diye de isimlendirebilir. Genel olarak negatif sınıf 0 la, pozitif sınıf 1 le gösterilir. Yani tahmin edilen değişken olan bağımlı değişken y iki farklı değer alabilir. (çok sınıflı logistik regresyonda y sınıf sayısı kadar farklı değer alabilir.)

Logistik regresyon fonksiyonunu (sigmoid fonksiyonu da denir) şöyle tanımlanır :

$$g(x) = \frac{1}{1 + e^{-x}}$$

x in büyük pozitif değerleri için , sigmoid 1 e yakındır ve x in büyük negatif değerleri için sigmoid 0 a yakındır. Sigmoid(0) 0.5'e eşittir.

Sigmoid fonksiyonu kullanılarak logistik regresyon fonksiyonu şöyle yazılır:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

Girdi x olarak verildiğinde, $h_\theta(x)$ i y nin 1 olması olasılığı olarak düşünülür, yani $h_\theta(x) = P(y = 1|x; \theta)$. Aynı zamanda y sadece 1 ve 0 değerlerini alabildiği için, $P(y = 0|x, \theta) = 1 - P(y = 1|x; \theta)$ doğrudur.

$z = \theta^T x$ i tanımlansın. Bu durumda $z \geq 0$ ise $g(z) \geq 0.5$ olur ve $z < 0$ ise $g(z) \leq 0.5$ olur. Başka bir deyişle, $\theta^T x \geq 0$ ise $p(y = 1)$ olasılığı 0.5 den büyük eşittir. $\theta^T x < 0$ ise aynı olasılık 0.5 den küçük eşittir. Sonuç olarak, $|\theta^{\mathbf{T}}\mathbf{x}|$ **sıfırdan oldukça farklıysa**, yeni örnek x için doğru sınıf rahatlıkla tahmin edilebilir [Hastie et al. (2009)].

İki sınıflı logistik regresyon veri noktalarını iki sınıfa parçalar. y 'si 0 olan data noktaları ile y 'si 1 olan data noktaları oluşturalan karar verme sınırın farklı taraflarındadır. Bu karar verme sınırı sadece logistik regresyon hipotezi ve parametretreleri tarafından belirlenir. Logistik regresyon hipotezi ve parametreleri kompleks ve doğrusal olmayan bir karar sınırı oluşturabilirler.

Maliyet Fonksiyonu

Sadece bir örnek için maliyet fonksiyonu şöyledir :

$$J(h_\theta(x), y) = -y \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$$

m tane nokta içeren bir eğitim verisi için maliyet fonksiyonu

$$\begin{aligned} J(\theta) =& \frac{1}{m} \sum_{i=0}^{m} cost(h_\theta(x^i), (y^i)) \\ =& \frac{1}{m} \sum_{i=0}^{m} (-y^i) \log(h_\theta(x^i)) - (1 - (y^i)) log(1 - h_\theta(x^i)) \end{aligned}$$

Bu maliyet fonksiyonu konvekstir ve lokal optimumu yoktur. Maliyet fonksiyonunun gradienti bir vektördür. Gradient vektör ve θ aynı uzunluktadır. Gradient vektörünün j. bileşeni :

$$\frac{dJ(\theta)}{d\theta_j} = \frac{1}{m}\sum_{i=0}^{m}(h_\theta(x^i) - (y^i))x_j^i$$

Bir çözücü gradient vektörünü kullanarak $J(\theta)$ 'yı minimize eder ve θ_j 'leri aynı anda güncelleyerek optimum θ 'yı çözer. Yeni verilen bir veri noktası için,

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

çözücü tarafından sunulan en son parametreleri kullanarak hesaplanır. Ve sonunda $P(y = 1|x, \theta)$ 'yı elde etmiş oluruz.

Düzenlileştirme

Eğer hipotezde çok fazla öznitelik varsa, veya yüksek dereceli bir polinom veriye oturtulursa, öğrenme hipotezi eğitim kümesine çok fazla uyabilir ve çok sıkı bağlı olabilir. Böyle durumlarda algoritma eğitim kümesi için çok doğru sonuçlar versede, test kümesi için iyi sonuçlar vermekten uzaktır. Bu tarz öğrenme hipotezi olan algoritmaların varyansı yüksektir. Bu duruma **aşırı öğrenme** denir. Diğer taraftan, eğer çok düşük dereceli, basit bir polinom eğitim datasına oturtulursa bu sefer de eğitim datası yeterince ifade edilmemiş olur. Bu duruma da **eksik öğrenme** (under fitting) denir. Bu tarz algoritmaların problemi yüksek yanlılıktır (high bias) [Ng (2000)].

Aşırı öğrenme (overfitting) problemi için çözümlerden bir tanesi öznitelik sayısını düşürmektir. Fakat bazı öznitelikleri silmek demek aslında bazı bilgileri silmek anlamına gelir. Aşırı öğrenme problemi için başka bir çözüm ise *düzenlileştirmedir.* Düzenlileştirme tekniğinde özniteliklerden silmek yerine, maliyet fonksununa parametrelerin (θ_J lerin) aşırı değerlerini cezalandırmak amaçlı ekstra bir cezalandırma terimi eklenir. En yaygın olarak kullanılan cezalandırma terimi θ_J vektörünün büyüklüğüdür. Eğer çok fazla öznitelik varsa ve her öznitelik tahmin değişkeni y 'ye biraz katkı yapıyorsa bu yöntem özellikle kullanışlıdır [Ng (2000)].

Logistik regresyon için düzenlileştirilmiş maliyet fonksiyonu :

$$J(\theta) = \frac{1}{m}\sum_{i=0}^{m}(-y^i)\log(h_\theta(x^i)) - (1-(y^i))\log(1-h_\theta(x^i)) + \frac{\lambda}{2m}\sum_{j=1}^{n}(\theta_j)^2.$$

Burada λ düzenlileştirme parametresidir ve $(\lambda/2m)\sum_{j=1}^{n}(\theta_j^2)$ düzenlileştirme terimi olarak adlandırılır. İlk amaç eğitim setine modelin uymasıdır yani

$$\sum_{i=0}^{m}(h_\theta(x^i) - (y^i))^2$$

küçük tutmaktır. İkinci amaçta düzenlileştirme terimini küçük tutmaktır. λ **bu iki amaç arasında bir denge güder, böylece hipotezi basit tutar ve karar sınırını değiştirir**. Çok kompleks bir eğriyi veriye uydurmaktansa, basit bir eğriyi veriyi uydurur ve aşırı öğrenmeyi engeller.

Düzenlileştirilmiş logistik regresyonda kullandığımız maliyet fonksiyonu için gradient :

$$\frac{d}{d\theta_j}J(\theta) = \frac{1}{m}\sum_{i=0}^{m}(h_\theta(x^i) - (y^i))x_j^i,\ j = 0 \text{ için}$$

ve

$$\frac{d}{d\theta_j}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i) - (y^i))x_j^i + (\lambda/m)\theta_j,\ j = 1 \text{ için}$$

şeklinde verilir.

2.3 Destek Vektör Makineleri

Destek vektör makineleri (Support Vector Machines) eğiticili öğrenme tekniklerinden birisidir ve veri sınıflandırma için oldukça kullanışlıdır. DVM istatistiksel öğrenme teorisinden geliştirilmiştir. DVM eğitim setine dayanarak bir model kurar ve bu model test datası için hedeflenen değerleri tahmin eder. DVM nin altındaki fikir veriyi geniş marjinlerle ayırmaktır [Vapnik and Vapnik (1998)].

Veri madenciliğinde [Burbidge and Buxton (2001)], kredi derecelendirme uygulamalarında [Chen and Shih (2006)], tıbbi araştırmalarda[Übeyli (2007)] ve benzeri işlerde destek vektör makineleri kullanılmıştır.

Destek vektör makineleri oldukça güçlü bir öğrenme algoritmasıdır. Küçük eğitim verilerinden başarılı genelleme yapabilir ve yüksek boyutlu az miktarda data için de iyi sonuçlar verir [Shen et al. (2006)].

2.3.1 Lineer Destek Vektörleri

Linear Olarak Ayrılabilen Data, Sert Marjin

Eğitim setinde m tane veri noktası olduğunu varsayılsın. Ayrıca her data noktasının n tane özniteliği olduğu ve $y_i = +1,\ y_i = -1$ sınıflarından birine ait olduğu varsayılsın. (iki sınıflı sınıflandırma için)

$$(x_i, y_i), i = 1, \ldots, m,\ \ y_i \in \{-1, 1\}\ ,\ \text{ve}\ x_i \in \mathbb{R}^n.$$

DVM iki sınıfı ayıran bir hiperdüzlem çizer. Bu ayırıcı hiperdüzlem denklemi :

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 0.$$

Burada $\mathbf{w}$ hiperdüzlemin normal vektörüdür ve $\frac{\mathbf{b}}{|\mathbf{w}|}$ hiperdüzlemden orijine olan uzaklıktır.

Bu hiperdüzleme optimum ayırıcı hiperdüzlem denir. Böyle denmesinin sebebi de destek vektör makinelerinin hiperdüzlemi destek vektörlerinden mümkün olduğunca uzak olacak şekilde yönlendirmesidir. Sınıflandırıcı parametreler $\mathbf{w}$ ve $\mathbf{b}$ dir ve bunların bulunması gerekir [Cherkassky and Mulier (2007)]. Eğitim datasındaki noktalar için şu doğrudur:

$\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b} \geq 0$ eğer $y_i = +1$.
$\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b} \leq 0$ eğer $y_i = -1$.

Bu iki eşitsizlik tek bir eşitsizlik şeklinde yazılırsa

$y_i(\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b}) - 1 \geq 0$ her i için doğrudur.

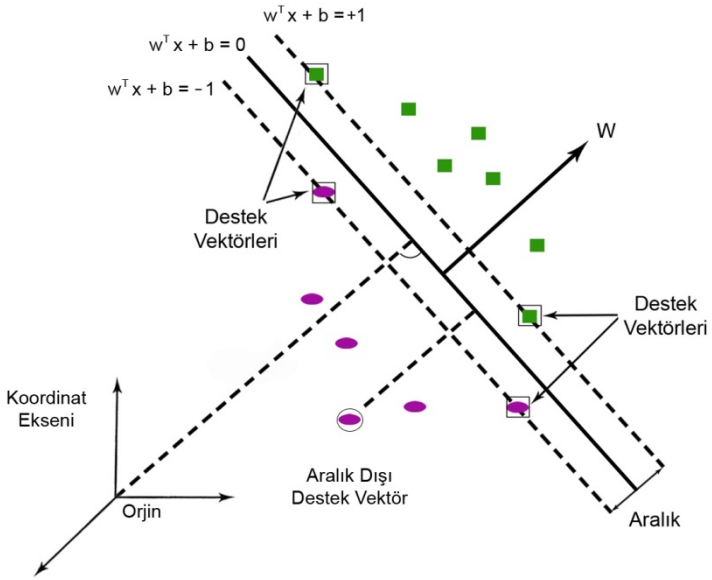


Figure 2.1: Lineer sınıflandırma
Kaynak: Melgani and Bruzzone (2004)

Yukarıdaki şekilde destek vektörleri H_1 ve H_2 hiper düzlemlerini belirler. Burada

$\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b} = -1$, H_1 için
$\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b} = +1$, H_2 için

doğrudur.

H_1, H_2, H hiper düzlemleri paraleldir ve H_1, $H2$ arası uzaklık $\frac{2}{|\mathbf{w}|}$ dır, yani marjin uzunluğu $\frac{2}{|\mathbf{w}|}$ dir. DVM marjini maksimize eder, yani $|\mathbf{w}|$ nin minimum değeri gereklidir [Fletcher (2009)]. Bu problem ikinci derece bir optimizasyon problemidir.

$$min_{x,y}\ \frac{1}{2}|\mathbf{w}|^2 \text{ öyleki } y_i(\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b}) - 1 \geq 0 \text{ her i için.}$$

Eğitim örnekleri için kısıtlar ayırıcı hiper düzlemin doğru tarafını formülize eder. Bu problem için Lagrange fonksiyonu :

$$L_p(\mathbf{w}, \mathbf{b}, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i y_i(\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b}) + \sum_{i=1}^{n} \alpha_i.$$

şeklinde yazılır. Burada bütün lagrange çarpanları,yani tüm α_i ler pozitifdir.

$L_p(\mathbf{w}, \mathbf{b}, \alpha)$ 'nin $\mathbf{w}$ ve and $\mathbf{b}$ ye göre türevini alınırsa ve türevleri sıfıra eşitlenirse

$$\frac{\partial L_p}{\partial w} = 0 \implies \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i(\mathbf{x}_i).$$
$$\frac{\partial L_p}{\partial b} = 0 \implies \sum_{i=1}^{n} \alpha_i y_i = 0.$$

Bu türevler yerine koyulduğunda lagrange fonksiyonu maksimize etmek için şu dual formu verir:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \text{ öyleki } \alpha_i \geq 0 \text{ her i için ve } \sum_{i=1}^{n} \alpha_i y_i = 0.$$

Bunun denk formatı da

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\alpha^{\mathbf{T}}\mathbf{H}\alpha, \text{ burada } H_{ij} = y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}.$$

Sonuç olarak çözülmesi gereken

$$max_{\alpha}(\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\alpha^{\mathbf{T}}\mathbf{H}\alpha) \text{ öyleki } \alpha_i \geq 0 \text{ her i için ve } \sum_{i=1}^{n} \alpha_i y_i = 0.$$

Bir ikinci derece problem çözücü (QP solver) yardımıyla bu problem çözülür. Bu ikinci dereceden problem çözücü α yı verir [Fletcher (2009)].

İlk olarak α yı bulduktan sonra , w = $\sum_{i=1}^{n} \alpha_i y_i(x_i)$ denklemi sayesinde w elde edilir. Bu son denkleme birinci lagrange şartı denir. İkinci lagrange şartını, $\sum_{i=1}^{n} \alpha_i y_i(x_i.w + b)$ = 0, sağlayan her destek vektörü için

$$y_s(x_s \cdot w + b) = 1$$

doğrudur.

$y_i^2 = 1$ olduğu için, y_s ile çarpılarak ve w yerine koyularak, b şu şekilde elde edilir :

$$b = y_s - \sum_{m \in S}(\alpha_m y_m \mathbf{x_m} \cdot \mathbf{x_s}).$$

Burada S destek vektörlerinin indislerinin oluşturduğu kümedir. **Uygulamalarda genel olarak b nin destek vektörleri kümesi üzerinde ortalaması alınarak bulunan değer kullanılır.**

En son basamakta w ve b bilindiği için, ayırıcı hiper düzlem, **w . x + b = 0**, çizilebilir. Sonuçta, **w.x' + b işareti** yeni örnek $\mathbf{x}'$ in sınıfını söyler [Fletcher (2009)].

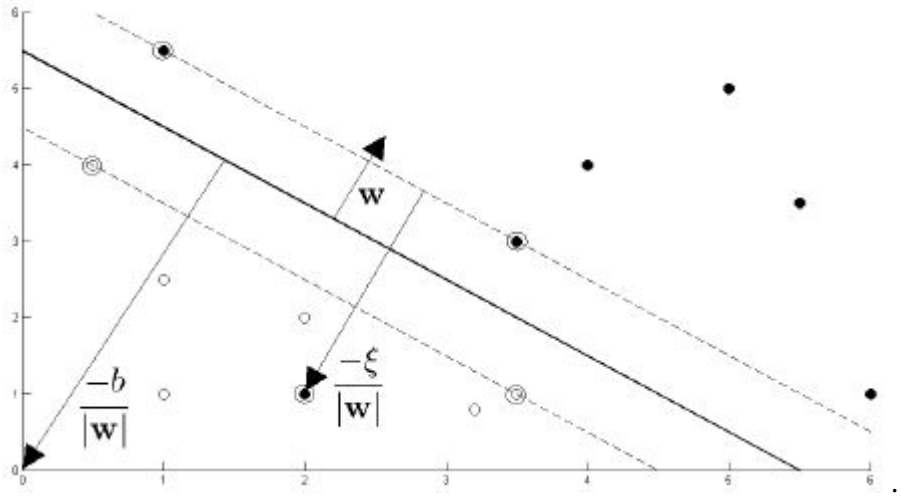


Figure 2.2: Lineer olarak ayrılamayan iki sınıf arasındaki hiper düzlem
Kaynak : Fletcher (2009)

Tam Olarak Lineer Ayrılamayan Data , Gevşek Marjin

Positive artık değişken (slack variable) ξ_i i= 1, . . . , n eklenerek yanlış sınıflandırılan data noktaları için kısıtlar rahatlatılır.

$\mathbf{x_i} \cdot \mathbf{w} + \mathbf{b} \geq 1 - \xi_\mathbf{i}$ $y_i = +1$ için.
$\mathbf{x_i} \cdot \mathbf{w} + \mathbf{b} \leq -1 + \xi_\mathbf{i}$ $y_i = -1$ için.
$\xi_\mathbf{i} \geq 0$ her i için.

Bu şöyle de yazılabilir :

$y_i(x_i.w + b) - 1 + \xi \geq 0$ öyleki $\xi_i \geq 0$ $\forall i$.

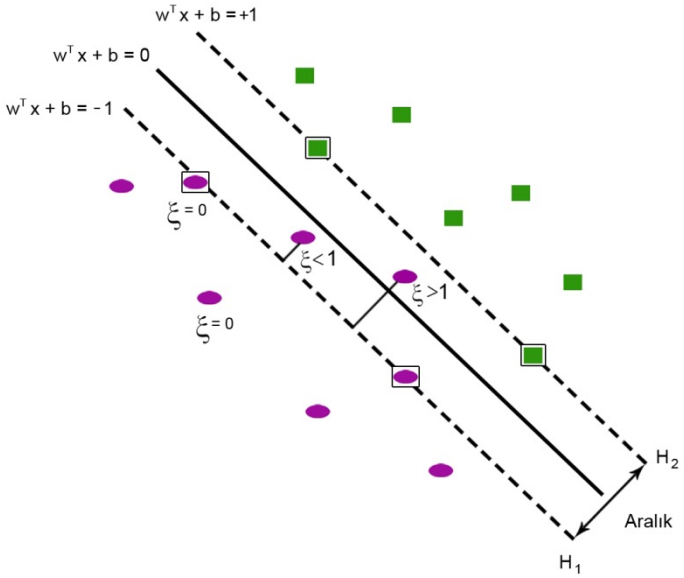


Figure 2.3: Gevşek değişkenler
Kaynak : Gönen and Alpaydin (2008)

Marjinin yanlış tarafındaki noktalar için ceza uzaklıkla orantılı olarak artmalıdır. C marjin büyüklüğü ve ceza arasında bir denge kurar.

Kısıtlar dikkate alındığında Lagrange fonksiyonu :

$$L_p(w, b, \xi, \alpha) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}[\alpha_i y_i(x_i.w + b) - 1 + \xi_i] - \sum_{i=1}^{n}\mu_i\xi_i$$

Burada tüm α_i 'lar sıfırdan büyük eşittir.

L_p(w, b, α) nin w, b ve ξ göre türevleri alınıp sıfıra eşitlenirse

$$\frac{\partial L_p}{\partial w} = 0 \implies w = \sum_{i=1}^{n}\alpha_i y_i x_i$$
$$\frac{\partial L_p}{\partial b} = 0 \implies \sum_{i=1}^{n}\alpha_i y_i = 0$$
$$\frac{\partial L_p}{\partial \xi_i} = 0 \implies C = \alpha_i + \mu_i$$

Burada $\mu_i \geq 0 \,\forall$ i olduğu için $\alpha \leq$ C dir.

Bu türevler dual formda yerine koyulunca şu problem oluşur :

$$\max_{\alpha} \left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\alpha^T \mathrm{H}\alpha\right) \text{ burada } H_{ij} = y_i y_j \mathbf{x_i}\cdot\mathbf{x_j}\ 0 \leq \alpha_i \leq C \text{ her i değeri için ve } \sum_{i=1}^{n}\alpha_i y_i = 0$$

Sonrasında ikinci derece denklem çözücü α yı bulur. Takip eden basamak $0 \leq \alpha_i \leq$ C bütün i değerleri için sağlayan destek vektörlerini bulmaktır. Böylelikle lineer ayrılabilir durumda olduğu gibi w ve b yi elde edilir. $\mathbf{w}\cdot\mathbf{x}' + \mathbf{b}$ ın **işareti** yeni örnek $\mathbf{x}'$ in sınıfını verecektir [Fletcher (2009)].

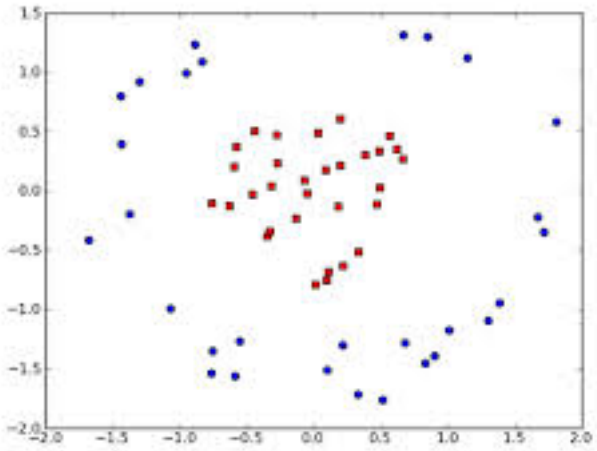


Figure 2.4: Lineer olarak ayrılamayan data

2.3.2 Lineer Olmayan Destek Vektör Makineleri

Çoğu veri kümesi küçük boyutlarda lineer olarak ayrılamaz. Bir örneği üstteki datadır.

Böyle veri kümeleri girdi uzayında lineer olarak ayrılabilir değildir. Böyle durumlarda *kernel* kullanmak iyi bir fikirdir.

Lineer olarak ayrılamayan veriye destek vektör makinelerini uygulamanın ilk basamağı

$$H_{ij} = y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} = y_i y_j \mathbf{k}(\mathbf{x_i}, \mathbf{x_j})$$

yazmaktır.

Burada $K(x_i, x_j)$ lineer kerneldir ve **kernel fonksiyonlarının** bir örneğidir. Kernel fonksiyonları vektörlerin iç çarpımlarına dayanır. Kernel fonksiyonları veriyi girdi uzayından daha yüksekboyutlu uzaylara taşır. Girdi uzayında lineer olarak ayrılamayan veri, daha yüksek boyutlu öznitelik uzayına taşındığında lineer olarak ayrılabilir. Daha yüksek boyutta lineer olarak ayırma şu şekilden görebilir.

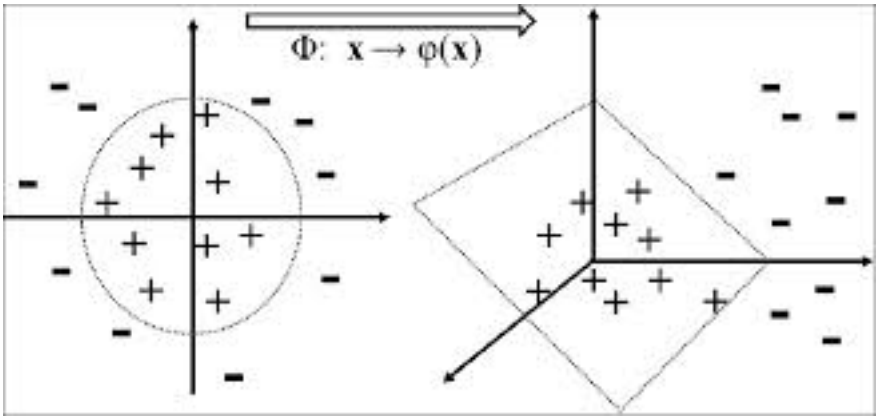


Figure 2.5: Yüksek boyuta taşındığında sınıfların ayrılması
Kaynak : Al-ani and Trad (2010)

$$\phi : \mathbf{X} \rightarrow \mathbf{F}$$

girdi uzayından daha yüksek boyutlu öznitelik uzayına bir taşıma fonksiyonudur.

ϕ taşıma foksiyonu ile aşağıdaki tanım yapılır :

$$\langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle = \mathbf{k}(\mathbf{x_i}, \mathbf{x_j}).$$

Burada $k(x_i, x_j)$ 'e kernel fonksiyonu denir. Kernel fonksiyonu simetrik, sürekli ve kesin artıdır (pozitive definite) [Alpaydin (2004)].

En meşhur ve yaygın kullanılan kernel fonksiyonu Gaussian Kerneldir ve şöyle tanımlanır:

$$k(x_i, x_j) = exp(-||x_i - x_j||^2/2\sigma^2)$$

Aşağıda soldaki şekil Dichotomous verisidir. Bu veri girdi uzayında lineer olarak ayrılabilir değildir. Sağındaki şekil bu Dichotomous datasının dairesel kernel kullanılarak taşınmış halini gösterir. Bu yeni hal taşınan öznitelik uzayında lineer olarak ayrılabilir durumdadır.

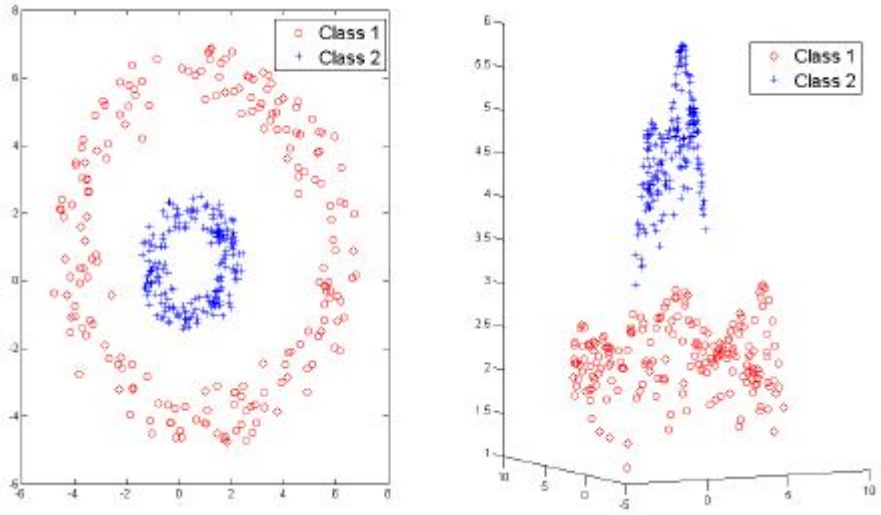


Figure 2.6: Dichotomous datası ve dairesel kernelle taşınmış hali
Kaynak: Fletcher (2009)

Destek vektör makineleri yardımıyla sınıflandırma problemini çözmenin ilk basamağı kerneli seçmektir. İkinci basamak $H_{ij} = (y_i y_j \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$ ı oluşturmaktır.

Sonrasında, denklem çözücü (QP) α yı aşağıdaki şu problemi

$$\max_{\alpha} (\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\alpha^{\mathbf{T}}\mathbf{H}\alpha), 0 \leq \alpha_i \leq \ C \ \forall \text{i ve } \sum_{i=1}^{n} \alpha_i y_i = 0$$

çözerek bulur.

İlk istenen değer α bulunduktan sonra, w = $\sum_{i=1}^{n} \alpha_i y_i \phi(x_i)$ formülüyle w elde edilir. Sonraki basamak hangi destek vektörlerinin $0 \leq \alpha_i \leq$ C $\forall$ i sağladığını belirlemektir. Sağlayan destek vektörlerinin indislerinin oluşturduğu kümeye S denir. Bu S kümesini kullanarak, b yi belirleme yolu şöyledir :

$$b = y_s - \sum_{m \in S} (\alpha_m y_m \phi(x_m) \cdot \phi(x_s))$$

Uygulamalarda genel olarak b nin destek vektörleri kümesi üzerinde ortalaması alınarak bulunan değer kullanılır. Neticede, $\mathbf{w}.\mathbf{x}' + \mathbf{b}$ **işareti** yeni örnek $\mathbf{x}'$ in sınıfını söyleyecektir [Fletcher (2009)].

2.4 En Yakın k Komşu Algoritması (k-NN)

En yakın k komşu algoritması, k-NN de denir, en basit öğrenme algoritmalarından bir tanesidir. k-NN hesaplamalı geometride, gen çıkarımında, protein etkileşim tahmininde ve bir çok benzeri uygulamada sıklıkla kullanılır. k-NN hem sınıflandırma hem de regresyon için kullanılabilir. Regresyon için olan uygulamalarda çıktı $y \in \mathbb{R}$ ve y en yakın k komşunun ortalaması ile hesaplanır. kNN ile yapılan sınıflandırmada çıktı y en çok karşılaşılan sınıfın ismidir, yani yeni data noktası en yakın k komşu içinde en yaygın olan sınıfa atanır. kNN çoklukla sınıflandırma için kullanılır, bu nedenle algoritmaya dair takip eden açıklamalar sınıflandırma düşünülerek yapılmıştır [Duda et al. (1999)].

k-NN altta yatan veri dağılımına dair bir varsayımda bulunmaz, yani parametrik olmayan bir algoritmadır. k-NN in eğitim aşaması hızlıdır. Fakat test aşaması için zaman ve hafıza gereksinimi tüm eğitim verisinin kullanılması sebebiyle fazladır.

Veri noktaları sayıdır veya çok boyutlu vektörlerdir. Veri noktaları bir metrik uzayın içindedirler ve genel olarak noktalar arasındaki uzaklık hesabında Öklid metriği kullanılır. Eğitim verisi hem veri noktalarını hem de sınıfları içerir.

Eğitim kümesi, test noktası ve k algoritmanın girdileridir. k sonucu belirlemede etkili olan komşuların sayısıdır. k dikkatli seçilmelidir. k yüksek seçilirse hesaplama maliyeti artar. k küçük seçilirse datadaki gürültülü kısım sonuçta belirleyici olur. k için basit ve çokça kullanılan bir seçim $\sqrt{n}$ 'dir.

Bir test noktası verilsin. İlk iş olarak algoritma bu test noktasının eğitim kümesindeki tüm noktalara olan uzaklıklarını hesaplar. İkinci olarak eğitim setindeki noktalar yakın olandan uzak olana doğru sıralanır. Son olarak en yakın k noktanın sınıflarına bakılır. Bunlar içinde en çok olan sınıf kazanır ve yeni gelen test noktasının sınıfı o olur.

2.5 Karar Ağaçları

Gözeticili öğrenme metodlarından olan karar ağaçları yaygın olarak kullanılan sınıflandırma algoritmalarından bir tanesidir. Karar ağaçları karar alma süreçlerini anlatır. Kategorileme problemlerinde sıkça kullanılırlar. Bir karar ağacı örnekleri sınıflandırmanın sade bir gösterimidir. Karar ağaçları girdi olarak verilen bir çok özniteliğe dayanarak hedef değişkenin değerini tahmin eder [Rokach (2008)].

Karar ağacı akış şeması şeklinde bir ağaç yapısıdır. Ağacın her iç düğümü (node) bir özniteliğe dair bir test gösterir. Her dal bu testin çıktılarını gösterir. Uç düğümler (yapraklar) da sınıf etiketlerini (isimlerini) gösterir.En tepedeki düğüm kök düğümüdür. Yapraklar kararlardır. Kararları oluşturmak için kökten başlayarak yaprağa doğru olan yol üzerinde ilerlenir. Yani kökten başlayarak yaprağa doğru ilerlenirse, çıkarımlar elde edilir. Bu çıkarımlarla sınıflandırma kurallarını elde edilir. Okumayı kolaylaştırmak amacıyla bu kurallar eğer-ise kuralları şeklinde ifade edilebilir [Maindonald and Braun (2010)].

Karar ağaçları şu tarz problemler için kullanışlıdır :

- Eğer örneklerin gösterimi öznitelik-değer çifti şeklindeyse (Özniteliklerin sonlu bir listesi vardır ve her örnek her bir öznitelik için sadece bir değer tutar)
- Eğer hedef fonksiyon kesikli (discrete) çıktılar veriyorsa (Bu algoritma sürekli çıktı durumunda da kullanılabilir)
- Ayırıcı (disjunctive) tanımlar cevapta gerektiğinde

Bir veri için değişik karar ağaçları kurmak mümkündür. Fakat asıl amaç veriye uyan en kısa, en küçük karar ağacını çizmektir. ID3 algoritması bu en kısa karar ağacını çizmek için kullanılır [Mitchell (1999)].

2.5.1 Karar Ağacı Kurma

Geçmiş bilgiler ve deneyimler karar ağacının nasıl kurulacağına rehberlik eder. Doğru sınıfları da içeren bir eğitim verisiyle başlanır. Örnekleri tanımlayan bir öznitelik kümesi vardır. Her öznitelik sonlu sayıda değer alır. **Eğitim seti üstündeki örnekler karar ağacının yapısını ortaya çıkarmak için kullanılır. Bu yapı netleştikten sonra da test kümesindeki veri noktaları için rahatlıkla doğru karar verilebilir.**

Eğer iki örnek öznitelikler için tam olarak aynı değerlere sahipse, bunların aynı kategoride olduğu varsayılır. Bu varsayım verinin tutarlı olmasını garanti eder. Veriyi değişik karar ağaçlarıyla sınıflara bölmek mümkündür, ve yüzde yüz doğru olacak şekilde ayıran bir karar ağacı bulmak da her zaman mümkündür.

2.5.2 Rastsal Bir Dağılımın Entropisi

Her düğümde hangi özniteliği kullanacağını anlamak (hangi özniteliğe göre parçalayacağını anlamak) karar ağacı kurmada kilit öneme sahiptir. Özniteliklerin herhangi bir örnek için aldığı değerleri yan yana sıralayıp bir vektör oluşturduğunu düşünelim. Bu yapıya tüp (tuple) denir. Bu tüpleri birbirinden farklı sınıflara göre ayırmak için kullanılan özniteliği bulmanın kriteri bilgi kazanımıdır. Eğer bilgi kazanımı kriter olarak alınırsa, ikiye yada daha fazla parçaya dallanma mümkündür. Bilgi kazanımın bir ölçümü vardır. Bu ölçüme **entropi** denir. Entropi rastgele bir örnekler setinin ne kadar saf olmadığına dair bilgi verir [Mitchell (1999)].

X rastsal değişkeni verilsin. Bu değişkenin alabileceği değerler X_1, X_2, X_3, ..., X_n olsun. Bu değerlerin olasılıklarıda sırasıyla p_1, p_2, p_3, ..., p_n olsun. Bu durumda dağılımın entropisi :

$$H(X) = H(p_1, p_2, p_3, \ldots, p_n) = -\sum_{i=1}^{n} p_i \, log_2 p_i.$$

Küçük entropi çoğu örneğin tek bir sınıfta olduğuna işaret eder. Elemanlar güzel bir şekilde sıralanmışlardır ve özel bir örneği bulmak oldukça kolaydır. Bu karışık olmayan durumdur. Eğer örnekler değişik sınıflara dağılmışsa, özel bir elemanı bulmak zordur. Bu öncekinden daha karmaşık ve kaoslu bir ortamdır. Böyle ortam için entropi değeri daha yüksektir.

Verilen X için yukardaki entropi formülünü uygulanır.

Örneğin, p_1 = 0.5, p_2 = 0.5 ise, dağılımın entropisi H(p_1, p_2) = -(-0.5 - 0.5) = 1 dir

Eğer $p_1 = p_2 = p_3 = ... = p_n = \frac{1}{n}$, H($\frac{1}{n}$, $\frac{1}{n}$, $\frac{1}{n}$, ..., $\frac{1}{n}$) = $\log_2 n$ olur.

2.5.3 Entropi Kazancı

Özel bir düğüm için en iyi özniteliği yani en iyi ayırma kriterini şeçmenin yolu entropiye bakmaktır.

Entropi hesabında kullanılan terimler ve açıklamaları şöyledir.

S= Örneklerin oluşturduğu kümedir.

Değer(A) : A özniteliğinin alması mümkün olan değerlerdir.

v : Değer(A) kümesinin bir elemanıdır.

S_v = A özniteliği için v değeri alan örneklerin kümesidir.

$$\text{Kazanç}(S, A) = \text{Entropi}(S) - \sum_{v \in \text{Değer}(A)} \frac{|S|\,\text{Entropi}(S_v)}{|S|}.$$

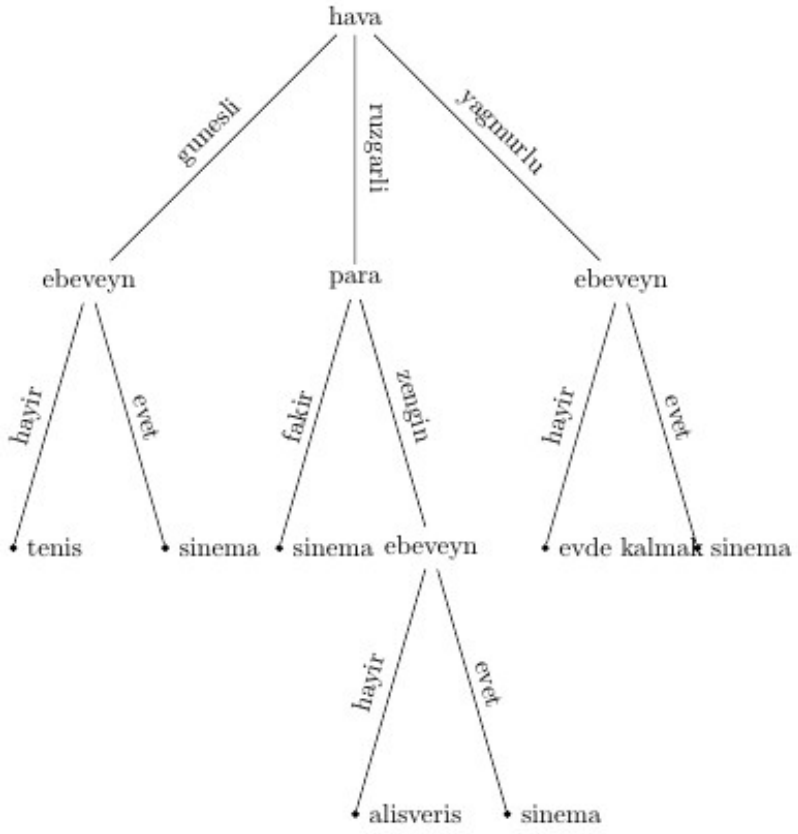
2.5.4 ID3 Algoritması

Veriye uyan en kısa karar ağacını bulmayı ID3 algoritması sağlar [Hastie et al. (2009)]. Bu algoritmanın detayları aşağıdadır.

1. Her bir öznitelik için entropi hesaplanır.
2. En düşük entropili özniteliği seçilir. (Buna A diyelim)
3. Veri A ya dayanarak değişik ayrık parçalara ayrılır. (Yani her grupta A tek bir değer alacak.)
4. Değişik dalları olan bir ağaç oluşturulur. (A nın mümkün olan değerleri dalları oluşturur.)
5. Oluşan her bir alt ağaç için, bu süreç birinci basamaktan itibaren tekrarlanır.
6. Her bir ötelemede, toplamda dikkate alınması gereken öznitelik sayısı bir azaltılmış olur. Bu süreç düşünülmesi gereken öznitelik kalmassa veya oluşan alt ağaçta veri öznitelik için tek bir değer içerirse sonlanır.

2.5.5 Karar Ağacı Örneği

[Monz (2007)]


hava
gunesli
ruzgarli
yagmurlu
ebeveyn
para
ebeveyn
hayir
evet
fakir
zengin
hayir
evet
tenis
sinema
sinema
ebeveyn
evde kalmak
sinema
hayir
evet
alisveris
sinema

Karar Agaci

Yukarıdaki karar ağacının kurulma ayrıntıları:

ID3 bir öznitelik seçer ve veriyi bu özniteliğe göre ayırmaya başlar. Her ayırma entropi kazancını maksimize etme kriterine göre yapılır. Kullanılan veri tablo 2.1'de gösterilmiştir ve kategoriler kümesi S şöyledir :

S = { sinema, tenis, alışveriş, evde kalmak }.

Table 2.1: Haftasonu Örneği

Haftasonu(Örnek)	Hava	Ebeveyn	Para	Karar kategorisi)
H1	güneşli	evet	zengin	sinema
H2	güneşli	hayır	zengin	tenis
H3	rüzgarlı	evet	zengin	sinema
H4	yağmurlu	evet	fakir	sinema
H5	yağmurlu	hayır	zengin	evde kalmak
H6	yağmurlu	evet	fakir	sinema
H7	rüzgarlı	hayır	fakir	sinema
H8	rüzgarlı	hayır	zengin	alışveriş
H9	rüzgarlı	evet	zengin	sinema
H10	güneşli	hayır	zengin	tenis

Entropi(S) = -$p_{sinema}\log_2(p_{sinema})$ - $p_{tenis}\log_2(p_{tenis})$ -$p_{\text{alışveriş}}\log_2(p_{\text{alışveriş}})$ - $p_{\text{evde kalmak}}\log_2(p_{evdekalmak})$

=-(6/10)*$\log_2$(6/10)-(2/10)*$\log_2$(2/10)-(1/10)*$\log_2$(1/10) -(1/10)*$\log_2$(1/10) = 1.571

kazanç(S,hava) = 1.571 - ($|S_{\text{güneşli}}|/10$)*Entropi($S_{\text{güneşli}}$) - ($|S_{\text{rüzgarlı}}|/10$)*entropi($S_{\text{rüzgarlı}}$) - ($|S_{\text{yağmurlu}}|/10$)*Entropi($S_{\text{yağmurlu}}$) =1.571 - (0.3)*(0.918) - (0.4)*(0.81125) - (0.3)*(0.918) = 0.70

kazanç(S, ebeveyn) = 1.571 - ($|S_{\text{hayır}}|/10$)*Entropi(S_{evet}) = 1.571 - (0.5)*0 -(0.5)*1.922 = 1.571 -0.961 = 0.61

kazanç(S,para) = 1.571 - ($|S_{\text{zengin}}|/10$)*Entropi(S_{zengin}) - ($|S_{\text{fakir}}|/10$)*Entropi(S_{fakir}) = 1.571 - (0.7)*(1.842)- (0.3)*0 = 1.571 - 1.2894 = 0.2816

kazanç(S, hava) en büyük kazanç. Yani karar ağacının ilk düğümü hava olmalıdır.

Güneşli, rüzgarlı ve yağmurlu hava özniteliği için muhtemel değerler. Bu sebeble ilk düğümden güneşli, rüzgarlı ve yağmurlu dalları oluşturur.

İlk dal $S_{\text{güneşli}}$. 1, 2 ve 10 uncu haftanın hava değeri güneşli, yani $S_{\text{güneşli}}$ =H1, H2, H10.

Table 2.2: Haftasonu Örneği Parça 1

haftasonu	hava	ebeveyn	para	karar
H1	güneşli	evet	zengin	sinema
H2	güneşli	hayır	zengin	tenis
H10	güneşli	hayır	zengin	tenis

H1, H2, ve H10 un kategorileri sırasıyla sinema, tenis ve sinema. $S_{\text{güneşli}}$ kümesi boş olmadığından ve birden fazla elemana sahip olduğundan , bir A öznitelik düğümü koyulur. A hava olamaz çünkü hava kullanıldı. Kullanabilecek öznitelikler kümesinin elemanları ebeveyn ve paradır.

kazanç($S_{\text{güneşli}}$, ebeveyn)= 0.918 - ($|S_{\text{evet}}|/10$)*Entropi(S_{evet}) - ($|S_{\text{hayır}}|/10$)*Entropi($S_{\text{hayır}}$) = 0.918 - (1/3)*0 - (2/3)*0 = 0.918

Gain($S_{\text{güneşli}}$, para)= 0.918 - ($|S_{\text{zengin}}|/10$)*Entropi(S_{zengin}) - ($|S_{\text{fakir}}|/10$)*Entropi(S_{fakir}) = 0.918 - (3/3)*0.918 - (0/3)*0 = 0

Buradan A nın ebeveyn olduğunu görülmüştür. Burda S_{evet} in entropisinin sıfırdır çünkü tüm örnekleri sinema kategorisindedir. Benzer mantıkla $S_{\text{hayır}}$ da sıfıra eşittir. Bunlara bakarak karar ağacını kurarken neden entropi kazancını kullandığımıza dair fikir edinebilir. S_{evet} içindeki örneklerin hepsi aynı sınıfta, $S_{\text{hayır}}$ ın içindeki örneklerin hepsi aynı sınıftadır. Bu da zaten karar ağacında yapılmak istenen şeydir.

Ebeveyn özniteliğinin alabileceği muhtemel değerler evet ve hayırdır. Yani ebeveyn düğümünden evet ve hayır dalları çizilecektir. Başlangıç kümesi S 'dir. S den sonra $S_{\text{güneşli}}$ kümesine bakıldı. Şimdiki küme $S_{\text{güneşli, evet}}$. Bu kümedeki tek örnek H1. Yani evet dalı kategoriyi söyleyen yaprak sinemadır. Benzer şekilde $S_{\text{güneşli, hayır}}$ sadece H2, H10 u içeriyor. Bu iki haftanın karar değeri tenis. O zaman hayır dalının karar yaprağında tenis vardır.

İkinci ana dal $S_{\text{rüzgarlı}}$. $S_{\text{rüzgarlı}} = \{H3, H7, H8, H9\}$.

Table 2.3: Haftasonu Örneği Parça 2

Haftasonu	hava	ebeveyn	para	karar
H3	rüzgarlı	evet	zengin	sinema
H7	rüzgarlı	hayır	fakir	sinema
H8	rüzgarlı	hayır	zengin	alışveriş
H9	rüzgarlı	evet	zengin	sinema

$H3, H7$ ve $H9$ un karar değeri sinemadır. H8 in karar değişkeni alışveriştir. Farklı olduğu için bir B düğümü oluşturulur. B hava olamaz. B nin olabileceği öznitelik kümesi $\{ebeveyn, para\}$.

kazanç($S_{\text{rüzgarlı}}$, ebeveyn)= 0.811 - ($|S_{\text{evet}}|/10$)*Entropy(S_{evet}) - ($|S_{\text{hayır}}|/10$)*Entropi($S_{\text{hayır}}$) = 0.811 - (2/10)*0 - $(2/10)[-(1/2)log_2(1/2) - (1/2)log_2(1/2)]$ - = 0.311

kazanç($S_{\text{rüzgarlı}}$, para)= 0.811 - ($|S_{\text{zengin}}|/10$)*Entropi(S_{zengin}) - ($|S_{\text{fakir}}|/10$)*Entropi(S_{fakir}) =0.811 - $(3/10)[-(2/3)log_2 2/3 - (1/3)log_2 1/3] - (1/10) * 0$ =0.811 - (3/10)[(2/3)*0.584 + (1/3)*1.584] - (1/10)*0 =0.536

B düğümü için para seçilir. Bu sebeble zengin ve fakir dalları olacaktır. $S_{\text{rüzgarlı, fakir}} = \{H7\}$. O zaman fakir dalı sinema yaprağına sahip. $S_{\text{rüzgarlı, zengin}} = \{H3, H8, H9\}$.

Table 2.4: Haftasonu Örneği Parça 3

Haftasonu Örneği	Hava	ebeveyn	para	karar
H3	rüzgarlı	evet	zengin	sinema
H8	rüzgarlı	hayır	zengin	alışveriş
H9	rüzgarlı	evet	zengin	sinema

$H3, H8, H9$ aynı kategoriye sahiptir. Hava ve para yukarısında kullanıldığı için son düğüm ebeveyn olmalıdır. $S_{\text{rüzgarlı, zengin, evet}} = \{H3, H9\}$. Bu yeni evet dalı için karar yaprağı sinema olmuştur. $S_{\text{rüzgarlı, zengin, hayır}} = \{H8\}$ diğer bir karar yaprağıdır ve kategorisi alışveriştir.

Son ana dal $S_{\text{yağmurlu}}$. $S_{\text{yağmurlu}} = \{H4, H5, H6\}$.

Table 2.5: Haftasonu Örneği Parça 4

Haftasonu	hava	ebeveyn	para	karar
H4	yağmurlu	evet	fakir	sinema
H5	yağmurlu	hayır	zengin	evde kalmak
H6	yağmurlu	evet	fakir	sinema

H4, H6 karar özniteliği sinemadır. H5 için karar evde kalmaktır. Farklı olduğundan bir C düğümü koyulur. Hava öncesinde kullanımıştır. Bu sebeble C için muhtemel öznitelikler $\{ebeveyn, para\}$.

Entropi($S_{yağmurlu}$)= - $[(2/3)log_2 2/3 + (1/3)log_2 1/3]$ = 0.917

kazanç($S_{yağmurlu,\ ebeveyn}$)= 0.917 - ($|S_{evet}|/10$)*Entropi(S_{evet}) - ($|S_{hayır}|/10$)*Entropi($S_{hayır}$) = 0.917 - (2/10)*0 - (1/10)*0 =0.917

kazanç($S_{rüzgarli,\ para}$) = 0.917 - ($|S_{zengin}|/10$)*Entropi(S_{zengin}) - ($|S_{fakir}|/10$)*Entropi(S_{fakir}) = 0.917 - (1/10)*0 - (2/10)*0 = 0. 917

kazanç($S_{yağmurlu,\ ebeveyn}$) = kazanç($S_{yağmurlu,\ para}$) olduğu için, C düğümü için ebeveyn ya da para özniteliklerinden hangisini kullandığı fark etmez. Ebeveyn seçilsin. $S_{yağmurlu,\ evet}$ = $\{H4, H6\}$. Yani evet dalının karar yaprağında sinema vardır. $S_{yağmurlu,\ hayır}$ = $\{H5\}$. Yani hayır dalının karar yaprağında evde kalmak vardır.

2.6 Lineer Ayırtaç Analizi (Linear Discriminant Analysis)

Lineer ayırtaç analizi 1936 yılında R.A. Fisher tarafından icat edilmiş bir sınıflandırma algoritmasıdır. Kısaca LDA denir.

Lineer ayırtaç analizininde amaç sınıfları ayıran bilgiyi mümkün olduğunca tutarken boyutu düşürmektir. Lineer ayırtaç analizi sınıflar arası saçılımı (between class scatter) maksimize ederken, sınıf içi saçılımı (within class scatter) minimize eder [Zhang (2004)].

Çok Sınıflı LDA (C-sınıf)

d boyutlu şu örneklemler (samples) $x^1, x^2, \ldots, x^N$ olsun. Bunlardan N_1 tanesi w_1 sınıfına, N_2 tanesi w_2 sınıfına, ..., N_C tanesi w_C ait olsun.

Bu örneklemlerin oluşturduğu matrise X diyelim. Yapılmak istenen X deki örneklemlerin $C-1$ boyutlu bir hiper düzlemin üstüne izdüşümünü alıp X i Y ye taşımaktır.

$C-1$ tane izdüşüm vektörü olan $w_1, w_2, \ldots, w_{C-1}$izdüşüm matrisi W nun sütünlarını oluşturur, öylel $\mathbf{w_i^T x},\ y = W^T X$. Burada X $d \times 1$, Y $C-1 \times 1$ ve W d $\times$ $C-1$ 'dir.

- Sınıflar içi saçılım S_W, $S_1 + S_2 + \ldots + S_C$toplamına eşittir.

$$S_i = \sum_{x \in w_i} (x - \mu_i)(x - \mu_i)^T \text{ and } \mu_i = \frac{1}{N_i} \sum_{x \in w_i} x.$$

- Sınıflar arası saçılım $S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^T$'dır. Burada $\mu = \frac{1}{N} \sum_{\forall x} x = \frac{1}{N} \sum_{i=1}^{C} N_i \mu_i$ olarak alınır.
- Toplam saçılım S_T, $S_B + S_W$ 'ya eşittir.
- İzdüşümü yapılmış örneklemler için, ortalama vektörü ve saçılım matrisi şöyledir:

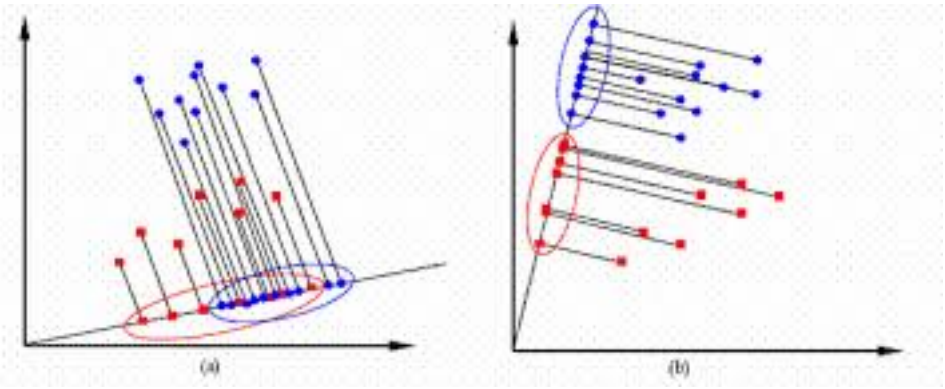


Figure 2.7: İzdüşüme göre sınıfların karışması (a) ve LDA in yaptığı sınıfları ayırma (b)

Kaynak: Gutierrez-Osuna (2005)

$$\widetilde{\mu_i} = \frac{1}{N_i} \sum_{y \in w_i} y.$$

$$\widetilde{\mu} = \frac{1}{N} \sum_{\forall y} y.$$

$$\widetilde{S_W} = \sum_{i=1}^{C} \sum_{y \in w_i} (y - \widetilde{\mu_i})(y - \widetilde{\mu_i})^T = W^T S_W W.$$

$$\widetilde{S_B} = \sum_{i=1}^{C} N_i (\widetilde{\mu_i} - \widetilde{\mu})(\widetilde{\mu_i} - \widetilde{\mu})^T = W^T S_B W.$$

LDA sınıflar arası saçılımın sınıf içi saçılıma oranını maksimize eder. İzdüşüm $C - 1$ boyutlu olduğu için saçılım matrislerinin determinanatları esas alınır. Yani amaç aşağıdaki hedef fonksiyonunu maximize eden W* izdüşüm matrisini bulmaktır [Hastie et al. (2009)].

$$J(W) = \frac{|\widetilde{S_B}|}{|\widetilde{S_W}|} = \frac{|W^T S_B W|}{|W^T S_W W|}$$

Optimal izdüşüm matrisi W* 'ın sütünları aslında $(S_B - \lambda_i S_W) W_i^* = 0$ özdeğer probleminin en büyük özdeğerlerine karşılık gelen özvektörlerdir [Hastie et al. (2009)].

2.7 Naive Bayes Sınıflandırma Algoritması (NB)

Bayesian sınıflandırma Bayes Teoremine dayanan bir gözeticili öğrenme metodudur. Naive Bayes algoritmasının bazı kullanımlarını metin sınıflandırmada, istemeyen e-posta süzgeçlemede, basit duygu modellemede görülür.

Bayes Kuralı [Mitchell (1999)] :

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}.$$

Gözlemlerin ve sınıf adlarının olduğu bir veri seti vardır.

Bayes Kuralını sınıflandırma problemine uygulayarak, şu yazalır :

$$P(\text{sınıf} \mid \text{gözlemler}) = \frac{P(\text{gözlemler} \mid \text{sınıf})P(\text{sınıf})}{P(\text{gözlemler})}.$$

$P($ sınıf $\mid$ gözlemler $)$ sonsal olasılık diye adlandırılır ve sonsal olasılık en büyük yapılmak istenir. $P($ gözlemler $\mid$ sınıf$)$ $P($sınıf$)$ olabilirlik diye adlandırılır ve $P($sınıf$), P($gözlemler$)$ önseller olarak adlandırılır. Sonsal olasılığı direk olarak hesaplamak genelde zordur. Bundan dolayı önseller ve olabilirlik kullanılarak hesaplanır.

Tüm öğrenme problemleri için amaç $P($sınıf$|$gözlemler$)$ olasılığını maksimize etmektir.

O zaman problem sonsal olasılığı, $P($sınıf$|$gözlemler$)$, maksimize eden sınıfı bulmaktır. **Her sınıf için sonsal olasılık hesaplanır ve sonsal olasılığı en yüksek yapan sınıf tahmin edilir** [Hastie et al. (2009)].

Bu problemin daha formal bir ifadesi:

$$\arg\max_{c \in \text{sınıf}} P(c|\text{gözlemler}) = \frac{\arg\max_{c \in \text{sınıf}} P(\text{gözlemler}|c)P(c)}{P(\text{gözlemler})}.$$

Burada sınıf tüm sınıf isimlerinin oluşturduğu kümeyi anlatır, c de o kümedeki herhangi bir sınıf ismini gösterir.

Bu probleme denk olan problem $\arg\max_{c\in\text{sınıf}} P(\text{gözlemler}|c)P(c)$ problemidir çünkü P(gözlemler) sınıfa bağlı değildir ve sabit bir sayıdır.

Eğer n tane gözlem $O_1, O_2, O_3, \ldots, O_n$ ise, $P(\text{gözlemler}|c)P(c)$ olasılığı $P(c, O_1, O_2, \ldots, O_n)$ olasılığıyla aynıdır.

$$\begin{aligned} P(c, O_1, O_2, \ldots, O_n) &= P(c)P(O_1, O_2, \ldots, O_n|c). \\ &= P(c)P(O_1|c)P(O_2, \ldots, O_n|c, O_1). \\ &= P(c)p(O_1|c)P(O_2|c, O_1)P(O_3, \ldots, O_n|c, O_1, O_2). \\ &= P(c)P(O_1|c)P(O_2|c, O_1)P(O_3|c, O_1, O_2)\ldots P(O_n|c, O_1, 0_2, 0_3, \ldots, \end{aligned}$$

Naive Bayes sınıflandırmanın genel varsayımı özniteliklerin bağımsız olduğudur., yani c sınıfı verildiğinde O_i ve O_j 'nin $i \neq j$ için birbirinden bağımsız olduğu varsayılır. Bu varsayım yardımıyla, şunu yazabiliriz :

$$\begin{aligned} &P(O_2|c, O_1) = P(O_2|c). \\ &P(O_2|c, O_1) = P(O_2|c). \\ &P(O_3|c, O_1, O_2) = P(O_3|c). \\ &P(O_4|c, O_1, O_2, 0_3) = P(O_4|c). \\ &P(O_n|c, O_1, 0_2, 0_3, \ldots, O_{n-1}) = P(O_n|c). \end{aligned}$$

Bu eşitlikleri kullanarak,

$$\begin{aligned} P(c, O_1, O_2, O_3, \ldots, O_n) &= P(c)P(O_1|c)P(O_2|c)P(O_3|c)\ldots P(O_{n-1}|c)P(0_n|c) \\ &= P(c)\prod_{i=1}^{n} P(O_i|c). \end{aligned}$$

yazılabilir.

Özetle, ilk basamak maksimizasyon problemini değişik sınıflar üzerinde şu şekilde yazmaktır :

$$\arg \max_{\text{c} \in \text{ sınıf}} P(C|\text{ gözlemler}) = \frac{\arg\max_{c \in \text{sınıf}} P(\text{gözlemler}|c)P(C)}{P(\text{gözlemler})}.$$

Ve son basamakda bu problemin daha iyi bir ifadesini Bayes kuralı ve özniteliklerin bağımsızlığı yardımıyla şu şekilde yazmaktır :

$$\arg \max_{c \in \text{ Class}} P(c) \prod_{i=1}^{n} P(O_i|c).$$

Naive Bayes sınıflandırıcın karar kuralı en olası hipotezi tercih etmektir. Naive Bayes sınıflandırıcı en olasılıklı, muhtemel sınıfı kazanan olarak açıklar. Yukardaki sonsal olasılığı maximize etme probleminin çözümü kazanan sınıftır ve en muhtemel sınıftır [Hastie et al. (2009)].

3. GÖZETİCİSİZ ÖĞRENME ALGORİTMALARI

3.1 K-merkezli Gruplama (K-Means Clustering)

K-merkezli gruplama en sade eğiticisiz öğrenme algoritmalarından bir tanesidir [MacQueen et al. (1967)].

K merkezli gruplama algoritması veriyi k parçaya bölümler. Burada k önsel sabit bir sayıdır.

K tane gruba böleceğimiz n tane nokta $x_1, x_2, \ldots, x_n$ olsun. Başlangıç basamağı k tane merkezi belirlemektir, yani her grup için bir merkez seçmiş oluruz. Gruplar için seçilen değişik ilk merkezler değişik gruplamalar ortaya çıkarır. Bu sebeble başlangıç merkezleri dikkatli seçilmelidir. Genel olarak başlangıç merkezlerinin birinden olduğunca uzak olması düşünülür. Sonraki basamak ise her bir data noktasına bir grup seçme işlemidir. K merkezli gruplama her bir veri noktasını en yakınındaki merkezin grubuyla ilişkilendirir. Bir data noktası x_i^j ve bir merkez c_j arasındaki uzaklık $||x_i^j - c_j||$ dir. Sonuçta k merkezli gruplama algoritmasının maliyet yada amaç foksiyonu :

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^j - c_j||^2.$$

K merkezli gruplama algoritması bu amaç foksiyonunu minimize etmeye çalışır ve k tane en son merkezi $c_1, c_2, \ldots, c_k$ bulmaya çalışır.

K merkezli gruplama algoritması şu şekilde çalışır:

1. İlk etapta k tane merkez $c_1, c_2, \ldots, c_k$ seçer.
2. Her bir veri noktasını kendisine en yakın merkezin grubuna atar.
3. Her bir grubun doğal merkezini o gruptaki elemanların ortalamasını alarak bulur. Yani bütün grupların merkezleri ortalama alınarak güncellenir.
4. **2. ve 3.** basamaklar merkezler değişmeyinceye kadar, yani yakınsama sağlanıncaya kadar tekrar edilir.

Algoritma kesin olarak yakınsar, fakat yakınsamanın maliyet fonksiyonunun local minimumuna olması da mümkündür.

K merkezli gruplama algoritmasının zayıf tarafları

- Başlangıçtaki ilk k merkezi seçmenin genel bir kuralı yoktur, genelde rastgele seçilirler.
- Seçilen grup sayısı k, makul olmalıdır. Uygun olmayan bir grup sayısı sürecin geçerliliğini (validity) azaltır. (Genel yaklaşım k mekezli gruplama algoritmasını değişik k sayıları için çalıştırıp en doğru k yı görmeye çalışmaktır.)
- Algoritmanın sonucu kullanılanan metriğe bağlıdır [Luke (2008)].

3.2 Temel Bileşen Analizi (Principal Component Analysis)

Son 30 yılda çok fazla sayıda gözlem ve her bir gözlemle ilgili öznitelik sayısının artması çoğu bilim dalında bilgi doluluğuna ve büyük veri setlerine sebeb olmuştur. Geleneksel yöntemler büyük boyutlu veriler için iyi sonuçlar vermemiştir. Fakat büyüt boyutlu verileri anlamak için ölçülen tüm öznitelikler önemli değildir. Yani büyük olasılıkla veriyi analiz etmemiz için gerekenden daha fazla öznitelik verilmiştir. Veri için herhangi bir modelleme yapmadan önce orjinal verinin boyutu düşürülmelidir. Boyut düşürme algoritmaları hızlandırır ve verinin daha az yer kaplamasını sağlar. Ayrıca boyut değiştirme sayesinde verinin görselleştirilmesi kolaylaşır [Fodor (2002)].

Bir gereğinden fazla (redundant) data örneği figure 3.1 de gösterilmiştir.:

Bu veri iki boyutludur. İlk boyut X_1 bir özniteliğin santimetre cinsinden ölçümüdür. İkinci boyut X_2 ise aynı özniteliğin inç cinsinden ölçümüdür. Bu iki şey aslında aynı özniteliğin ölçümü olduğundan, bu veri kümesini figure 3.2 de gösterildiği gibi tek boyutta ifade etmek mümkündür.

Yani bu data setinin gösterimi sadece z_1 boyutu ile yapılabilir.

Başka bir datanın boyutunu düşürme örneği, 3 boyuttan 2 boyuta, şudur :

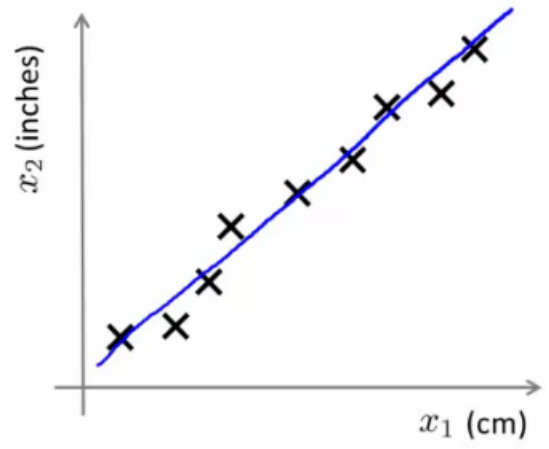


Figure 3.1: iki boyutlu basit bir veri
Kaynak: Ng (2000)

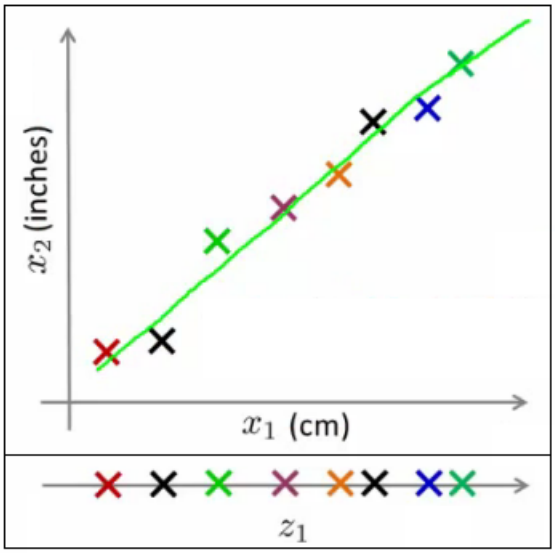


Figure 3.2: İki boyuttan bir boyuta indirerek aynı veriyi gösterme
Kaynak: Ng (2000)

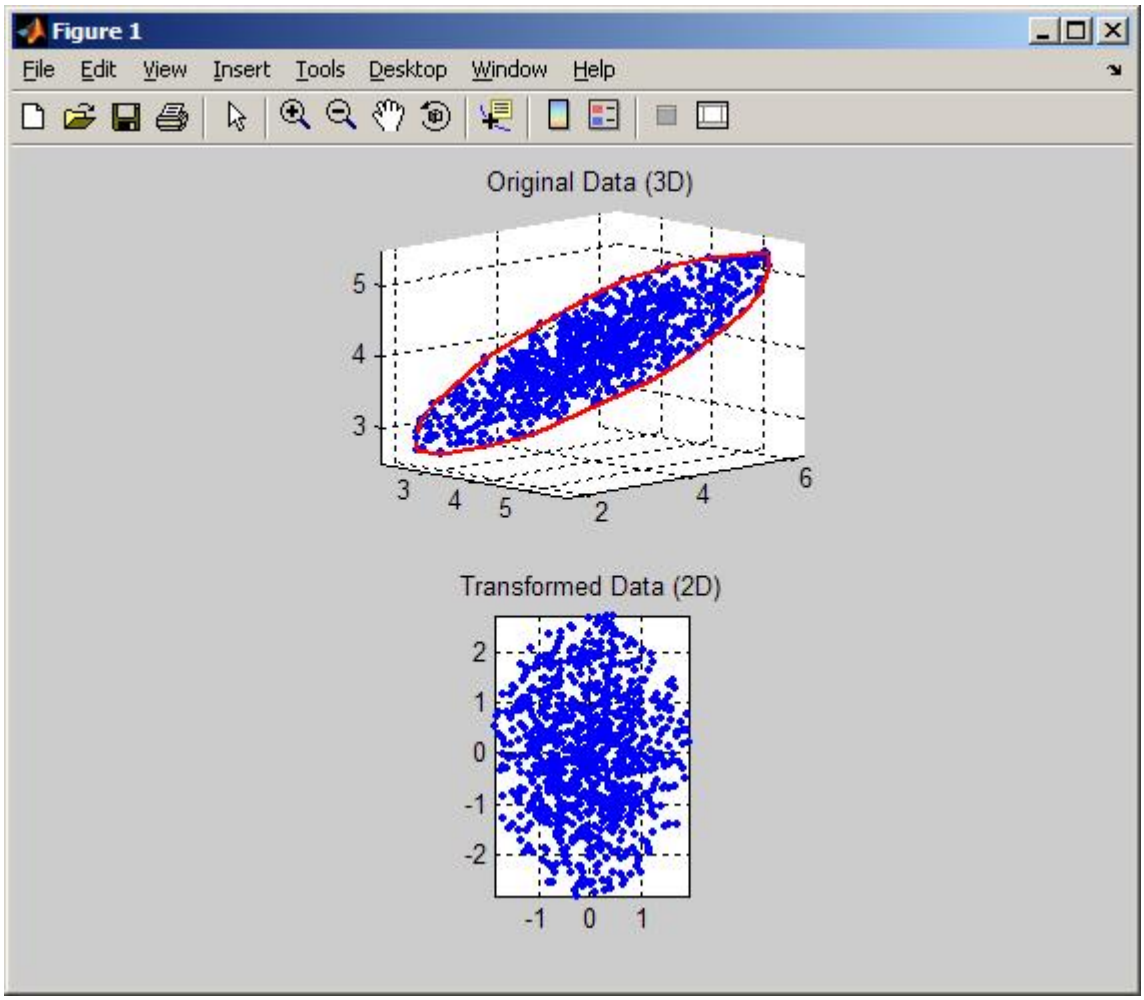


Figure 3.3: Üç boyutlu veriyi iki boyuta indirerek gösterme
Kaynak: Kleder (2005)

3.2.1 Temel Bileşen Analizi (PCA)

Boyut düşürme problemi için en çok kullanılan algoritma temel bileşen analizidir. Temel bileşen analizi **varyansı en yükek orjinal özniteliklerin** biribirine dik lineer birleşimlerini kullanarak boyutu düşürmeye çalışır. PCA in meşhur uygulamaları data görselleştirme, resim sıkıştırma, sayfaları derecelendirme (ranking), anlamsal indisleme (Latent Semantic Indexing) dir [Jackson (2005)].

PCA yüksek boyuttaki rastsallığı mümkün olduğunca muhafaza ederek boyutu düşürmeye çalışır. PCA tarafından modellenen düşük boyutlu alt uzay datadaki maximum çeşitliliği yakalar ve eşdeğişinti (covariance) yapısını modelliyor olarak düşünülebilir [Jolliffe (2005)].

3.2.2 Eşdeğişinti Matrisi

Eşdeğişinti verideki fazlalığı belirlemek için iyi bir yoldur. Eşdeğişinti değişkenler arasında bağımlılığı ölçer. Bir veri setinde aralarında güçlü istatistiksel bağımlılık olan değişkenler varsa bu fazlalığa (redundancy) işaret eder.

Eşdeğişinti iki değişken arasında ölçülür. Boyutu d olan bir veri kümesi için, $\frac{(d)!}{(d-2)!2!}$ tane eşdeğişinti değeri bulunur. Tüm bu eşdeğişinti değerlerini bir matrise yerleştirip, eşdeğişinti matrisi oluşturulur.

Eşdeğişinti matrisi :

C = $\frac{1}{N-1}X^TX$ burada X, N x d lik bir matrisdir.

Eşdeğişinti matrisi kare, simetrik, dxd lik bir matrisdir.

- $C_{i,j}$ = $\frac{1}{N-1}\sum_{q=1}^{N} X_{q,i}.X_{q,j}$.
- $C_{i,i}$ (diagonal) i değişkeninin varyansıdır. İlgilenilen varyansı büyük olan özniteliklerdir. Varyansı küçük olan özniteliklerin belirleyiciliği çok azdır.
- $C_{i,j}$ (diagonal dışı) i ve j arasındaki eşdeğişintidir. Diagonal dışındaki kısımlarda öznitelikler arasındaki korelasyonu gösterir. Diagonal dışı terimler büyükse, değişkenler arasındaki bağımlılık yüksektir ve bu verideki fazlalığa işaret eder. Diagonal dışındaki terimlerin küçük olması da bağımsızlığa ve veride fazlalık olmadığına işaret eder.

Eşdeğişinti matrisini **diagonal hale getirmenin** iki yolu :

1. Özdeğer ayrışımı (eigenvalue decomposition)
2. Tekil değer ayrışımı (SVD decomposition)

yollarıdır.

3.2.3 Özdeğer Ayrışımı

Eşdeğişinti matrisi ideal bir tabanda yazılabilir, yani diagonalize edilebilir. Bu yeni tabanda değişkenlerin en büyük varyansları sıralanır ve fazlalıklar atılır. Bu ideal taban veya doğru kordinatlar temel bileşenlerdir.

Eşdeğişinti matrisi kare ve simetrik olduğundan özdeğerleri reel ve farklıdır.

XX^T = $S\ \Lambda\ S^{-1}$ eşitliği doğrudur. Burada S XX^T nin özvektörlerinin matrisidir. XX^T simetrik olduğu için, özvektör sütunlar birbirine diktir. Böylece S birimsel (unitary) bir matris olarak yazılabilir ve $S^{-1} = S^T$. Burada Λ, XX^T un değişik özdeğerlerinin bir matrisidir.

Temel bileşenler tabanında çalışmak için, taşınmış şu değişkeni tanımlanır Y = $S^T X$.

Y nin eşdeğişintisi yazılır.

$$\begin{aligned}
\mathbf{C_Y} &= \frac{1}{N-1} Y^T Y \\
&= \frac{1}{N-1} (S^T X)(S^T X) \\
&= \frac{1}{N-1} S^T (X X^T) S \\
&= \frac{1}{N-1} S^T S \mathbf{\Lambda} S S^T \\
C_Y &= \frac{1}{N-1} \mathbf{\Lambda}.
\end{aligned}$$

Bu tabanda, temel bileşenler XX^T 'un özvektörleridir. C_Y 'nin j inci diagonal değeri X in x_j boyunca olan varyansıdır.

Özetle özdeğer ayrışımı yapılır ve orjinal verinin temel bileşenler üzerine izdüşümü elde edilir.

3.2.4 Tekil Değer Ayrışımı (SVD)

Her matris UΣV^* şeklinde çarpanlarına ayırılabilir. Burada U birimsel matrisdir, Σ diagonalı reel, negatif olmayan değerlerden oluşan bir matrisdir ve V^* birimsel bir matrisdir. SVD herhangi bir matrisi uygun U, V çiftini bularak diagonilize edebilir.

Taşınmış değişken Y = U^*X tanımlanır. Burada X = UΣV^* tekil değer ayrışımından elde edilen matrisdir.

$$\begin{aligned} C_Y &= \frac{1}{N-1} YY^T \\ &= \frac{1}{N-1} (U^*X)(U^*X)^T \\ &= \frac{1}{N-1} U^*(XX^T)U \\ &= \frac{1}{N-1} U^*U\Sigma^2UU^* \\ C_Y &= \frac{1}{N-1} \Sigma^2. \end{aligned}$$

Sonuçta $C_Y = \frac{1}{N-1}\Sigma^2 = \frac{1}{N-1}\Lambda$ doğrudur ve özdeğer metodu ile SVD arasındaki ilişki $\Sigma^2 = \Lambda$ olarak bulunur.

SVD daha sağlam bir metodtur ve temel bileşnleri bulmak için tercih edilmelidir.

3.2.5 PCA Algoritmasının Basamakları

PCA $\mathbf{Nxd}$ lık bir X veri matrisini $\mathbf{Nxk}$ lık bir Y matrisine taşır. $(k < d)$

Algoritmanın basamakları :

1. Önişleme : Ortalama normalleştirme (sütünların ortalamasını sıfırla) ve öznitelik ölçekleme (feature scaling)
2. dxd lik eşdeğişinti matrisi C = $\frac{1}{N-1}X^TX$ nin hesaplanması

3. Eşdeğişinti matrisinin özvektörlerinin SVD ile hesaplanması.

4. En yüksek k tane özdeğere karşılık gelen özvektörlerin yeni tabanı oluşturmak için seçilmesi. Bu k tane özvektörün $U_{\text{düşük}}$ isimli yeni matrisi oluşturması

5. Y matrisinin $U^T_{\text{düşük}} * X$ çarpımı sonucu bulunması

şeklindedir.

3.2.6 Temel Bileşen Sayısı k nın Seçilmesi

SVD tekil değerleri büyükten küçüğe doğru verir. Bu tekil değerlerin sırasına dikkatle bakılmalıdır. Bu sıralamada keskin düşüşün nerde olduğuna dikkat edilmelidir. Keskin düşüşten önceki tekil değerlerin sayısı **k** için makul bir tercih olabilir. **k** seçme konusunda ikinci bir yaklaşım daha vardır. Bu yaklaşıma göre baştan bir eşik λ_0 sabitlenir. Bu eşikten büyük olan özdeğerlerin özvektörleri temel bileşenler olarak tutulur. Ayrıca bu yaklaşımda en az 4 değişkenin tutulması tavsiye edilir. [Jolliffe (1972)]

3.2.7 Sıkıştırılmış Datadan Tekrar Yüksek Boyuta Dönüş

PCA k boyutlu sıkıştırılmış veriyi verir. Bu k boyutlu veriyi tekrar yüksek boyuta taşımak mümkündür. Yani yüksek boyutta orjinal verinin bir yaklaşığı $X_{\text{yaklaşık}}$ oluşturulabilir [Smith (2002)].

$X_{\text{yaklaşık}} = U_{\text{düşük}} * Y$. (PCA başında ortalama sıfırlama yapıldığı için, ortalamaları $X_{\text{yaklaşık}}$ 'a eklemek orjinal veriye daha yakınlaştırır.)

4. ANOMALİ TESPİT ETME TEKNİKLERİ

Veri kümesinde beklenen davranışlarla uyumlu olmayan örüntülere (pattern) anomali denir. Bu anomali tanımı kullanıldığı bağlama (context) göre değişir. Genel olarak veride normal olan bölger veya noktalar belirlenir, bunların dışında kalanlara anomali denir.

Anomali kelimesi yerine aykırı kelimesi de kullanılır. Aykırı (anomalous) olaylar nadir gerçekleşir. Sıklıkla gerçekleşmemelerine rağmen, gerçekleştirdikleri zaman oluşturdukları sonuçlar çok ciddi olduğu için anomali bulma önemli bir problemdir. Örneğin, beklendik şekilde olmayan MR görüntüsü kötü huylu bir tümörün varlığına işaret edebilir [Parra et al. (2003)] , kredi kartı harcama datasında aykırı bir harcama kart ya da kimlik hırsızlığına [Aleskerov et al. (1997)] işaret edebilir.

Veri de anomali tespit etme çalışması 19. yüzyılda istatistik topluluğu tarafından çalışılmaya başlanmıştır [Edgeworth (1887)]. Daha sonra değişik araştırma toplulukları tarafından bir çok anomali belirleme yöntemi geliştirilmiştir.

Anomali tespitinde karşılaşılan zorluklar şunlardır:

- Her normal davranışı yakalayan bir normal bölge tanımlamak zordur ve normal bölge ile anomali bölgesi arasında keskin bir sınır yoktur.
- Anomali kavramı uygulama alanına göre değişir. Bir alanda geliştirilen bir tekniği diğer bir alana taşımak çok kolay değildir.
- Çoğu zaman model kurmak için kullanılacak etiketlenmiş eğitim ve geçerleme (validation) verilerini bulmak zordur.
- Anomalilerin kötü niyetli fiillerin sonucu olduğu durumlarda, kötü niyetli rakipler anomalileri normalmiş gibi görünecek şekle sokabilirler. Bu da normal davranışı tanımlamayı daha da zorlaştırır.
- Çoğu çalışma alanında normal davranış evrildiği için başta yapılan normal tanımının temsili yeterli olmaz.

Anomali bulma problemi girdi veri,anomali tipi, çıktı veri , etiketlerin olup olmaması ve kullanılan anomali tespit tekniğinin değerlendirmesi bakımlarından incelenebilir [Chandola et al. (2009)].

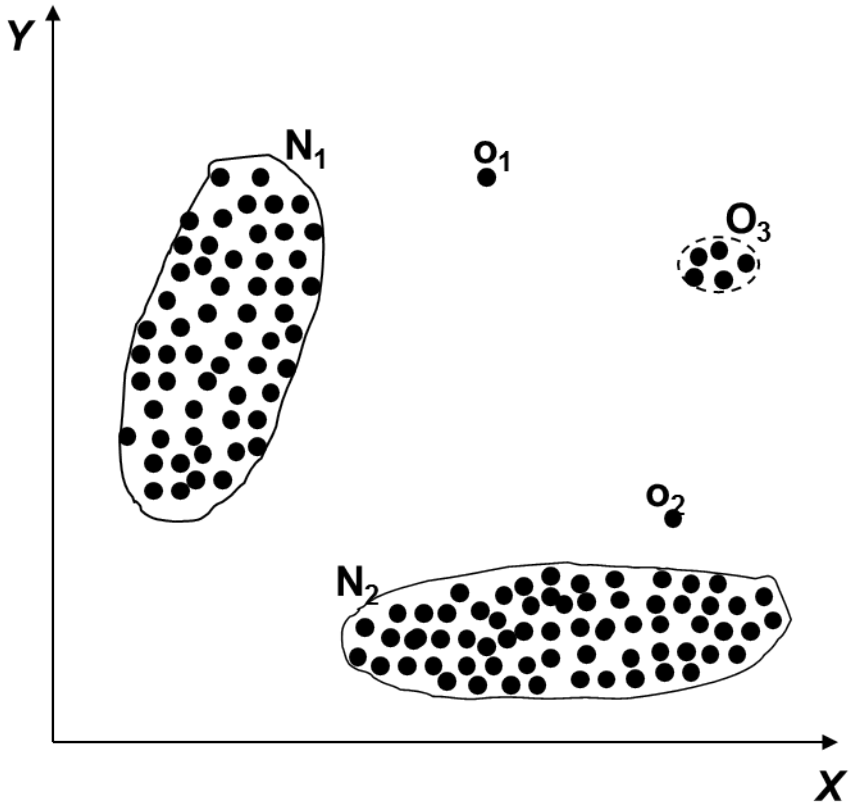


Figure 4.1: Noktasal anomaliler ve normal bölgeler
Kaynak : Chandola et al. (2009)

Girdi verisi çok öznitelikli ya da tek öznitelikli olabilir. Öznitellikler ikili (binary), kategorik, sürekli, karma (hybrid) olabilir. Değişkenler arasında zamana, konuma göre bir ilişki ya da çizgesel bir ilişki olabilir. Data tam etiketli, yarı etiketli ya da etiketsiz olabilir. Değişkenler arasında zamana, konuma göre bir ilişki ya da çizgesel bir ilişki olabilir [Tan et al. (2005)]. Girdi verisinin özellikleri anomali tespit tekniğinin uygulanabilirliğini belirler.

Noktasal, bağlamsal, ve toplu anomali olmak üzere 3 tip anomali vardır. Tek tek noktaların datanın diğer kalan kısımlarına göre farklı olması noktasal anomali oluşturur, örneğin yukardaki şekilde O_1 ve O_2 noktasal anomalidir. Bir data örneğinin anomali olması bulunduğu bağlama (context) bağlıysa, buna bağlamsal anomali denir. Bir grup data örnekleri sadece toplu halde bulunduğunda anomali oluşturabiliyorsa, bu tip anomaliye toplu (collective) anomali denir.

Anomali tespit etmede çıktı etiket olarak verilebilir. Bu tarz çıktıda yeni verilen örnek anomalidir ya da normaldir diye kategorilenir. Diğer çıktı verme şeklide yeni örnek için anomali skorunu vermektir. Eğer bu skor belli bir eşikten büyükse anomali, küçükse normal diye düşünülür. (İkinci yaklaşımda değişik eşikler için sonuçlar karşılaştırılır.)

Anomali tespit etme metodlarının değerlendirilmesinde doğruluk (accuracy) kullanılmaz. Anma (Recall), kesinlik (Precision), ve F-ölçütü (F-measure) tercih edilir [Nyalkalkar et al. (2011)]. Burada **anma** doğru şekilde tespit edilen anomali sayısının toplam anomali sayısına oranıdır. **Kesinlik** doğru şekilde tespit edilen anomali sayısının anomali olarak tahmin edilen toplam örnek sayısına oranıdır. F-ölçütü nün tanımını anma ve kesinlik kulanarak şöyle yaparız :

$$\text{F-ölçütü } = 2 \times \frac{\text{Anma } \times \text{ kesinlik}}{\text{Anma } + \text{ kesinlik}}$$

Anma ile kesinlik arasındaki ödünleşim (trade-off) bu F-ölçütü sayesinde ayarlanır. Bu F-ölçütü anomalilerin doğru tespitinin verilen bir ϵ eşiği için ne kadar iyi yapıdığını söyler.

Anomali tespitinin bazı uygulamaları şunlardır :

- Bilgisayar sistemlerinde kötü niyetli eylemlerin tespiti [D'haeseleer et al. (1996)]
- Kredi kartı, sigorta poliçesi gibi ticari ürünlerle ilgili suç teşkil eden işlemlerde sahtecilik tespiti [Aleskerov et al. (1997)]
- Tıpta hasta kayıtları datasındaki anomalilerin ve halk sağlığıyla alaklı anomalilerin tespiti [Horn et al. (2001)]
- Metin madenciliğinde (text mining) yeni konu, olay ya da hikayelerin tespiti [Baker et al. (1999)]
- Fotoğraf analizi [Byers and Raftery (1998)]

Günümüzde hem gözeticili hem de gözeticisiz öğrenme algoritmaları anomali tespitinde kullanılır.

4.1 Sınıflandırma Tabanlı Anomali Tespiti

Sınıflandırma temeline dayanan anomali tespiti algoritmaları iki fazda çalışır. İlk fazda sınıflandırıcı etiketli veri üstünde öğrenir, yani model kurulur. Model eğitim verisinnde çokça bulunan normal noktalar ve az rastlanan aykırı noktalar için kurulur. ikinci fazda yeni test örneği normal veya anomali diye sınıflanır.

Sınıflandırma temelli anomali tespit teknikleri çok sınıflı ve tek sınıflı olarak iki kategoride düşünülebilir. Çok sınıflılarda eğitim verisi birden fazla normal sınıfa ait örnekler içerir [De Stefano et al. (2000)]. Bu tarz teknikler her bir sınıflandırıcıya bir normal sınıfla kalan her şey arasındaki farkı öğretir. Yeni gelen bir örnek hiç bir sınıflandırıcı tarafından seçilmesse, anomalidir. Tek sınıflılarda tek bir normal sınıf olduğu düşünülür ve bu normal sınıfın sınırları öğrenilir. Bu sınırların dışına düşen örnekler anomalidir [Ratsch et al. (2002)].

Bir sinirsel ağ (neural network) değişik normal sınıfları eğitim verisi üzerinde öğrenir. Sonrasında yeni gelen test örneğini sinirsel ağ kabul ediyorsa, bu test örneği normal bir noktadır. Kabul edilmeyen test noktası anomalidir [De Stefano et al. (2000)]. Destek vektör makineleri normal bölgenin sınırlarını öğrenir. Eğer yeni gelen test örneği bu sınırların dışındaysa anomali olarak açıklanır. DVM tek sınıflı anamali problemlerinde kullanılmıştır [Ratsch et al. (2002)]. Sistem aramalarına dışardan müdahale tespitinde [Eskin et al. (2002)], görsel sinyal verisinde anomali tespitinde [Rabaoui et al. (2007)] ve benzeri çalışmalarda DVM yardımıyla anomali tespiti yapılmıştır.

Karar verme kuralı oluşturma fikrine dayanan sınıflandırma algoritmaları normal davranışı ifade edecek kuralı öğrenirler. Eğer bir test noktası bu kurallardan herhangi birinin kapsamında değilse anomalidir. Kural bulmaya dayalı çok sınıflı anomali tespitinde **ilk basamak karar ağacı ve benzeri öğrenme algoritmaları yardımıyla kuralları çıkarmaktır.** Her kuralla ilişkili bir güven değeri vardır. Kuralın doğru sınıflandırdığı eğitim örneklerinin kuralın sınıflandırdığı tüm eğitim örneklerine oranıyla bu güven değeri doğru orantılıdır. **İkinci basamak her bir test örneğini en iyi kapsayan kuralı bulmaktır.** Bir test örneği için en iyi olan kuralın güven değerinin çarpmaya göre tersi, test örneğinin anomali skoru olur. Kural çıkarmaya dayalı teknikler ağlara müdahala tespitinde (network intrusion) [Mahoney and Chan (2002)], sistem aramalarına müdahale tespitinde [Lee et al. (2000)] ve kredi kartı sahteciliği tespitinde [Yairi et al. (2001)] kullanılmıştır.

4.2 En Yakın k Komşu Algoritmasına Dayalı Anomali Tespiti

Normal veri noktalarının yoğun komşulukta olması, anomalilerinse en yakın komşularına bile oldukça uzak olması varsayımına dayanır.

En yakın komşuluğa dayalı anomali tespiti uzaklık ölçümü gerektirir. Sürekli öznitelikler için genel olarak Öklid metriği kullanılır ve değişik uzaklık ölçümleri de kullanılabilir [Tan and Steinbach (n.d.)]. Kategorik öznitelikler için genelikle basit eşleştirme katsayısı kullanılır ve daha karmaşık uzaklık ölçümleri de kullanılabilir [Chandola et al. (2009)]. Çok değişkenli veri örneklerinde önce her bir değişken için uzaklık hesaplanır ve sonra bu uzaklıklar birleştirilir [Tan and Steinbach (n.d.)].

En yakın komşuluğa dayali anomali tespitini ikiye ayırabiliriz :

- En yakın k. komşuya olan uzaklığı anomali skoru olarak alan teknikler
- Her bir data noktasının görece yoğunluğunu anomali skorunu bulmak için hesaplayan teknikler

En Yakın k. Komşuya Olan Uzaklığa Dayalı Teknikler

Bu teknikler bir veri noktasının anomali skorunu, o noktanın en yakın **k.** komşusuna olan uzaklığı olarak tanımlar. Bu temel teknikte, bir test noktası anomali mi değil mi diye bakarken, anomali eşiği de belirtilmelidir. Anomali skoru bu eşikten büyük olan veri noktaları anomali olarak seçilir. Diğer bir karar verme şekli de en yüksek anomali skoruna sahip n tane noktaya anomali demektir [Ramaswamy et al. (2000)].

Üst parağrafta bahsettiğimiz teknik anomali skorunun tanımı değiştirilerek, kullanılan uzaklık ölçümü değiştirilerek ya da verimlilik arttırılarak (karmaşıklık O(N^2) değiştirilerek) modifiye edilebilir.

Veri noktasının en yakın k tane komşusuna olan uzaklıkları ayrı ayrı hesaplanarak toplanmıştır ve bu toplam anomali skoru olarak alınmıştır [Eskin et al. (2002)]. Akran grubu analizinde benzer bir anomali skoru tanımı esas alınmıştır [Bolton et al. (2001)].

Bir veri noktasına uzaklığı d mesafesinden küçük olan komşuların sayısı kullanılarak da anomali skoru tanımlanmıştır [Knorr and Ng (1997)]. Bahsettiğimiz komşu sayısı n ise,

anomali skoru $\frac{1}{n}$ olarak alınır. Başka bir yaklaşım da komşu sayısı n i elde etmek için gereken uzaklık d ye bakmaktır. Bu durumda anomali skoru d olarak alınır.

Verinin öznitelikleri sürekli olmasa bile, temel en yakın k komşuluk tekniği kullanılabilir. Kategorik data için hiper çizgelere (graph) dayanan bir teknik önerilmiştir [Wei et al. (2003)] Bu teknikte kategorik değerler hiper çizge kullanılarak modellenmiştir. İki data örneği arasındaki uzaklık çizgedeki bağlantısıllıklar (connectivity) analiz edilerek ölçülür. Kategorik ve sürekli özniteliklerin karışık olduğu datalar için de uzaklık ölçümü tanımlanmıştır [Otey et al. (2006)] İki data noktası arasındaki mesafe sürekli öznitelikler arasındaki uzaklığa kategorik öznitelikler arasındaki uzaklıklar katılarak elde edilir. Kategorik öznitelikler arasındaki mesafede kaç tane özniteliğin aynı değeri aldığı belirleyicidir.

Verimliliği artırmak için temel en yakın komşuluk tekniğinin değişik varyasyonları üretilmiştir. Bazı teknikler anomali olamayacak veri örneklerine hiç bakmaz ve anomali olma ihtimali yüksek olan veri örneklerine odaklanır. Ayrıca, yeni bir veri örneği için anomali skoru bulunduğunda, o ana kadar bulunan anomalilerin skorlarından en küçüğünü eşik olarak almak iyi bir fikirdir [Bay and Schwabacher (2003)].

Görece Yoğunluğa Dayalı Teknikler

Her bir veri noktasının komşuluğunun yoğunluğuna bakmaya dayanır. Bu yoğunluk düşükse nokta anomalidir, yüksekse normaldir.

Bir veri noktasının k. komşuluğuna olan uzaklık o veri noktası merkezli hiper kürenin yarıçapı olarak düşünülür. Bu yarıçapın çarpmaya göre terside anomali skoru olarak alındığında yoğunluk tabanlı yaklaşımın aslında temel en yakın k komşuluk yaklaşımıyla örtüştüğü görülür.

Yoğunluğa dayalı teknikler değişik yoğunluklu bölgelere sahip veri kümeleri için yeterince iyi sonuçlar vermez. Bu tarz veri kümelerinde iyi sonuçlar almak için bazı teknikler geliştirilmiştir. Bunlardan ilki **Yerel Anomali Faktörüdür (LOF)** [Breunig et al. (2000)]

Lof skoru veri noktasının en yakın k komşusunun ortalama yerel yoğunluğu ile veri noktasının kendisinin ortalama yerel yoğunluğu oranı olarak tanımlanmıştır. Burada herhangi bir noktanın yerel yoğunluğunu bulmak için bu veri noktası merkezli ve en yakın k tane komşuyu içeren en küçük hiper kürenin yarıçapı bulunmalıdır. Bu noktanın yerel yoğunluğu $\mathbf{k}$ yı bu kürenin hacmine bölerek bulunur. Yoğun bölgede bulunan normal bir

noktanın yerel yoğunluğu komşularının yerel yoğunluğuna yakın olacaktır. Bir anomali için yerel yoğunluk komşularınınkine göre oldukça düşük olacaktır.

LOF un biraz değişmişi, **Bağlantısallık Temelli Anomali Faktörü (COF)** Tang tarafındandan üretilmiştir [Tang et al. (2002)]. COF bir veri noktası için k tane komşuyu artımlı (incremental) seçer. Bu LOF tan farklılaştığı kısımdır. Yani en başta veri noktasına en yakın nokta komşuluk kümesine eklenir. Kalan noktalar arasında komşuluk kümesine en yakın nokta komşuluk kümesine eklenir. Yeni gelecek noktalar bu şekilde sırayla komşuluk kümesine yakınlığına göre kalan noktalar arasından seçilir. Bu şekilde ekleme işlemine komşuluk sayısı k olunca son verilir. Anomali seçme basamağı LOF la aynıdır.

LOF un başka bir değişmişi de **Çok tanecikli Sapma Faktörü (MDEF)** tir. Bir veri noktası için MDEF o nokta ve en yakın komşularının yerel yoğunluklarının standard sapmasıdır. Bu standard sapmanın çarpaya göre tersi anomali skoru olarak alınır [Papadimitriou et al. (2003)].

4.3 Gruplamaya Dayalı Anomali Tespiti

Gruplama temelli anomali tespiti şu fikirlere dayanır :

1. Anomaliler bir gruba ait değillerdir yada küçük gruplar oluştururlar.
2. Normal veri noktaları kendilerine en yakın olan grubun merkezine yakındırlar. Anomaliler kendilerine en yakın olan grubun merkezine epey uzaktırlar.
3. Normal veri noktaları büyük ve yoğun gruplara aittir, anomaliler küçük ve seyrek gruplara aittir.

DBSCAN gibi her veri örneğini bir gruba ait olmaya zorlamayan gruplama algoritmaları yukardaki birinci fikre dayanır [Ester et al. (1996)].

İkinci fikirden kaynaklanan gruplama algoritmalarında, veri noktasının kendisine en yakın olan grup merkezine olan uzaklığı anomali skoru olarak hesaplanır. Bu fikre dayanarak k-merkezli gruplama ve Özörgütlemeli haritalar (Self-Oganizing Maps) anomali tespiti uygulamalarında kullanılmıştır [Kohonen and Maps (1995)].

Eğer anomali topluluğu bir grup oluşturuyorsa, üçüncü fikirden yararlanılarak tespit etmek daha iyidir. Bu yaklaşımda bir grubun büyüklüğünün ve yoğunluğunun verilen alt eşiklerden büyük olup olmadığına bakılır. Değilse, anomali olarak düşünülür. **C**BLOF, gruplama temelli yerel anomali faktörü, tekniği bu yaklaşıma dayanarak ortaya konmuştur [He et al. (2003)]. CBLOF skoru veri örneğinin hem ait olduğu grubun büyüklüğünü hem de grubun merkezini dikkate alarak hesaplanır.

Gruplama temelli ve en yakın k komşuluk metodu temelli anomali tespit algoritmaları farklıdır. Gruplama temelli anomali tespitinde her bir örnek kendi grubuna göre değerlendirilir. En yakın k komşuluk temelli anomali tespitinde her bir örnek kendi yerel komşuluğuna göre değerlendirilir.

4.4 Bilgi Teorisi (Information Theory) Temelli Anomali Tespiti

Verinin bilgi hacmi (information content) bu yaklaşımda belirleyicidir. Bilgi teorisine dayanan yaklaşımdaki varsayım anomalilerin veri setindeki toplam düzensizliği ve bilgi hacmini dikkat çekici biçimde değiştireceğidir. Yani anomali bulmak demek veri setinin bilgi hacminde (karmaşıklığında) ciddi oynama yapan örnekleri bulmak demektir.

Bir veri kümesi olarak D yi alalım ve $C(D)$ bu veri kümesinin karmaşıklığını (complexity) göstersin. Yukarda bahsettiğimiz bilgi hacmini ya da karmaşıklığı ciddi değiştiren örnekler kümesini bulmak aslında $C(D) - C(D - I)$ ı en büyük yapan en küçük **I** örnekler kümesini bulma problemidir. Burada D nin karmaşıklığını ölçmek için değişik yollar vardır : Kolmogorov karmaşıklığı [Li and Vitányi (2009)], sıkıştırılmış verinin boyutu [Keogh et al. (2004)] , entropi bu yollardan en yaygın kullanılanlardır.

4.5 İstatistik Temelli Anomali Tespiti

Bu yaklaşımda dataya bir olasılıksal model oturtulur ve anomali bu modelle çok ilgisiz olan, bu modelle ifade edilmeyen veri noktasıdır [Anscombe (1960)]. İstatiksel anomali tespit metoduna göre normal veri noktaları modelin yüksek olasılıklı dediği bölgelerde olur, anomaliler modelin düşük olasılıklı dediği bölgelerde olur.

İstatistiksel teknikler veriye (genelllikle normal örnekler için) bir model oturtur. Sonrasında yeni gelen örnek noktanın modele ait olup olmadığını anlamak için istatiksel çıkarım testleri uygulanır. Bu testler yeni örnek noktanın eldeki modelden çıkma olasılığı düşüktür derse, yeni nokta anomali olarak değerlendirilir.

4.5.1 Parametrik Teknikler

Bu tekniklerde normal verinin parametesi θ olan ve dağılım fonksiyonu $f(x, \theta)$ olan bir model tarafından üretildiği düşünülür. Bir test noktası x için, $f(x, \theta)$ nın tersi anomali skoru olarak alınabilir.

Bir diğer yolda hipotez testleri kullanmaktır. H_0 verilen yeni noktanın normal nokta olduğunu ve modelden üretildiğini varsayar. Eğer istatistiksel testler sonunda H_0 reddedilirse, yeni nokta anomali olarak düşünülür.

Eğer veri normal dağılımdan üretilmişse, parametreler en büyük olabilirlikle (EBO-MLE) bulunur. Herhangi bir veri noktasının beklenen ortalamaya uzaklığı anomali skoru olarak alınabilir. Bu skorun anomali eşiğine göre durumu, örneğin anomali olup olmadığını söyler. Ortalama μ den uzaklığı 3σ dan fazla olan noktaları anomali olarak almak sık kullanılan bir tekniktir [Shewhart (1931)].

Tıbbi bölge verisinde anomali tespiti için kutu çiziş kuralı (box plot rule) kullanılmıştır [Laurikkala et al. (2000)]. Anomali olmayan en küçük örneği (minimum), alt çeyreği, medyanı, üst çeyreği ve anomali olmayan en büyük örneği (maximum) kutu grafiksel olarak gösterir. Üst çeyrek ve alt çeyrek arası mesafeye çeyrekler açıklığı (inter quartile range) denir. Alt çeyrekten 1.5 çeyrekler açığı kadar aşağıda olan veya üst çeyrekten 1.5 çeyrekler açığı kadar yukarda olan noktalar anomali olarak alınır.

Grubb testi tek değişkenli verilerde anomali tespiti için kullanılır [Grubbs (1969)]. Bu test verinin normal dağılımlı olduğunu varsayar ve her bir test noktası $\mathbf{x}$ için $\mathbf{z}$. skoru hesaplanır.

$$z = \frac{|x - mean|}{|standarddeviation|}$$

Bir test noktası x için aşağıdaki şart sağlanırsa anomalidir diyebiliriz :

$$z > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N-2+t^2_{\alpha/(2N),N-2}}}.$$

Burada N verinin boyutu, $t^2_{\alpha/(2N),N-2}$ anomali eşiği, $\frac{\alpha}{2N}$ de anlamlılık (significance) seviyesidir. Anlamlılık seviyesi anomali eşiğinin güvenilirliğini belirlediği için aslında anomali olacak toplam örnek sayısını da belirler.

İşletim sistemi arama verisinde χ^2 istatistiği kullanılarak anomali tespiti yapılmıştır [Ye and Chen (2001)]. Anomali olmayan örneklerin çok değişkenli normal dağılıma sahip olduğu varsayılır.

$$\chi^2 = \sum_{i=1}^{N} \frac{(X_i - E_i)^2}{E_i} \text{ bulunur.}$$

Burada X_i $i.$ değişkenin değeri, E_i $i.$ değişken için beklenen değer, n de toplam değişken sayısıdır. χ^2 in değerinin yüksek çıkması veri setinde anomali olduğuna işarettir.

Regresyon tekniği de anomali tespitinde kullanılmıştır. Regresyonun ilk basamağı regresyon modelini veriye oturtmaktır. İkinci basamakta her bir test noktası için artık (residual) belirlenmesidir. Artık, test noktası için regresyon modeli tarafından açıklanmayan kısımdır ve anomali skoru olarak alınabilir [Anscombe (1960)].

Eğitim verisinin anomali içermesi regresyon parametrelerini etkiler ve modelin güvenilirliğini azaltır. Regresyon modelini veriye oturturken anomalilerin etkisini azaltmak için sağlam (robust) regresyon tekniği kulanılmıştır [Rousseeuw and Leroy (2005)].

Zaman serisi verilerinde anomali tespiti için regresyon kullanılmıştır [Fox (1972)]. Çok değişkenli zaman serilerinde anomali tespiti için değişik istatistikler üretilmiştir [Tsay et al. (2000)].

Bir diğer yaklaşımda veriyi modellemek için değişik istatiksel metodların karışımını kullanmaktır. Örneğin normal veri $N(0, \sigma^2)$ olarak, anomaliler aynı ortalama fakat yüksek varyansla $N(0, k^2\sigma^2)$ olarak modellenmiştir [Abraham and Box (1979)]. Sonrasında bu iki dağılım üstünde Grubb testi yardımıyla yeni noktanın normal mi anomali mi olduğuna karar verilmiştir.

Verideki normal örnekler ve anomaliler için ayrı ayrı modeller kurup daha sonra bu modelleri birleştirmek diğer bir yoldur [Eskin (2000)]. Her bir veri noktasının anomali olmasının önsel olasılığı λ, normal olmasının önsel olasılığı $1 - \lambda$ alınmıştır.

$\mathbf{D}$ tüm verinin dağılımını, $\mathbf{A}$ anomalilerin dağılımını, $\mathbf{M}$ normal noktaların dağılımını gösterdiğinde,

$$\mathbf{D} = \lambda\mathbf{A} + (1 - \lambda)\mathbf{D}$$

modeli kurulur. Bir test noktası $\mathbf{M}$ den çıkarılıp $\mathbf{A}$ ya eklendiğinde dağılımlardaki değişme miktarı o test noktasının anomali skorunu da verir.

4.5.2 Parametrik Olmayan Teknikler

Parametrik olmayan tekniklerde modelin yapısı baştan tanımlanmamıştır. Bu teknikler parametrik olanlara göre veri hakkında daha az varsayımda bulunurlar.

Histogram temelli (aynı zamanda sıklığa dayanan da denir) anomali tespiti basit parametrik metodlardandır. Histogram temelli teknikler saldırı tespitinde [Eskin (2000)], sahtecilik tespitinde [Fawcett and Provost (1999)] kullanılmıştır. Genelde histogramlar normal data için kurulur [Anderson et al. (1995)].

Tek değişkenli veriler için özniteliğin aldığı değişik değerlere göre histogram kurulur. Sonraki basamakta yeni gelen test noktası histogramın yan yana kutucuklarından birine düşüyormu diye bakılır. Eğer düşüyorsa normal bir noktadır, düşmüyorsa anomalidir. Her bir noktaya ait olduğu kutunun yüksekliğine göre (yani grubunun sıklığına göre) anomali skoru atanır. Histogram kurulurken kutucukların taban genişliği dikkali seçilmelidir. Eğer genişlik çok küçük olursa, bir çok normal nokta boş yada yüksekliği az olan kutucuğa düşebilir. Eğer genişlik çok büyük olursa bir çok anomali nokta yüksekliği fazla olan kutucuklardan birine düşebilir. Bu iki tip yanlışlığı dengeleyecek ideal bir taban genişliği seçilmelidir.

Çok değişkenli verilerin her bir özniteliği için ayrı histogramlar yapılır. Bu histogramlardaki kutuların yüksekliğine bakılarak özniteliklerin değerlerine dair anomali skorları

bulunur. Sonrasında test örneğinin özniteliklerinin değerleri için anomali skorlarının birleştirilmesiyle test örneğinin anomali skoru elde edilir. Sistem aramalarına müdahale tepitinde [Ghosh et al. (1999)] ve yapılardaki zarar tespitinde [Manson et al. (2001)] çok değişkenli veri için histogram temelli anomali bulma yöntemi kullanılmıştır.

4.6 İzgesel (Spectral) Anomali Tespiti

İzgesel anomali tespitinin altındaki varsayım normal noktaların ve anomalilerin veri daha küçük boyutlu bir uzaya taşındığında oldukça farklı görüneceğidir. Amaç da bu daha küçük boyutlu uzayı bulup anomalilerin tespitini orada kolayca yapmaktır [Agovic et al. (2008)].

Çoğu izgesel anomali tespit tekniği veriyi daha küçük boyutlu uzaya taşımak için temel bileşen analizi (PCA) kullanır [Jolliffe (2005)]. Temel bileşen analizindeki en büyük özdeğerlere karşılık gelen bir kaç temel bileşen normal verideki değişkenliği tutar. Normal veri için küçük özdeğerlere karşılık gelen temel bileşenler hemen hemen sabit değerlidir. Anomali tespit etme tekniklerinden bir tanesi bu varyasyonu düşük temel bileşenlere her bir veri noktasının izdüşümüne bakar. Normal noktaların izdüşümü küçüktür, anomaliler için bu izdüşümü büyüktür [Spence et al. (2001)]. Bu teknik astronami kataloglarında anomali tespitinde kullanılmıştır [Dutta et al. (2007)].

Çizgelerin zaman serisinde anomali tespiti için izgesel teknikler kullanılmıştır [Idé and Kashima (2004)]. Her bir zaman için bir çizge (graph) vardır ve bu çizgeler bitişiklik matrisleriyle (adjacency matrix) gösterilir. Her bir zamanın matrisine bakılır ve bu matrisin en büyük temel bileşeni aktivite vektörü olarak seçilir. Değişik zamanların aktivite vektörlerinin zaman serisi matris olarak düşünülür. Bu matrisin temel sol tekil vektörü verideki normal bağımlılığı ölçmek için kullanılır. Temel sol tekil vektörü ile yeni gelen test çizgesinin aktivite vektörü arasındaki açı bu test çizgesi için anomali skoru bulmada kullanılır.

Normal eğitim verisinin eşdeğişinti (covariance) matrisinin temel bileşenlerinin sağlam temel bileşen analizi yapılarak (robust PCA) bulunduğu anomali tespit yöntemi Shyu tarafından geliştirilmiştir [Shyu et al. (2003)]. Bu yöntemde her bir noktanın temel bileşenlere olan uzaklığına bakılarak anomali skoru belirlenir. Yani bir $\mathbf{x}$ noktasının $\lambda_1, \lambda_2, \ldots, \lambda_n$ özdeğerlerine karşılık gelen temel bileşen vektörlerine uzaklığı sırasıyla $y_1, y_2, y_3, \ldots, y_n$

olduğunda

$$\sum_{i=1}^{k} \frac{y_i^2}{\lambda_i}, k \leq n$$

ki-kare dağılıma sahiptir [Hawkins (1974)].

Veri noktası x ve anlamlılık derecesi α için eğer

$$\sum_{i=1}^{k} \frac{y_i^2}{\lambda_i} > \chi_q^2(\alpha)$$

sağlanıyorsa x anomali olarak alınır [Shyu et al. (2003)].

5. TARTIŞMA

Sınıflandırmaya dayanan anomali tespit yöntemleri değişik, güçlü algoritmalar kullanarak veriyi bilinen sınıflara bölebilirler ve bu yöntemlerin test aşaması hızlıdır. Etiketli data gerektirmeleri ve çıktı olarak sadece sınıf vermeleri (anomali skoru vermemeleri) dezavantaj olarak görülür.

En yakın k komşuluğa dayanan tekniklerin avantajı veri hakkında bir varsayım yapmamaları ve elde bir uzaklık ölçümü varsa her veriye uygulanabilmeleridir. Dezavantajları da uzaklık ölçümüne ciddi şekilde bağlı olmaları, test aşamasında çok fazla uzaklık hesabı gerektirmeleri, ve verideki normal noktaların az komşusu olduğunda yada anomalilerin çok komşusu olduğunda yanlış sonuçlar vermeleri olarak sıralanır.

Gruplamaya dayanan anomali tekniklerinin avantajları gözeticisiz çalıştığı için etiketli veri gerektirmemesi, karmaşık veri çeşitlerine adapte edilebilmesi ve test aşamasının hızlı olmasıdır. Dezavantajları ise sonucun gruplama algoritmasının veri içindeki normal yapıları doğru keşfetmesine bağlı olması, çoğu gruplama algoritmasının anomalileri gruplama sonucu kalan artık noktalar olarak ele alması ve anomaliler kendi aralarında bir grup oluşturduğunda algoritmanın onları normal gibi değerlendirmeye yatkın olması olarak sıralanabilir [Chandola et al. (2009)].

Bilgi teorisine dayanan teknikler verinin istatistiksel dağılımı hakkında varsayım yapmadıkları için avantajlıdır. Sadece veride yüksek sayıda anomali olduğu durumlarda kullanışlı olmaları ve sonuçların kullanılan ölçüme bağlı olması (entropi gibi) dezavantajdır.

İzgesel anomali tepit teknikleri yüksek boyutlu verilerin boyutunu düşürüp onları başka algoritmalar da uygulanabilir hale getirdikleri için kullanışlıdır. Hesaplama karmaşıklıklarının fazla olması dezavantajdır.

Eğer veri gerçekten istatistiksel metodun varsaydığı dağılıma göre oluşmuşsa, anomali tespiti istatiksel olarak doğrulanabilir. İstatiksel anomali tekniklerinin bir diğer iyi tarafı da dağılımın bulunma aşamasının anomalilerden etkilenmediği durumlarda, gözeticisiz modda (yani sınıfları olmayan verilerle) çalışabilmeleridir. Bu metodların kötü taraflarından ilki yüksek boyutlu çoğu veri için dağılıma dayanma varsayımın yanlış olmasıdır. Yüksek boyutlu verilerin karmaşık bir dağılımı varsa, en iyi hipotez testi istatistiğini seçmek zor-

dur [Motulsky (1995)]. Ayrıca histogram temelli parametrik olmayan anomali teknikleri çok boyutlu verilerde öznitelikler arasındaki ilişkiyi ihmal eder ve bazı öznitelik birleşimlerinin anomali olduğunu tespit edemez.

REFERANSLAR

Kitaplar

Alpaydin, E.: 2004, *Introduction to machine learning*, MIT press.

Anderson, D., Lunt, T. F., Javitz, H., Tamaru, A., Valdes, A. et al.: 1995, *Detecting unusual program behavior using the statistical component of the Next-generation Intrusion Detection Expert System (NIDES)*, SRI International, Computer Science Laboratory.

Bishop, C. M. et al.: 2006, *Pattern recognition and machine learning*, Vol. 1, springer New York.

Cherkassky, V. and Mulier, F. M.: 2007, *Learning from data: concepts, theory, and methods*, John Wiley & Sons.

Duda, R. O., Hart, P. E. and Stork, D. G.: 1999, *Pattern classification*, John Wiley & Sons,.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J. and Tibshirani, R.: 2009, *The elements of statistical learning*, Vol. 2, Springer.

Jackson, J. E.: 2005, *A user's guide to principal components*, Vol. 587, John Wiley & Sons.

Jolliffe, I.: 2005, *Principal component analysis*, Wiley Online Library.

Li, M. and Vitányi, P. M.: 2009, *An introduction to Kolmogorov complexity and its applications*, Springer.

Maindonald, J. and Braun, W. J.: 2010, *Data analysis and graphics using R: an example-based approach*, Vol. 10, Cambridge University Press.

Mitchell, T. M.: 1999, Machine learning and data mining, *Communications of the ACM* **42**(11), 30–36.

Monz, C.: 2007, *Model tree learning for query term weighting in question answering*, Springer.

Rokach, L.: 2008, *Data mining with decision trees: theory and applications*, Vol. 69, World scientific.

Rousseeuw, P. J. and Leroy, A. M.: 2005, *Robust regression and outlier detection*, Vol. 589, John Wiley & Sons.

Shewhart, W. A.: 1931, *Economic control of quality of manufactured product*, Vol. 509, ASQ Quality Press.

Simon, P.: 2013, *Too Big to Ignore: The Business Case for Big Data*, John Wiley & Sons.

Vapnik, V. N. and Vapnik, V.: 1998, *Statistical learning theory*, Vol. 2, Wiley New York.

Süreli Yayınlar

Abraham, B. and Box, G. E.: 1979, Bayesian analysis of some outlier problems in time series, *Biometrika* **66**(2), 229–236.

Agovic, A., Banerjee, A., Ganguly, A. R. and Protopopescu, V.: 2008, 6 anomaly detection in transportation corridors using manifold embedding, *Knowledge Discovery from Sensor Data* pp. 81–105.

Al-ani, T. and Trad, D.: 2010, Signal processing and classification approaches for brain-computer interface, *Intelligent and Biosensors, Edited by Vernon S. Somerset* pp. 25–66.

Anscombe, F. J.: 1960, Rejection of outliers, *Technometrics* **2**(2), 123–146.

Baker, L. D., Hofmann, T., McCallum, A. and Yang, Y.: 1999, A hierarchical probabilistic model for novelty detection in text, *NIPS'99, Unpublished manuscript* .

Bolton, R. J., Hand, D. J. et al.: 2001, Unsupervised profiling methods for fraud detection, *Credit Scoring and Credit Control VII* pp. 235–255.

Burbidge, R. and Buxton, B.: 2001, An introduction to support vector machines for data mining, *Keynote papers, young OR12* pp. 3–15.

Byers, S. and Raftery, A. E.: 1998, Nearest-neighbor clutter removal for estimating features in spatial point processes, *Journal of the American Statistical Association* **93**(442), 577–584.

Chandola, V., Banerjee, A. and Kumar, V.: 2009, Anomaly detection: A survey, *ACM Computing Surveys (CSUR)* **41**(3), 15.

Chen, W.-H. and Shih, J.-Y.: 2006, A study of taiwan's issuer credit rating systems using support vector machines, *Expert Systems with Applications* **30**(3), 427–435.

De Stefano, C., Sansone, C. and Vento, M.: 2000, To reject or not to reject: that is the question-an answer in case of neural classifiers, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **30**(1), 84–94.

Dutta, H., Giannella, C., Borne, K. D. and Kargupta, H.: 2007, Distributed top-k outlier detection from astronomy catalogs using the demac system., *SDM*, SIAM, pp. 473–478.

Edgeworth, F.: 1887, Xli. on discordant observations, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **23**(143), 364–375.

Eskin, E.: 2000, Anomaly detection over noisy data using learned probability distributions.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L. and Stolfo, S.: 2002, A geometric framework for unsupervised anomaly detection, *Applications of data mining in computer security*, Springer, pp. 77–101.

Fletcher, T.: 2009, Support vector machines explained, *Tutorial paper., Mar* .

Fox, A. J.: 1972, Outliers in time series, *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 350–363.

Grubbs, F. E.: 1969, Procedures for detecting outlying observations in samples, *Technometrics* **11**(1), 1–21.

Hawkins, D. M.: 1974, The detection of errors in multivariate data using principal components, *Journal of the American Statistical Association* **69**(346), 340–344.

He, Z., Xu, X. and Deng, S.: 2003, Discovering cluster-based local outliers, *Pattern Recognition Letters* **24**(9), 1641–1650.

Horn, P. S., Feng, L., Li, Y. and Pesce, A. J.: 2001, Effect of outliers and nonhealthy individuals on reference interval estimation, *Clinical Chemistry* **47**(12), 2137–2145.

Jolliffe, I. T.: 1972, Discarding variables in a principal component analysis. i: Artificial data, *Applied statistics* pp. 160–173.

Knorr, E. M. and Ng, R. T.: 1997, A unified approach for mining outliers, *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*, IBM Press, p. 11.

Kohonen, T. and Maps, S.-O.: 1995, Springer series in information sciences, *Self-organizing maps* **30**.

Laurikkala, J., Juhola, M., Kentala, E., Lavrac, N., Miksch, S. and Kavsek, B.: 2000, Informal identification of outliers in medical data, *Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, Citeseer, pp. 20–24.

Lee, W., Stolfo, S. J. and Mok, K. W.: 2000, Adaptive intrusion detection: A data mining approach, *Artificial Intelligence Review* **14**(6), 533–567.

Manson, G., Pierce, G. and Worden, K.: 2001, On the long-term stability of normal condition for damage detection in a composite panel, *Key Engineering Materials* **204**, 359–370.

Melgani, F. and Bruzzone, L.: 2004, Classification of hyperspectral remote sensing images with support vector machines, *Geoscience and Remote Sensing, IEEE Transactions on* **42**(8), 1778–1790.

Mitchell, T. M.: 1997, Machine learning. 1997, *Burr Ridge, IL: McGraw Hill* **45**.

Otey, M. E., Ghoting, A. and Parthasarathy, S.: 2006, Fast distributed outlier detection in mixed-attribute data sets, *Data Mining and Knowledge Discovery* **12**(2-3), 203–228.

Parra, L. C., Spence, C. D., Gerson, A. D. and Sajda, P.: 2003, Response error correction-a demonstration of improved human-machine performance using real-time eeg monitoring, *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* **11**(2), 173–177.

Rabaoui, A., Davy, M., Rossignol, S., Lachiri, Z. and Ellouze, N.: 2007, Using one-class svms and wavelets for audio surveillance systems, *submitted to IEEE trans. on Information Forensic and Security* .

Ratsch, G., Mika, S., Scholkopf, B. and Muller, K.: 2002, Constructing boosting algorithms from svms: an application to one-class classification, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **24**(9), 1184–1199.

Shen, J., Pei, Z., Fisher, G. and Lee, E.: 2006, Modelling and analysis of waviness reduction in soft-pad grinding of wire-sawn silicon wafers by support vector regression, *International journal of production research* **44**(13), 2605–2623.

Smith, L. I.: 2002, A tutorial on principal components analysis, *Cornell University, USA* **51**, 52.

Tsay, R. S., Peña, D. and Pankratz, A. E.: 2000, Outliers in multivariate time series, *Biometrika* **87**(4), 789–804.

Übeyli, E. D.: 2007, Combining eigenvector methods and support vector machines for detecting variability of doppler ultrasound signals, *Computer methods and programs in biomedicine* **86**(2), 181–190.

Ye, N. and Chen, Q.: 2001, An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems, *Quality and Reliability Engineering International* **17**(2), 105–112.

Zhang, H.: 2004, The optimality of naive bayes, *AA* **1**(2), 3.

Diğer Yayınlar

Aleskerov, E., Freisleben, B. and Rao, B.: 1997, Cardwatch: A neural network based database mining system for credit card fraud detection, *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, IEEE, pp. 220–226.

Bay, S. D. and Schwabacher, M.: 2003, Mining distance-based outliers in near linear time with randomization and a simple pruning rule, *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 29–38.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J.: 2000, Lof: identifying density-based local outliers, *ACM Sigmod Record*, Vol. 29, ACM, pp. 93–104.

Chandola, V., Boriah, S. and Kumar, V.: 2009, A framework for exploring categorical data., *SDM*, SIAM, pp. 187–198.

D'haeseleer, P., Forrest, S. and Helman, P.: 1996, An immunological approach to change detection: Algorithms, analysis and implications, *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, IEEE, pp. 110–119.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X.: 1996, A density-based algorithm for discovering clusters in large spatial databases with noise., *Kdd*, Vol. 96, pp. 226–231.

Fawcett, T. and Provost, F.: 1999, Activity monitoring: Noticing interesting changes in behavior, *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 53–62.

Fodor, I. K.: 2002, A survey of dimension reduction techniques.

Ghosh, A. K., Schwartzbard, A. and Schatz, M.: 1999, Learning program behavior profiles for intrusion detection., *Workshop on Intrusion Detection and Network Monitoring*, Vol. 51462.

Gönen, M. and Alpaydin, E.: 2008, Localized multiple kernel learning, *Proceedings of the 25th international conference on Machine learning*, ACM, pp. 352–359.

Gutierrez-Osuna, R.: 2005, Introduction to pattern analysis, *Lecture Notes, Texas A&M University* .

Idé, T. and Kashima, H.: 2004, Eigenspace-based anomaly detection in computer systems, *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 440–449.

Keogh, E., Lonardi, S. and Ratanamahatana, C. A.: 2004, Towards parameter-free data mining, *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 206–215.

Kleder, M.: 2005, Rapid lossless data compression of numerical or string variables, *MATLAB Central File Exchange: http://www. mathworks. com/matlabcentral/fileexchange/8899* .

Luke, B. T.: 2008, K-means clustering, *Tutorial Slides, http://fconyx ncifcrf. gov/˜lukeb/kmeans. html* .

MacQueen, J. et al.: 1967, Some methods for classification and analysis of multivariate observations, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1, California, USA, pp. 281–297.

Mahoney, M. V. and Chan, P. K.: 2002, Learning nonstationary models of normal network traffic for detecting novel attacks, *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 376–385.

Motulsky, H.: 1995, Choosing a statistical test.

Ng, A.: 2000, Cs229 lecture notes, *CS229 Lecture notes* **1**(1), 1–3.

Nyalkalkar, K., Sinha, S., Bailey, M. and Jahanian, F.: 2011, A comparative study of two network-based anomaly detection methods, *INFOCOM, 2011 Proceedings IEEE*, IEEE, pp. 176–180.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B. and Faloutsos, C.: 2003, Loci: Fast outlier detection using the local correlation integral, *Data Engineering, 2003. Proceedings. 19th International Conference on*, IEEE, pp. 315–326.

Ramaswamy, S., Rastogi, R. and Shim, K.: 2000, Efficient algorithms for mining outliers from large data sets, *ACM SIGMOD Record*, Vol. 29, ACM, pp. 427–438.

Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K. and Chang, L.: 2003, A novel anomaly detection scheme based on principal component classifier, *Technical report*, DTIC Document.

Spence, C., Parra, L. and Sajda, P.: 2001, Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model, *Mathematical Methods in Biomedical Image Analysis, 2001. MMBIA 2001. IEEE Workshop on*, IEEE, pp. 3–10.

Tan, P.-N., Steinbach, M. and Kumar, V.: 2005, Introduction to data mining, addison.

Tan, P. and Steinbach, M.: n.d., e kumar, v.(2005) introduction to data mining.

Tang, J., Chen, Z., Fu, A. W.-C. and Cheung, D. W.: 2002, Enhancing effectiveness of outlier detections for low density patterns, *Advances in Knowledge Discovery and Data Mining*, Springer, pp. 535–548.

Wei, L., Qian, W., Zhou, A., Jin, W. and Jeffrey, X. Y.: 2003, Hot: Hypergraph-based outlier test for categorical data, *Advances in Knowledge Discovery and Data Mining*, Springer, pp. 399–410.

Yairi, T., Kato, Y. and Hori, K.: 2001, Fault detection by mining association rules from house-keeping data, *Proc. of International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Vol. 3, Citeseer.