

Makine Öğrenimi El Kitabı Super VIP

Afshine AMIDI ve Shervine AMIDI

April 30, 2019

Contents

1 Gözetimli Öğrenme	2
1.1 Gözetimli Öğrenmeye Giriş	2
1.2 Gösterimler ve genel konsept	2
1.3 Lineer modeller	3
1.3.1 Lineer regresyon	3
1.3.2 Sınıflandırma ve lojistik regresyon	3
1.3.3 Genelleştirilmiş Lineer Modeller	3
1.4 Destek Vektör Makineleri	3
1.5 Üretici Öğrenme	4
1.5.1 Gauss Diskriminant (Ayrıtıcı) Analizi	4
1.5.2 Naive Bayes	4
1.6 Ağaç temelli ve topluluk yöntemleri	4
1.7 Diğer parametrik olmayan yaklaşımlar	5
1.8 Öğrenme Teorisi	5
2 Gözetimsiz Öğrenme	6
2.1 Gözetimsiz Öğrenmeye Giriş	6
2.2 Kümeleme	6
2.2.1 Beklenti-Ençoklama (Maksimizasyon)	6
2.2.2 k -ortalamlar (k -means) kümeleme	6
2.2.3 Hiyerarşik kümeleme	6
2.2.4 Kümeleme değerlendirme metrikleri	7
2.3 Boyut küçültme	7
2.3.1 Temel bileşenler analizi	7
2.3.2 Bağımsız bileşen analizi	7
3 Derin Öğrenme	8
3.1 Sinir Ağları	8
3.2 Evrimsel Sinir Ağları	9
3.3 Yinelenebilir Sinir Ağları	9
3.4 Pekiktirmeli Öğrenme ve Kontrol	9

4 İpuçları ve püf noktaları	10
4.1 Sınıflandırma metrikleri	10
4.2 Regresyon metrikleri	11
4.3 Model seçimi	11
4.4 Tam	12
5 Hatırlatma	13
5.1 Olasılık ve İstatistik	13
5.1.1 Olasılık ve Kombinasyonlara Giriş	13
5.1.2 Koşullu Olasılık	13
5.1.3 Rastgele Değişkenler	13
5.1.4 Ortak Dağılımlı Rastgele Değişkenler	14
5.1.5 Parameter estimation	14
5.2 Doğrusal Cebir ve Kalkülüs	15
5.2.1 Genel notasyonlar	15
5.2.2 Matris işlemleri	15
5.2.3 Matris özellikleri	16

1 Gözetimli Öğrenme

Başak Buluz ve Ayyüce Kızrak tarafından çevrilmiştir

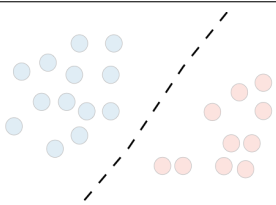
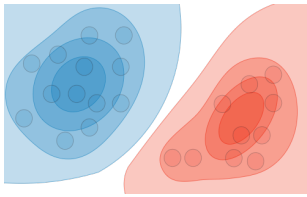
1.1 Gözetimli Öğrenmeye Giriş

$\{y^{(1)}, \dots, y^{(m)}\}$ çıktı kümesi ile ilişkili olan $\{x^{(1)}, \dots, x^{(m)}\}$ veri noktalarının kümesi göz önüne alındığında, y 'den x 'i nasıl tahmin edebileceğimizi öğrenen bir sınıflandırıcı tasarlamak istiyoruz.

□ **Tahmin türü** – Farklı tahmin modelleri aşağıdaki tabloda özetlenmiştir:

	Regresyon	Sınıflandırıcı
Çıktı	Sürekli	Sınıf
Örnekler	Lineer regresyon (bağlanım)	Lojistik regresyon (bağlanım), Destek Vektör Makineleri (DVM), Naive Bayes

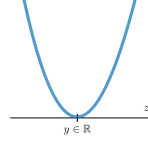
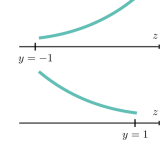
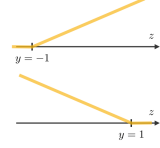
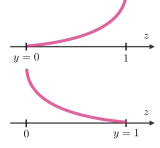
□ **Model türleri** – Farklı modeller aşağıdaki tabloda özetlenmiştir:

	Ayırt edici model	Üretici model
Amaç	Doğrudan tahmin $P(y x)$	$P(y x)$ 'i tahmin etmek için $P(x y)$ 'i tahmin etme
Öğrenilenler	Karar Sınırı	Verilerin olasılık dağılımı
Örnekleme		
Örnekler	Regresyon, DVM	GDA, Naive Bayes

1.2 Gösterimler ve genel konsept

□ **Hipotez** – Hipotez h_θ olarak belirtilmiştir ve bu bizim seçtiğimiz modeldir. Verilen $x^{(i)}$ verisi için modelin tahminlediği çıktı $h_\theta(x^{(i)})$ 'dir.

□ **Kayıp fonksiyonu** – $L : (z, y) \in \mathbb{R} \times Y \mapsto L(z, y) \in \mathbb{R}$ şeklinde tanımlanan bir kayıp fonksiyonu y gerçek değerine karşılık geleceği öngörülen z değerini girdi olarak alan ve ne kadar farklı olduklarını gösteren bir fonksiyondur. Yaygın kayıp fonksiyonları aşağıdaki tabloda özetlenmiştir:

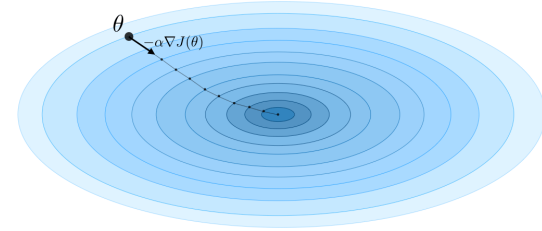
En küçük kareler hatası	Lojistik yitimi (kayıbı)	Menteşe yitimi (kayıbı)	Çapraz entropi
$\frac{1}{2}(y - z)^2$	$\log(1 + \exp(-yz))$	$\max(0, 1 - yz)$	$-[y \log(z) + (1 - y) \log(1 - z)]$
			
Lineer regresyon (bağlanım)	Lojistik regresyon (bağlanım)	DVM	Sinir Ağı

□ **Maliyet fonksiyonu** – J maliyet fonksiyonu genellikle bir modelin performansını değerlendirmek için kullanılır ve L kayıp fonksiyonu aşağıdaki gibi tanımlanır:

$$J(\theta) = \sum_{i=1}^m L(h_\theta(x^{(i)}), y^{(i)})$$

□ **Bayır inisi** – $\alpha \in \mathbb{R}$ öğrenme oranı olmak üzere, bayır inisi için güncelleme kuralı olarak ifade edilen öğrenme oranı ve J maliyet fonksiyonu aşağıdaki gibi ifade edilir:

$$\theta \leftarrow \theta - \alpha \nabla J(\theta)$$



Not: Stokastik bayır inisi her eğitim örneğine bağlı olarak parametreyi günceller, ve yığın bayır inisi bir dizi eğitim örneği üzerindedir.

□ **Olabilirlik** – θ parametreleri verilen bir $L(\theta)$ modelinin olabilirliğini, olabilirliği maksimize ederek en uygun θ parametrelerini bulmak için kullanılır. bulmak için kullanılır. Uygulamada, optimize edilmesi daha kolay olan log-olabilirlik $\ell(\theta) = \log(L(\theta))$ 'i kullanıyoruz. Sahip olduklarımız:

$$\theta^{\text{opt}} = \arg \max_{\theta} L(\theta)$$

□ **Newton'un algoritması** – $\ell'(\theta) = 0$ olacak şekilde bir θ bulan nümerik bir yöntemdir. Güncelleme kuralı aşağıdaki gibidir:

$$\theta \leftarrow \theta - \frac{\ell'(\theta)}{\ell''(\theta)}$$

Not: Newton-Raphson yöntemi olarak da bilinen çok boyutlu genelleme aşağıdaki güncelleme kuralına sahiptir:

$$\theta \leftarrow \theta - \left(\nabla_{\theta}^2 \ell(\theta) \right)^{-1} \nabla_{\theta} \ell(\theta)$$

1.3 Lineer modeller

1.3.1 Lineer regresyon

$y|x; \theta \sim \mathcal{N}(\mu, \sigma^2)$ olduğunu varsayıyoruz

□ **Normal denklemler** – X matris tasarımı olmak üzere, maliyet fonksiyonunu en aza indiren θ değeri X 'in matris tasarımı not ederek, maliyet fonksiyonunu en aza indiren θ değeri kapalı formülü bir çözümdür:

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

□ **En Küçük Ortalama Kareler algoritması** – α öğrenme oranı olmak üzere, m veri noktasını içeren eğitim kümesi için Widrow-Hoff öğrenme oranı olarak bilinen En Küçük Ortalama Kareler Algoritmasının güncelleme kuralı aşağıdaki gibidir:

$$\forall j, \quad \theta_j \leftarrow \theta_j + \alpha \sum_{i=1}^m \left[y^{(i)} - h_{\theta}(x^{(i)}) \right] x_j^{(i)}$$

Not: güncelleme kuralı, bayır yükselişinin özel bir halidir.

□ **Yerel Ağırlıklı Regresyon** – LWR olarak da bilinen Yerel Ağırlıklı Regresyon ağırlıkları her eğitim örneğini maliyet fonksiyonunda $w^{(i)}(x)$ ile ölçen doğrusal regresyonun bir çeşididir.

$$w^{(i)}(x) = \exp \left(-\frac{(x^{(i)} - x)^2}{2\tau^2} \right)$$

1.3.2 Sınıflandırma ve lojistik regresyon

□ **Sigmoid fonksiyonu** – Lojistik fonksiyonu olarak da bilinen sigmoid fonksiyonu g , aşağıdaki gibi tanımlanır:

$$\forall z \in \mathbb{R}, \quad g(z) = \frac{1}{1 + e^{-z}} \in]0,1[$$

□ **Lojistik regresyon** – $y|x; \theta \sim \text{Bernoulli}(\phi)$ olduğunu varsayıyoruz. Aşağıdaki forma sahibiz:

$$\phi = p(y = 1|x; \theta) = \frac{1}{1 + \exp(-\theta^T x)} = g(\theta^T x)$$

Not: Lojistik regresyon durumunda kapalı form çözümü yoktur.

□ **Softmax regresyonu** – Çok sınıflı lojistik regresyon olarak da adlandırılan Softmax regresyonu 2'den fazla sınıf olduğunda lojistik regresyonu genelleştirmek için kullanılır. Genel kabul olarak, her i sınıfı için Bernoulli parametresi ϕ_i 'nin eşit olmasını sağlaması için $\theta_K = 0$ olarak ayarlanır.

$$\phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^K \exp(\theta_j^T x)}$$

1.3.3 Genelleştirilmiş Lineer Modeller

□ **Üstel aile** – Eğer kanonik parametre veya bağlantı fonksiyonu olarak adlandırılan doğal bir parametre η , yeterli bir istatistik $T(y)$ ve aşağıdaki gibi bir log-partition fonksiyonu $a(\eta)$ şeklinde yazılabilirse, dağılım sınıfının üstel ailede olduğu söylenir:

$$p(y; \eta) = b(y) \exp(\eta T(y) - a(\eta))$$

Not: Sık sık $T(y) = y$ olur. Ayrıca, $\exp(-a(\eta))$, olasılıkların birleştiğinden emin olan normalleştirme parametresi olarak görülebilir.

Aşağıdaki tabloda özetlenen en yaygın üstel dağılımlar:

Dağılım	η	$T(y)$	$a(\eta)$	$b(y)$
Bernoulli	$\log \left(\frac{\phi}{1-\phi} \right)$	y	$\log(1 + \exp(\eta))$	1
Gauss	μ	y	$\frac{\eta^2}{2}$	$\frac{1}{\sqrt{2\pi}} \exp \left(-\frac{y^2}{2} \right)$
Poisson	$\log(\lambda)$	y	e^{η}	$\frac{1}{y!}$
Geometrik	$\log(1 - \phi)$	y	$\log \left(\frac{e^{\eta}}{1 - e^{\eta}} \right)$	1

□ **Genelleştirilmiş Lineer Modellerin Yaklaşımları** – Genelleştirilmiş Lineer Modeller $x \in \mathbb{R}^{n+1}$ için rastgele bir y değişkenini tahminlemeyi hedeflen ve aşağıdaki 3 varsayıma dayanan bir fonksiyondur:

$$(1) \quad y|x; \theta \sim \text{ExpFamily}(\eta) \quad (2) \quad h_{\theta}(x) = E[y|x; \theta] \quad (3) \quad \eta = \theta^T x$$

Not: sıradan en küçük kareler ve lojistik regresyon, genelleştirilmiş doğrusal modellerin özel durumlarıdır.

1.4 Destek Vektör Makineleri

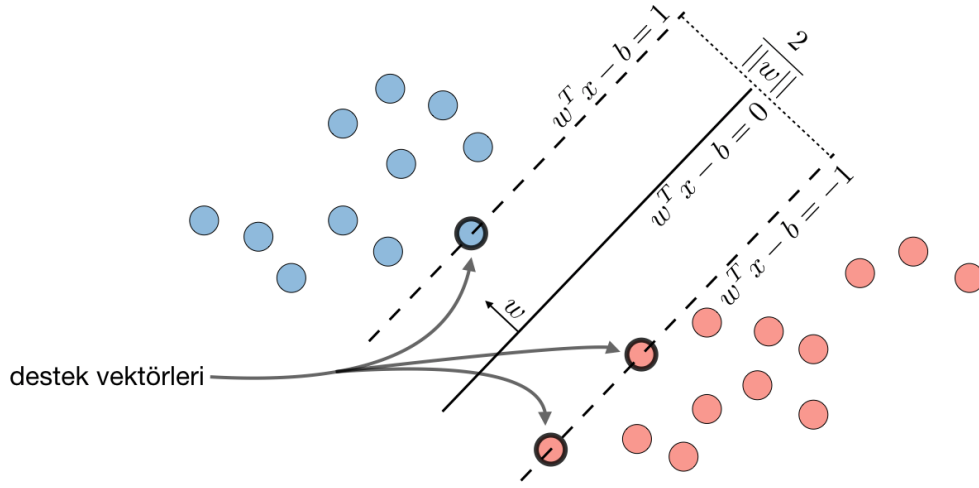
Destek Vektör Makinelerinin amacı minimum mesafeyi maksimuma çıkaran doğruyu bulmaktır.

□ **Optimal marj sınıflandırıcısı** – h optimal marj sınıflandırıcısı şöyledir:

$$h(x) = \text{sign}(w^T x - b)$$

burada $(w, b) \in \mathbb{R}^n \times \mathbb{R}$, aşağıdaki optimizasyon probleminin çözümüdür:

$$\min \frac{1}{2} \|w\|^2 \quad \text{öyle ki} \quad y^{(i)} (w^T x^{(i)} - b) \geq 1$$



Not: doğru $w^T x - b = 0$ şeklinde tanımlanır.

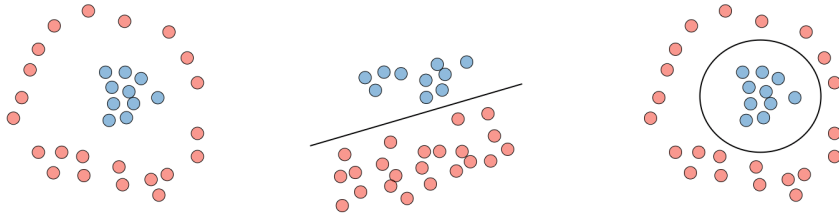
□ **Menteşe yitimi (kayıbı)** – Mentese yitimi Destek Vektör Makinelerinin ayarlarında kullanılır ve aşağıdaki gibi tanımlanır:

$$L(z, y) = [1 - yz]_+ = \max(0, 1 - yz)$$

□ **Çekirdek** – ϕ gibi bir özellik haritası verildiğinde, K olarak tanımlanacak çekirdek tanımlarız:

$$K(x, z) = \phi(x)^T \phi(z)$$

Uygulamada, $K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$ tarafından tanımlanan çekirdek K , Gauss çekirdeği olarak adlandırılır ve yaygın olarak kullanılır.



Lineer olmayan ayrılabilirlik \Rightarrow Çekirdek Haritalarının Kullanımı $\phi \Rightarrow$ Orjinal uzayda karar sınırı

Not: Çekirdeği kullanarak maliyet fonksiyonunu hesaplamak için "çekirdek numarası" nı kullandığımızı söylüyoruz çünkü genellikle çok karmaşık olan ϕ açık haritalamasını bilmeye gerek yok. Bunun yerine, yalnızca $K(x, z)$ değerlerine ihtiyacımız vardır.

□ **Lagranj** – Lagranj $\mathcal{L}(w, b)$ şeklinde şöyle tanımlanır:

$$\mathcal{L}(w, b) = f(w) + \sum_{i=1}^l \beta_i h_i(w)$$

Not: β_i katsayılarına Lagranj çarpanları denir.

1.5 Üretici Öğrenme

Üretken bir model, önce Bayes kuralını kullanarak $P(y|x)$ değerini tahmin etmek için kullanabileceğimiz $P(x|y)$ değerini tahmin ederek verilerin nasıl üretildiğini öğrenmeye çalışır.

1.5.1 Gauss Diskriminant (Ayırtaç) Analizi

□ **Yöntem** – Gauss Diskriminant Analizi y ve $x|y = 0$ ve $x|y = 1$ 'in şu şekilde olduğunu varsayar:

$$y \sim \text{Bernoulli}(\phi)$$

$$x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma) \quad \text{ve} \quad x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$$

□ **Tahmin** – Aşağıdaki tablo, olasılığı en üst düzeye çıkarırken bulduğumuz tahminleri özetlemektedir:

$\hat{\phi}$	$\hat{\mu}_j \quad (j = 0, 1)$	$\hat{\Sigma}$
$\frac{1}{m} \sum_{i=1}^m 1_{\{y^{(i)}=1\}}$	$\frac{\sum_{i=1}^m 1_{\{y^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{y^{(i)}=j\}}}$	$\frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$

1.5.2 Naive Bayes

□ **Varsayım** – Naive Bayes modeli, her veri noktasının özelliklerinin tamamen bağımsız olduğunu varsayar:

$$P(x|y) = P(x_1, x_2, \dots | y) = P(x_1|y)P(x_2|y) \dots = \prod_{i=1}^n P(x_i|y)$$

□ **Çözümler** – Log-olabilirliğinin $k \in \{0, 1\}, l \in \llbracket 1, L \rrbracket$ ile birlikte aşağıdaki çözümlerle maksimize edilmesi:

$$P(y = k) = \frac{1}{m} \times \#\{j|y^{(j)} = k\} \quad \text{ve} \quad P(x_i = l|y = k) = \frac{\#\{j|y^{(j)} = k \text{ ve } x_i^{(j)} = l\}}{\#\{j|y^{(j)} = k\}}$$

Not: Naive Bayes, metin sınıflandırması ve spam tespitinde yaygın olarak kullanılır.

1.6 Ağaç temelli ve topluluk yöntemleri

Bu yöntemler hem regresyon hem de sınıflandırma problemleri için kullanılabilir.

□ **CART** – Sınıflandırma ve Regresyon Ağaçları (Classification and Regression Trees (CART)), genellikle karar ağaçları olarak bilinir, ikili ağaçlar olarak temsil edilirler.

□ **Rastgele orman** – Rastgele seçilen özelliklerden oluşan çok sayıda karar ağacı kullanan ağaç tabanlı bir tekniktir. Basit karar ağacının tersine, oldukça yorumlanamaz bir yapıdadır ancak genel olarak iyi performansı onu popüler bir algoritma yapar.

Not: Rastgele ormanlar topluluk yöntemlerindendir.

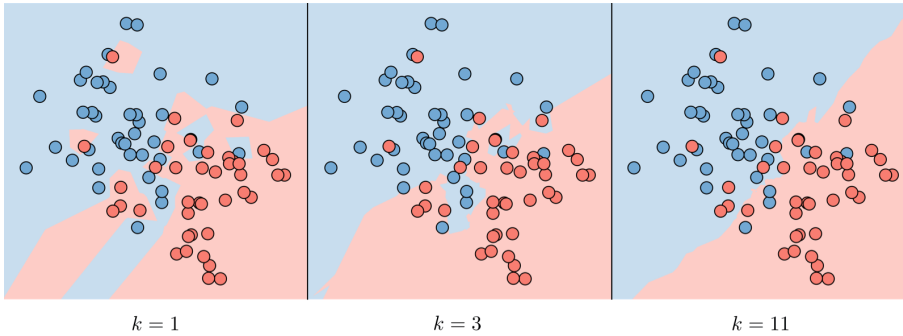
□ **Artırım** – Artırım yöntemlerinin temel fikri bazı zayıf öğrenicileri bir araya getirerek güçlü bir öğrenici oluşturmaktır. Temel yöntemler aşağıdaki tabloda özetlenmiştir:

Adaptif artırma	Gradyan artırma
Yüksek ağırlıklar bir sonraki artırma adımında iyileşmesi için hatalara maruz kalır.	Zayıf öğreniciler kalan hatalar üzerinde eğitildi

1.7 Diğer parametrik olmayan yaklaşımlar

□ **k -en yakın komşular** – Genellikle k -NN olarak adlandırılan k -en yakın komşular algoritması, bir veri noktasının tepkisi eğitim kümesindeki kendi k komşularının doğası ile belirlenen parametrik olmayan bir yaklaşımdır. Hem sınıflandırma hem de regresyon yöntemleri için kullanılabilir.

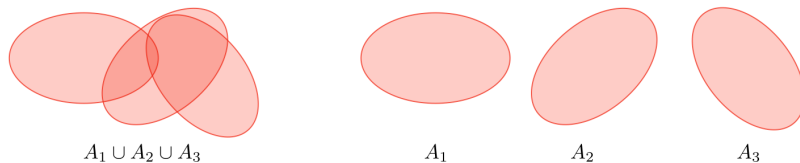
Not: k parametresi ne kadar yüksekse, yanlılık okadar yüksek ve k parametresi ne kadar düşüğe, varyans o kadar yüksek olur.



1.8 Öğrenme Teorisi

□ **Birleşim sınırı** – A_1, \dots, A_k k olayları olsun. Sahip olduklarımız:

$$P(A_1 \cup \dots \cup A_k) \leq P(A_1) + \dots + P(A_k)$$



□ **Hoeffding eşitsizliği** – Z_1, \dots, Z_m, ϕ parametresinin Bernoulli dağılımından çizilen değişkenler olsun. Örnek ortalamaları mean ve $\gamma > 0$ sabit olsun. Sahip olduklarımız:

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2 \exp(-2\gamma^2 m)$$

Not: Bu eşitsizlik, Chernoff sınırı olarak da bilinir.

□ **Eğitim hatası** – Belirli bir h sınıflandırıcısı için, ampirik risk veya ampirik hata olarak da bilinen eğitim hatasını $\hat{\epsilon}(h)$ şöyle tanımlarız:

$$\hat{\epsilon}(h) = \frac{1}{m} \sum_{i=1}^m 1_{\{h(x^{(i)}) \neq y^{(i)}\}}$$

□ **Olası Yaklaşık Doğru** – PAC, öğrenme teorisi üzerine sayısız sonuçların kanıtlandığı ve aşağıdaki varsayımlara sahip olan bir çerçevedir:

- eğitim ve test kümeleri aynı dağılımı takip ediyor
- eğitim örnekleri bağımsız olarak çizilir

□ **Parçalanma** – $S = \{x^{(1)}, \dots, x^{(d)}\}$ kümesi ve \mathcal{H} sınıflandırıcıların kümesi verildiğinde, \mathcal{H} herhangi bir etiketler kümesi S' 'e parçalar.

$$\exists h \in \mathcal{H}, \quad \forall i \in [1, d], \quad h(x^{(i)}) = y^{(i)}$$

□ **Üst sınır teoremi** – $|\mathcal{H}| = k$, δ ve örneklem sayısı m 'nin sabit olduğu sonlu bir hipotez sınıfı \mathcal{H} olsun. Ardından, en az $1 - \delta$ olasılığı ile elimizde:

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + 2 \sqrt{\frac{1}{2m} \log \left(\frac{2k}{\delta} \right)}$$

□ **VC boyutu** – $VC(\mathcal{H})$ olarak ifade edilen belirli bir sonsuz \mathcal{H} hipotez sınıfının Vapnik-Chervonenkis (VC) boyutu, \mathcal{H} tarafından parçalanılan en büyük kümenin boyutudur.

Not: $\mathcal{H} = \{2 \text{ boyutta doğrusal sınıflandırıcılar kümesi}\}$ 'nin VC boyutu 3'tür.



□ **Teorem (Vapnik)** – \mathcal{H} , $VC(\mathcal{H}) = d$ ve eğitim örneği sayısı m verilmiş olsun. En az $1 - \delta$ olasılığı ile, sahip olduklarımız:

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + O \left(\sqrt{\frac{d}{m} \log \left(\frac{m}{d} \right)} + \frac{1}{m} \log \left(\frac{1}{\delta} \right) \right)$$

2 Gözetimsiz Öğrenme

Yavuz Kömeçoğlu ve Başak Buluz tarafından çevrilmiştir

2.1 Gözetimsiz Öğrenmeye Giriş

□ **Motivasyon** – Gözetimsiz öğrenmenin amacı etiketlenmemiş verilerdeki gizli örüntüleri bulmaktır $\{x^{(1)}, \dots, x^{(m)}\}$.

□ **Jensen eşitsizliği** – f bir konveks fonksiyon ve X bir rastgele değişken olsun. Aşağıdaki eşitsizliklerimiz:

$$E[f(X)] \geq f(E[X])$$

2.2 Kümeleme

2.2.1 Beklenti-Ençoklama (Maksimizasyon)

□ **Gizli değişkenler** – Gizli değişkenler, tahmin problemlerini zorlaştıran ve çoğunlukla z olarak adlandırılan gizli/gözlemlenmemiş değişkenlerdir. Gizli değişkenlerin bulunduğu yerlerdeki en yaygın ayarlar şöyledir:

Yöntem	Gizli değişken z	$x z$	Açıklamalar
k Gaussianların birleşimi	Multinomial(ϕ)	$\mathcal{N}(\mu_j, \Sigma_j)$	$\mu_j \in \mathbb{R}^n, \phi \in \mathbb{R}^k$
Faktör analizi	$\mathcal{N}(0, I)$	$\mathcal{N}(\mu + \Lambda z, \psi)$	$\mu_j \in \mathbb{R}^n$

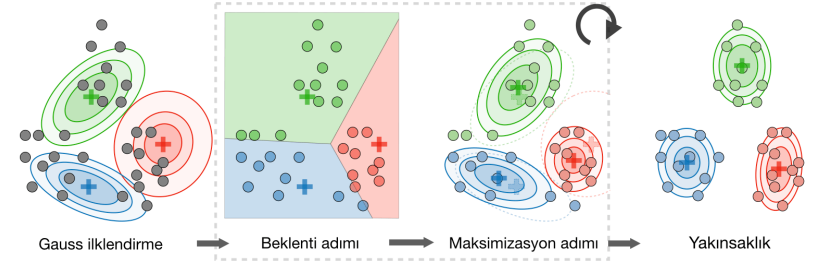
□ **Algoritma** – Beklenti-Ençoklama (Maksimizasyon) (BE) algoritması, θ parametresinin maksimum olabilirlik kestirimiyle tahmin edilmesinde, olasılığa ard arda alt sınırlar oluşturan (E-adımı) ve bu alt sınırın (M-adımı) aşağıdaki gibi optimize edildiği etkin bir yöntem sunar:

- **E-adımı:** Her bir veri noktasının $x^{(i)}$ 'in belirli bir kümeden $z^{(i)}$ geldiğinin sonsal olasılık değerinin $Q_i(z^{(i)})$ hesaplanması aşağıdaki gibidir:

$$Q_i(z^{(i)}) = P(z^{(i)} | x^{(i)}; \theta)$$

- **M-adımı:** Her bir küme modelini ayrı ayrı yeniden tahmin etmek için $x^{(i)}$ veri noktalarındaki kümeye özgü ağırlıklar olarak $Q_i(z^{(i)})$ sonsal olasılıklarının kullanımı aşağıdaki gibidir:

$$\theta_i = \arg\max_{\theta} \sum_i \int_{z^{(i)}} Q_i(z^{(i)}) \log \left(\frac{P(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right) dz^{(i)}$$

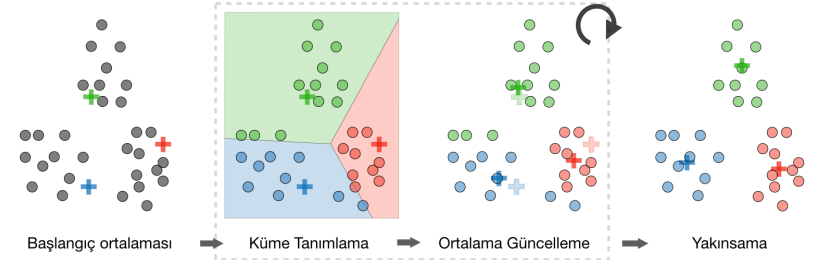


2.2.2 k -ortalamlar (k -means) kümeleme

$c^{(i)}$, i veri noktasının bulunduğu küme olmak üzere, μ_j j kümesinin merkez noktasıdır.

□ **Algoritma** – Küme ortalamaları $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$ rasgele olarak başlatıldıktan sonra, k -ortalamlar algoritması yakınsayana kadar aşağıdaki adımı tekrar eder:

$$c^{(i)} = \arg \min_j \|x^{(i)} - \mu_j\|^2 \quad \text{and} \quad \mu_j = \frac{\sum_{i=1}^m 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{c^{(i)}=j\}}}$$



□ **Bozulma fonksiyonu** – Algoritmanın yakınsadığını görmek için aşağıdaki gibi tanımlanan bozulma fonksiyonuna bakarız:

$$J(c, \mu) = \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

2.2.3 Hiyerarşik kümeleme

□ **Algoritma** – Ardışık olarak iç içe geçmiş kümelerden oluşturan hiyerarşik bir yaklaşıma sahip bir kümeleme algoritmasıdır.

□ **Türler** – Aşağıdaki tabloda özetlenen farklı amaç fonksiyonlarını optimize etmeyi amaçlayan farklı hiyerarşik kümeleme algoritmaları vardır:

Ward bağlantı	Ortalama bağlantı	Tam bağlantı
Küme mesafesi içinde minimize edin	Küme çiftleri arasındaki ortalama uzaklığı en aza indirin	Küme çiftleri arasındaki maksimum uzaklığı en aza indirin

2.2.4 Kümeleme değerlendirme metrikleri

Gözetimsiz bir öğrenme ortamında, bir modelin performansını değerlendirmek çoğu zaman zordur, çünkü gözetimli öğrenme ortamında olduğu gibi, gerçek referans etiketlere sahip değiliz.

□ **Siluet katsayısı** – Bir örnek ile aynı sınıftaki diğer tüm noktalar arasındaki ortalama mesafeyi ve bir örnek ile bir sonraki en yakın kümedeki diğer tüm noktalar arasındaki ortalama mesafeyi not ederek, tek bir örnek için siluet katsayısı aşağıdaki gibi tanımlanır:

$$s = \frac{b - a}{\max(a, b)}$$

□ **Calinski-Harabaz indeksi** – k kümelerin sayısını belirtmek üzere B_k ve W_k sırasıyla, kümeler arası ve küme içi dağılım matrisleri olarak aşağıdaki gibi tanımlanır

$$B_k = \sum_{j=1}^k n_{c(i)} (\mu_{c(i)} - \mu)(\mu_{c(i)} - \mu)^T, \quad W_k = \sum_{i=1}^m (x^{(i)} - \mu_{c(i)})(x^{(i)} - \mu_{c(i)})^T$$

Calinski-Harabaz indeksi $s(k)$, kümelenme modelinin kümeleri ne kadar iyi tanımladığını gösterir, böylece skor ne kadar yüksek olursa, kümeler daha yoğun ve iyi ayrılır. Aşağıdaki şekilde tanımlanmıştır:

$$s(k) = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{N - k}{k - 1}$$

2.3 Boyut küçültme

2.3.1 Temel bileşenler analizi

Verilerin yansıtılacağı yönleri maksimize eden varyansı bulan bir boyut küçültme tekniğidir.

□ **Özdeğer, özvektör** – Bir matris $A \in \mathbb{R}^{n \times n}$ verildiğinde λ 'nın, özvektör olarak adlandırılan bir vektör $z \in \mathbb{R}^n \setminus \{0\}$ varsa, A 'nın bir özdeğeri olduğu söylenir:

$$Az = \lambda z$$

□ **Spektral teorem** – $A \in \mathbb{R}^{n \times n}$ olsun. Eğer A simetrik ise, o zaman A gerçek ortogonal matris $U \in \mathbb{R}^{n \times n}$ ile diyagonalleştirilebilir. $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ yazarak, bizde:

$$\exists \Lambda \text{ diyagonal, } A = U \Lambda U^T$$

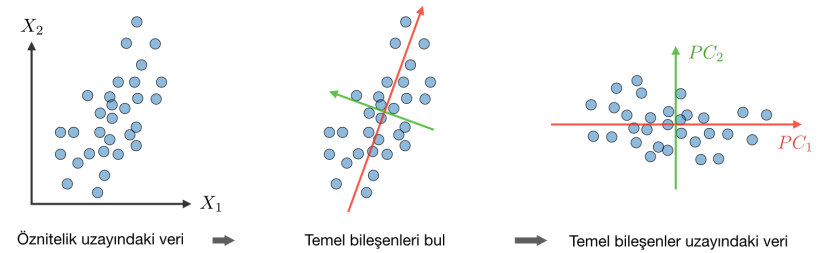
Not: En büyük özdeğere sahip özvektör, matris A 'nın temel özvektörü olarak adlandırılır.

□ **Algoritma** – Temel Bileşen Analizi (TBA) yöntemi, verilerin aşağıdaki gibi varyansı en üst düzeye çıkararak veriyi k boyutlarına yansıtan bir boyut azaltma tekniğidir:

- **Adım 1:** Verileri ortalama 0 ve standart sapma 1 olacak şekilde normalleştirin.

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j} \quad \text{where} \quad \mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)} \quad \text{and} \quad \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$$

- **Adım 2:** Gerçek özdeğerler ile simetrik olan $\Sigma = \frac{1}{m} \sum_{i=1}^m x^{(i)} x^{(i)T} \in \mathbb{R}^{n \times n}$ hesaplayın.
- **Adım 3:** $u_1, \dots, u_k \in \mathbb{R}^n$ olmak üzere Σ ort'nin ortogonal ana özvektörlerini, yani k en büyük özdeğerlerin ortogonal özvektörlerini hesaplayın.
- **Adım 4:** $\text{span}_{\mathbb{R}}(u_1, \dots, u_k)$ üzerindeki verileri gösterin. Bu yöntem tüm k -boyutlu uzaylar arasındaki varyansı en üst düzeye çıkarır.



2.3.2 Bağımsız bileşen analizi

Temel oluşturan kaynakları bulmak için kullanılan bir tekniktir.

□ **Varsayımlar** – Verilerin x 'in n boyutlu kaynak vektörü $s = (s_1, \dots, s_n)$ tarafından üretildiğini varsayıyoruz, burada s_i bağımsız rasgele değişkenler, bir karışım ve tekil olmayan bir matris A ile aşağıdaki gibi:

$$x = As$$

Amaç, işlem görmemiş matrisini $W = A^{-1}$ bulmaktır.

□ **Bell ve Sejnowski ICA algoritması** – Bu algoritma, aşağıdaki adımları izleyerek işlem görmemiş matrisi W 'yi bulur:

- $x = As = W^{-1}s$ olasılığını aşağıdaki gibi yazınız:

$$p(x) = \prod_{i=1}^n p_s(w_i^T x) \cdot |W|$$

- Eğitim verisi $\{x^{(i)}, i \in [1, m]\}$ ve g sigmoid fonksiyonunu not ederek log olasılığını yazınız:

$$l(W) = \sum_{i=1}^m \left(\sum_{j=1}^n \log \left(g'(w_j^T x^{(i)}) \right) + \log |W| \right)$$

Bu nedenle, rassal (stokastik) eğitim yükselme öğrenme kuralı, her bir eğitim örneği için $x^{(i)}$, W 'yi aşağıdaki gibi güncelleştiririz:

$$W \leftarrow W + \alpha \left(\begin{pmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{pmatrix} x^{(i)T} + (W^T)^{-1} \right)$$

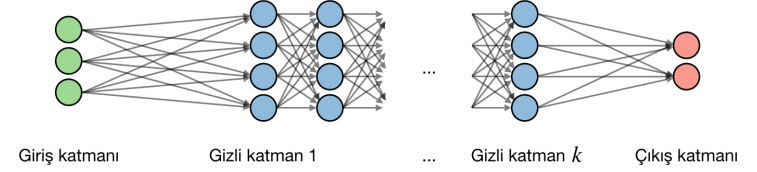
3 Derin Öğrenme

Ekrem Çetinkaya ve Omer Bukte tarafından çevrilmiştir

3.1 Sinir Ağları

Sinir ağları, katmanlarla inşa edilen bir modeller sınıfıdır. Sinir ağlarının yaygın kullanılan çeşitleri evrimsel sinir ağları ve yinelenen sinir ağlarını içerir.

□ **Mimari** – Sinir ağları mimarisi aşağıdaki figürde açıklanmaktadır:



Ağın i . sırasındaki katmana i ve katmandaki j . sırasındaki gizli birime j dersek, elimizde:

$$z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]}$$

burada w , b , z değerleri sırasıyla ağırlık, eğilim ve ürünü temsil eder.

□ **Etkinleştirme fonksiyonu** – Etkinleştirme fonksiyonları gizli birimlerin sonunda, modele lineer olmayan karmaşıklıklar katmak için kullanılır. Aşağıda en yaygın kullanılanlarını görebilirsiniz:

Sigmoid	Tanh	ReLU	Leaky ReLU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ ile $\epsilon \ll 1$

□ **Çapraz-entropi kaybı** – Sinir ağları içeriğinde, çapraz-entropi kaybı $L(z, y)$ sık olarak kullanılır ve aşağıdaki gibi tanımlanır:

$$L(z, y) = - \left[y \log(z) + (1 - y) \log(1 - z) \right]$$

□ **Öğrenme oranı** – Öğrenme oranı, sıklıkla α veya bazen η olarak belirtilir, ağırlıkların hangi tempoda güncellendiğini gösterir. Bu derece sabit olabilir veya uyarlamalı olarak değişebilir. Mevcut en gözde yöntem Adam olarak adlandırılan ve öğrenme oranını uyarlayan bir yöntemdir.

□ **Geri yayılım** – Geri yayılım sinir ağındaki ağırlıkları güncellemek için kullanılan ve bunu yaparken de asıl sonuç ile istenilen sonucu hesaba katan bir yöntemdir. Ağırlık w değerine göre türev, zincir kuralı kullanılarak hesaplanılır ve aşağıdaki şekildedir:

$$\frac{\partial L(z,y)}{\partial w} = \frac{\partial L(z,y)}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w}$$

Sonuç olarak, ağırlık güncellenmesi aşağıdaki gibidir:

$$w \leftarrow w - \eta \frac{\partial L(z,y)}{\partial w}$$

□ **Ağırlıkları güncelleme** – Sinir ağında ağırlıklar, aşağıdaki gibi güncellenir:

- 1. Adım: Bir eğitim verisi kümesi alınır.
- 2. Adım: Denk gelen kaybı elde etmek için, ileri yayılım gerçekleştirilir.
- 3. Adım: Gradyanları elde etmek için kayba geri yayılım uygulanır.
- 4. Adım: Ağın ağırlıklarını güncellemek için gradyanlar kullanılır.

□ **Düşürme** – Düşürme, eğitim verisinin aşırı uymasını engellemek için sinir ağındaki birimleri düşürmek yoluyla uygulanan bir tekniktir. Pratikte, nöronlar ya p olasılığla düşürülür ya da $1 - p$ olasılığla tutulur.

3.2 Evrişimsel Sinir Ağları

□ **Evrişimsel katman gereksinimleri** – Girdi boyutuna W , evrişimsel katman nöronlarının boyutlarına F , sıfır dolgulama miktarına P dersek, belirlenmiş bir boyuta sığacak neuron sayısı N şu şekildedir:

$$N = \frac{W - F + 2P}{S} + 1$$

□ **Küme normalleştirilmesi** – γ, β Hiper-parametresinin, $\{x_i\}$ kümesini normalleştiren bir adımdır. μ_B, σ_B^2 ifadelerine düzeltmek istediğimiz kümenin ortalaması ve varyansı dersek, normalleştirme işlemi şu şekilde yapılır:

$$x_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

Bu işlem genelde tamamıyla bağlantılı/evrişimsel olan bir katmandan sonra ve lineer olmayan bir katmandan önce yapılır. Bu işlem ile daha yüksek öğrenme derecesi elde etmeye imkan sağlamak ve de öndeger atamaya olan güçlü bağımlılığı azaltmak amaçlanır.

3.3 Yinelenen Sinir Ağları

□ **Kapı çeşitleri** – Aşağıda tipik bir yinelenen sinir ağlarında karşımıza çıkan farklı kapı örnekleri görülebilir:

Girdi kapısı	Unutma kapısı	Kapı	Çıktı kapısı
Hücreye yaz/yazma ?	Hücreyi sil/silme ?	Hücreye ne kadar yazmalı ?	Hücresinin ne kadarını açığa çıkarmalı ?

□ **LSTM** – Uzun, kısa vadeli hafıza (LSTM) ağı, 'unutma' kapılarını ekleyerek yok olan gradyan problemlinden kurtulabilen bir çeşit RNN modelidir.

3.4 Pekiştirmeli Öğrenme ve Kontrol

Pekiştirmeli öğrenmenin hedefi, bir hedefin bir ortamda nasıl değişiklik geçireceğini öğrenmesini sağlamaktır.

□ **Markov karar süreci** – Markov karar süreci (MDP) 5 öğelidir $(\mathcal{S}, \mathcal{A}, \{P_{sa}\}, \gamma, R)$ ve bu ifadeler şunları temsil eder:

- \mathcal{S} , hallerin setidir
- \mathcal{A} , aksiyonların setidir
- $\{P_{sa}\}$ $s \in \mathcal{S}$ ve $a \in \mathcal{A}$ için hal değişimlerinin olasılıklarıdır
- $\gamma \in [0, 1[$ azaltma unsurudur
- $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ veya $R : \mathcal{S} \rightarrow \mathbb{R}$ algoritmanın en yüksek düzeye çıkartmak istediği ödül fonksiyonudur

□ **Prensip** – π prensibi hal-aksiyon eşleşmesini yapan $\pi : \mathcal{S} \rightarrow \mathcal{A}$ fonksiyonudur.

Dipnot: Eğer s hali verildiğinde $a = \pi(s)$ aksiyonunu uyguluyorsak, π prensibini yerine getirdik deriz.

□ **Değer fonksiyonu** – π prensibi ve s hali verildiğinde, V^π değer fonksiyonu aşağıdaki gibi tanımlanır:

$$V^\pi(s) = E \left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots | s_0 = s, \pi \right]$$

□ **Bellman denklemi** – İdeal Bellman denklemleri, ideal prensip π^* değerinin değer fonksiyonu V^{π^*} değerini simgeler:

$$V^{\pi^*}(s) = R(s) + \max_{a \in \mathcal{A}} \gamma \sum_{s' \in \mathcal{S}} P_{sa}(s') V^{\pi^*}(s')$$

Dipnot: s hali verildiğinde, ideal π^ prensibini şu şekilde tanımlarız:*

$$\pi^*(s) = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{sa}(s') V^*(s')$$

□ **Değer iterasyon algoritması** – Değer iterasyon algoritması iki adımdan oluşur:

- Değere ilk değer atarız:

$$V_0(s) = 0$$

- Daha önceki değerlere göre değere iterasyon uyguluyoruz:

$$V_{i+1}(s) = R(s) + \max_{a \in \mathcal{A}} \left[\sum_{s' \in \mathcal{S}} \gamma P_{sa}(s') V_i(s') \right]$$

□ **Maksimum ihtimal tahmini** – Maksimum ihtimal hal geçişi olasılıklarını aşağıdaki şekilde tahmin eder:

$$P_{sa}(s') = \frac{\text{\#times took action } a \text{ in state } s \text{ and got to } s'}{\text{\#times took action } a \text{ in state } s}$$

□ **Q-Öğrenimi** – Q-Öğrenimi modelden bağımsız bir Q tahmini yapılan bir yöntemdir ve aşağıdaki gibi yapılır:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

4 İpuçları ve püf noktaları

Seray Beşer, Ayyüce Kızrak ve Yavuz Kömeçoğlu tarafından çevrilmiştir

4.1 Sınıflandırma metrikleri

İkili bir sınıflandırma durumunda, modelin performansını değerlendirmek için gerekli olan ana metrikler aşağıda verilmiştir.

□ **Karışıklık matrisi** – Karışıklık matrisi, bir modelin performansını değerlendirirken daha eksiksiz bir sonuca sahip olmak için kullanılır. Aşağıdaki şekilde tanımlanmıştır:

		Tahmini sınıf	
		+	-
Gerçek sınıf	+	TP True Positives	FN False Negatives Type II error
	-	FP False Positives Type I error	TN True Negatives

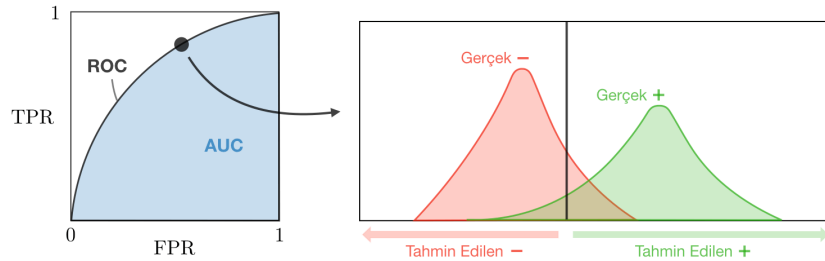
□ **Ana metrikler** – Sınıflandırma modellerinin performansını değerlendirmek için aşağıda verilen metrikler yaygın olarak kullanılmaktadır:

Metrik	Formül	Açıklama
Doğruluk	$\frac{TP + TN}{TP + TN + FP + FN}$	Modelin genel performansı
Kesinlik	$\frac{TP}{TP + FP}$	Doğru tahminlerin ne kadar kesin olduğu
Geri çağırma	$\frac{TP}{TP + FN}$	Gerçek pozitif örneklerin oranı
Specificity	$\frac{TN}{TN + FP}$	Gerçek negatif örneklerin oranı
F1 skoru	$\frac{2TP}{2TP + FP + FN}$	Dengesiz sınıflar için yararlı hibrit metrik

□ **İşlem Karakteristik Eğrisi (ROC)** – İşlem Karakteristik Eğrisi (receiver operating curve), eşik değeri değiştirilerek Doğru Pozitif Oranı-Yanlış Pozitif Oranı grafiğidir. Bu metrikler aşağıdaki tabloda özetlenmiştir:

Metrik	Formül	Eşdeğer
True Positive Rate TPR	$\frac{TP}{TP + FN}$	Geri çağırma
False Positive Rate FPR	$\frac{FP}{TN + FP}$	1-specificity

□ **Eğri Altında Kalan Alan (AUC)** – Aynı zamanda AUC veya AUROC olarak belirtilen işlem karakteristik eğrisi altındaki alan, aşağıdaki şekilde gösterildiği gibi İşlem Karakteristik Eğrisi (ROC)’nin altındaki alandır:



4.2 Regresyon metrikleri

□ **Temel metrikler** – Bir f regresyon modeli verildiğinde aşağıdaki metrikler genellikle modelin performansını değerlendirmek için kullanılır:

Toplam karelerinin toplamı	Karelerinin toplamının açıklaması	Karelerinin toplamından artanlar
$SS_{\text{tot}} = \sum_{i=1}^m (y_i - \bar{y})^2$	$SS_{\text{reg}} = \sum_{i=1}^m (f(x_i) - \bar{y})^2$	$SS_{\text{res}} = \sum_{i=1}^m (y_i - f(x_i))^2$

□ **Belirleme katsayısı** – Genellikle R^2 veya r^2 olarak belirtilen belirleme katsayısı, gözlemlenen sonuçların model tarafından ne kadar iyi kopyalandığının bir ölçütüdür ve aşağıdaki gibi tanımlanır:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

□ **Ana metrikler** – Aşağıdaki metrikler, göz önüne aldıkları değişken sayısını dikkate alarak regresyon modellerinin performansını değerlendirmek için yaygın olarak kullanılır:

Mallow's Cp	AIC	BIC	Adjusted R^2
$\frac{SS_{\text{res}} + 2(n+1)\hat{\sigma}^2}{m}$	$2[(n+2) - \log(L)]$	$\log(m)(n+2) - 2\log(L)$	$1 - \frac{(1-R^2)(m-1)}{m-n-1}$

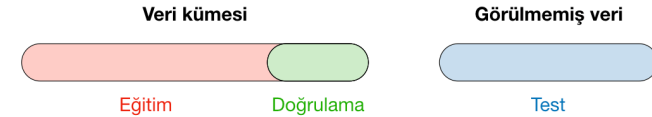
burada L olabilirlik ve $\hat{\sigma}^2$, her bir yanıtla ilişkili varyansın bir tahminidir.

4.3 Model seçimi

□ **Kelime Bilgisi** – Bir model seçerken, aşağıdaki gibi sahip olduğumuz verileri 3 farklı parçaya ayırırız:

Eğitim seti	Doğrulama seti	Test seti
- Model eğitildi - Genelde veri kümesinin %80'i	- Model değerlendirildi - Genelde veri kümesinin %20'si - Ayrıca doğrulama için bir kısmını bekletme veya geliştirme seti olarak da bilinir	- Model tahminleri gerçekleştiriyor - Görülmemiş veri

Model bir kere seçildikten sonra, tüm veri seti üzerinde eğitilir ve görünmeyen test setinde test edilir. Bunlar aşağıdaki şekilde gösterilmiştir:



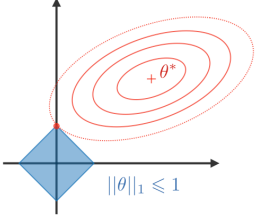
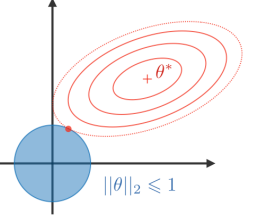
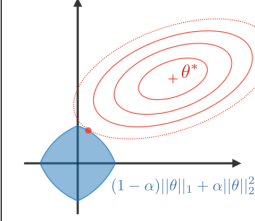
□ **Çapraz doğrulama** – Çapraz doğrulama, başlangıçtaki eğitim setine çok fazla güvenmeyen bir modeli seçmek için kullanılan bir yöntemdir. Farklı tipleri aşağıdaki tabloda özetlenmiştir:

k -fold	Leave- p -out
- $k - 1$ katı üzerinde eğitim ve geriye kalanlar üzerinde değerlendirme - Genel olarak $k = 5$ veya 10	- $n - p$ gözlemleri üzerine eğitim ve kalan p üzerinde değerlendirme - Durum $p = 1$ 'e bir tanesini dışarıda bırak denir

En yaygın olarak kullanılan yöntem k -kat çapraz doğrulama olarak adlandırılır ve $k - 1$ diğer katlarda olmak üzere, bu k sürelerinin hepsinde model eğitimi yapılırken, modeli bir kat üzerinde doğrulamak için eğitim verilerini k katlarına ayırır. Hata için daha sonra k -katlar üzerinden ortalama alınır ve çapraz doğrulama hatası olarak adlandırılır.

Kat	Veri kümesi	Doğrulama hatası	Çapraz doğrulama hatası
1		ϵ_1	
2		ϵ_2	
\vdots	\vdots	\vdots	
k		ϵ_k	$\frac{\epsilon_1 + \dots + \epsilon_k}{k}$

□ **Düzenleştirme (Regularization)** – Düzenleştirme prosedürü, modelin verileri aşırı öğrenmesinden kaçınılmasını ve dolayısıyla yüksek varyans sorunları ile ilgilenmeyi amaçlamaktadır. Aşağıdaki tablo, yaygın olarak kullanılan düzenleştirme tekniklerinin farklı türlerini özetlemektedir:

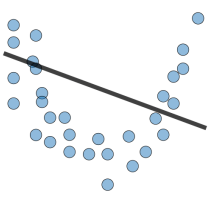
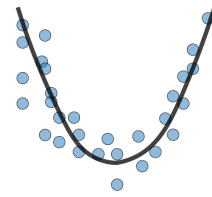
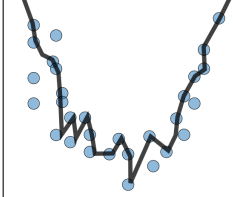
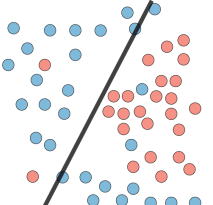
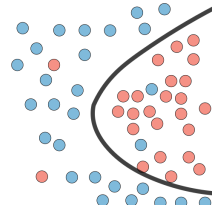
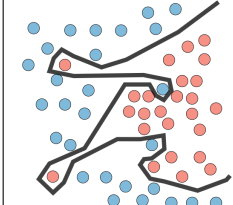
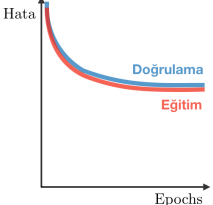
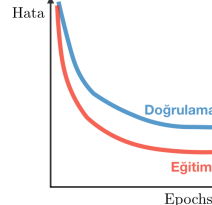
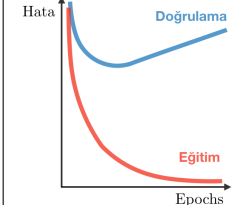
LASSO	Ridge	Elastic Net
<ul style="list-style-type: none"> - Değişkenleri 0'a kadar küçült - Değişken seçimi için iyi 	Katsayıları daha küçük yap	Değişken seçimi ile küçük katsayılar arasındaki çelişki
		
$\dots + \lambda \theta _1$ $\lambda \in \mathbb{R}$	$\dots + \lambda \theta _2^2$ $\lambda \in \mathbb{R}$	$\dots + \lambda \left[(1 - \alpha) \theta _1 + \alpha \theta _2^2 \right]$ $\lambda \in \mathbb{R}, \alpha \in [0, 1]$

4.4 Tanı

❑ **Önyargı** – Bir modelin önyargısı, beklenen tahmin ve verilen veri noktaları için tahmin etmeye çalıştığımız doğru model arasındaki farktır.

❑ **Varyans** – Bir modelin varyansı, belirli veri noktaları için model tahmininin değişkenliğidir.

❑ **Önyargı/varyans çelişkisi** – Daha basit model, daha yüksek önyargı, ve daha karmaşık model, daha yüksek varyans.

	Underfitting	Just right	Overfitting
Belirtiler	<ul style="list-style-type: none"> - Yüksek eğitim hatası - Test hatasına yakın eğitim hatası - Yüksek önyargı 	<ul style="list-style-type: none"> - Eğitim hatasından biraz daha düşük eğitim hatası 	<ul style="list-style-type: none"> - Çok düşük eğitim hatası - Eğitim hatası test hatasının çok altında - Yüksek varyans
Regresyon			
Sınıflandırma			
Derin öğrenme			
Olası çareler	<ul style="list-style-type: none"> - Model karmaşıktığında - Daha fazla özellik ekle - Daha uzun eğitim süresi ile eğit 		<ul style="list-style-type: none"> - Düzenleştirme gerçekleştir - Daha fazla bilgi edin

❑ **Hata analizi** – Hata analizinde mevcut ve mükemmel modeller arasındaki performans farkının temel nedeni analiz edilir.

❑ **Ablatif analiz** – Ablatif analizde mevcut ve başlangıç modelleri arasındaki performans farkının temel nedeni analiz edilir.

5 Hatırlatma

5.1 Olasılık ve İstatistik

Ayyüce Kızrak ve Başak Buluz tarafından çevrilmiştir

5.1.1 Olasılık ve Kombinasyonlara Giriş

□ **Örnek alanı** – Bir deneyin olası tüm sonuçlarının kümesidir, deneyin örnek alanı olarak bilinir ve S ile gösterilir.

□ **Olay** – Örnek alanın herhangi bir E alt kümesi, olay olarak bilinir. Yani bir olay, deneyin olası sonuçlarından oluşan bir kümedir. Deneyin sonucu E' 'de varsa, E 'nin gerçekleştiğini söyleriz.

□ **Olasılık aksiyomları** – Her E olayı için, E olayının meydana gelme olasılığı $P(E)$ olarak ifade edilir:

$$(1) \quad 0 \leq P(E) \leq 1 \quad (2) \quad P(S) = 1 \quad (3) \quad P\left(\bigcup_{i=1}^n E_i\right) = \sum_{i=1}^n P(E_i)$$

□ **Permütasyon** – Permütasyon, n nesneler havuzundan r nesnelerinin belirli bir sıra ile düzenlenmesidir. Bu tür düzenlemelerin sayısı $P(n, r)$ tarafından aşağıdaki gibi tanımlanır:

$$P(n, r) = \frac{n!}{(n-r)!}$$

□ **Kombinasyon** – Bir kombinasyon, sıranın önemli olmadığı n nesneler havuzundan r nesnelerinin bir düzenlemesidir. Bu tür düzenlemelerin sayısı $C(n, r)$ tarafından aşağıdaki gibi tanımlanır:

$$C(n, r) = \frac{P(n, r)}{r!} = \frac{n!}{r!(n-r)!}$$

Not: $0 \leq r \leq n$ için $P(n, r) \geq C(n, r)$ değerine sahibiz.

5.1.2 Koşullu Olasılık

□ **Bayes kuralı** – A ve B olayları için $P(B) > 0$ olacak şekilde:

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

Not: $P(A \cap B) = P(A)P(B|A) = P(A|B)P(B)$.

□ **Parça** – Tüm i değerleri için $A_i \neq \emptyset$ olmak üzere $\{A_i, i \in [1, n]\}$ olsun. $\{A_i\}$ bir parça olduğunu söyleriz eğer:

$$\forall i \neq j, A_i \cap A_j = \emptyset \quad \text{ve} \quad \bigcup_{i=1}^n A_i = S$$

Not: Örneklem uzaydaki herhangi bir B olayı için $P(B) = \sum_{i=1}^n P(B|A_i)P(A_i)$ 'ye sahibiz.

□ **Genişletilmiş Bayes kuralı formu** – $\{A_i, i \in [1, n]\}$ örneklem uzayının bir bölümü olsun. Elde edilen:

$$P(A_k|B) = \frac{P(B|A_k)P(A_k)}{\sum_{i=1}^n P(B|A_i)P(A_i)}$$

□ **Bağımsızlık** – İki olay A ve B birbirinden bağımsızdır ancak ve ancak eğer:

$$P(A \cap B) = P(A)P(B)$$

5.1.3 Rastgele Değişkenler

□ **Rastgele değişken** – Genellikle X olarak ifade edilen rastgele bir değişken, bir örneklem uzayındaki her öğeyi gerçek bir çizgiye eşleyen bir fonksiyondur.

□ **Kümülatif dağılım fonksiyonu (KDF)** – Monotonik olarak azalmayan ve $\lim_{x \rightarrow -\infty} F(x) = 0$ ve $\lim_{x \rightarrow +\infty} F(x) = 1$ olacak şekilde kümülatif dağılım fonksiyonu F şu şekilde tanımlanır:

$$F(x) = P(X \leq x)$$

Not: $P(a < X \leq B) = F(b) - F(a)$.

□ **Olasılık yoğunluğu fonksiyonu (OYF)** – Olasılık yoğunluğu fonksiyonu f , X 'in rastgele değişkenin iki bitişik gerçekleşmesi arasındaki değerleri alması ihtimalidir.

□ **OYF ve KDF'yi içeren ilişkiler** – Ayrık (D) ve sürekli (C) olaylarında bilmeniz gereken önemli özelliklerdir.

Olay	KDF F	OYF f	OYF Özellikleri
(D)	$F(x) = \sum_{x_i \leq x} P(X = x_i)$	$f(x_j) = P(X = x_j)$	$0 \leq f(x_j) \leq 1$ ve $\sum_j f(x_j) = 1$
(C)	$F(x) = \int_{-\infty}^x f(y)dy$	$f(x) = \frac{dF}{dx}$	$f(x) \geq 0$ ve $\int_{-\infty}^{+\infty} f(x)dx = 1$

□ **Varyans** – Genellikle $\text{Var}(X)$ veya σ^2 olarak ifade edilen rastgele değişkenin varyansı, dağılım fonksiyonunun yayılmasının bir ölçüsüdür. Aşağıdaki şekilde belirlenir:

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

□ **Standart sapma** – Genellikle σ olarak ifade edilen rastgele bir değişkenin standart sapması, gerçek rastgele değişkenin birimleriyle uyumlu olan dağılım fonksiyonunun yayılmasının bir ölçüsüdür. Aşağıdaki şekilde belirlenir:

$$\sigma = \sqrt{\text{Var}(X)}$$

□ **Beklenti ve Dağılım Momentleri** – Burada, ayrık ve sürekli durumlar için beklenen değer $E[X]$, genelleştirilmiş beklenen değer $E[g(X)]$, k . Moment $E[X^k]$ ve karakteristik fonksiyon $\psi(\omega)$ ifadeleri verilmiştir :

Olay	$E[X]$	$E[g(X)]$	$E[X^k]$	$\psi(\omega)$
(D)	$\sum_{i=1}^n x_i f(x_i)$	$\sum_{i=1}^n g(x_i) f(x_i)$	$\sum_{i=1}^n x_i^k f(x_i)$	$\sum_{i=1}^n f(x_i) e^{i\omega x_i}$
(C)	$\int_{-\infty}^{+\infty} x f(x) dx$	$\int_{-\infty}^{+\infty} g(x) f(x) dx$	$\int_{-\infty}^{+\infty} x^k f(x) dx$	$\int_{-\infty}^{+\infty} f(x) e^{i\omega x} dx$

□ **Rastgele değişkenlerin dönüşümü** – X ve Y değişkenlerinin bazı fonksiyonlarla bağlanır. f_X ve f_Y 'ye sırasıyla X ve Y 'nin dağılım fonksiyonu şöyledir:

$$f_Y(y) = f_X(x) \left| \frac{dx}{dy} \right|$$

□ **Leibniz integral kuralı** – g , x 'e ve potansiyel olarak c 'nin, c 'ye bağlı olabilecek potansiyel c ve a, b sınırlarının bir fonksiyonu olsun. Elde edilen:

$$\frac{\partial}{\partial c} \left(\int_a^b g(x) dx \right) = \frac{\partial b}{\partial c} \cdot g(b) - \frac{\partial a}{\partial c} \cdot g(a) + \int_a^b \frac{\partial g}{\partial c}(x) dx$$

□ **Chebyshev'in eşitsizliği** – X beklenen değeri μ olan rastgele bir değişken olsun. $k, \sigma > 0$ için aşağıdaki eşitsizliği elde edilir:

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

5.1.4 Ortak Dağılımlı Rastgele Değişkenler

□ **Koşullu yoğunluk** – Y 'ye göre X 'in koşullu yoğunluğu, genellikle $f_{X|Y}$ olarak elde edilir:

$$f_{X|Y}(x) = \frac{f_{XY}(x, y)}{f_Y(y)}$$

□ **Bağımsızlık** – İki rastgele değişkenin X ve Y olması durumunda bağımsız olduğu söylenir:

$$f_{XY}(x, y) = f_X(x) f_Y(y)$$

□ **Marjinal yoğunluk ve kümülatif dağılım** – f_{XY} ortak yoğunluk olasılık fonksiyonundan:

Olay	Marjinal yoğunluk	Kümülatif fonksiyon
(D)	$f_X(x_i) = \sum_j f_{XY}(x_i, y_j)$	$F_{XY}(x, y) = \sum_{x_i \leq x} \sum_{y_j \leq y} f_{XY}(x_i, y_j)$
(C)	$f_X(x) = \int_{-\infty}^{+\infty} f_{XY}(x, y) dy$	$F_{XY}(x, y) = \int_{-\infty}^x \int_{-\infty}^y f_{XY}(x', y') dx' dy'$

□ **Kovaryans** – σ_{XY}^2 veya daha genel olarak $\text{Cov}(X, Y)$ olarak elde ettiğimiz iki rastgele değişken olan X ve Y 'nin kovaryansını aşağıdaki gibi tanımlarız:

$$\text{Cov}(X, Y) \triangleq \sigma_{XY}^2 = E[(X - \mu_X)(Y - \mu_Y)] = E[XY] - \mu_X \mu_Y$$

□ **Korelasyon** – σ_X, σ_Y , X ve Y 'nin standart sapmalarını elde ederek, ρ_{XY} olarak belirtilen rastgele X ve Y değişkenleri arasındaki korelasyonu şu şekilde tanımlarız:

$$\rho_{XY} = \frac{\sigma_{XY}^2}{\sigma_X \sigma_Y}$$

Not: X, Y 'nin herhangi bir rastgele değişkeni için $\rho_{XY} \in [-1, 1]$ olduğuna dikkat edin. Eğer X ve Y bağımsızsa, $\rho_{XY} = 0$ olur.

□ **Ana dağıtımlar** – İşte akılda tutulması gereken ana dağıtımlar:

Tür	Dağılım	OYF	$\psi(\omega)$	$E[X]$	$\text{Var}(X)$
(D)	$X \sim \mathcal{B}(n, p)$ Binomial	$P(X = x) = \binom{n}{x} p^x q^{n-x}$ $x \in \llbracket 0, n \rrbracket$	$(pe^{i\omega} + q)^n$	np	npq
	$X \sim \text{Po}(\mu)$ Poisson	$P(X = x) = \frac{\mu^x}{x!} e^{-\mu}$ $x \in \mathbb{N}$	$e^{\mu(e^{i\omega} - 1)}$	μ	μ
(C)	$X \sim \mathcal{U}(a, b)$ Tekdüze	$f(x) = \frac{1}{b-a}$ $x \in [a, b]$	$\frac{e^{i\omega b} - e^{i\omega a}}{(b-a)i\omega}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
	$X \sim \mathcal{N}(\mu, \sigma)$ Gauss	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ $x \in \mathbb{R}$	$e^{i\omega\mu - \frac{1}{2}\omega^2\sigma^2}$	μ	σ^2
	$X \sim \text{Exp}(\lambda)$ Üstel	$f(x) = \lambda e^{-\lambda x}$ $x \in \mathbb{R}_+$	$\frac{1}{1 - \frac{i\omega}{\lambda}}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$

5.1.5 Parameter estimation

□ **Rastgele örnek** – Rastgele bir örnek, bağımsız ve aynı şekilde X ile dağıtılan X_1, \dots, X_n değişkeninin rastgele değişkenidir.

□ **Tahminci (Kestirimci)** – Tahmin edici, istatistiksel bir modelde bilinmeyen bir parametrenin değerini ortaya çıkarmak için kullanılan verilerin bir fonksiyonudur.

□ **Önyargı** – Bir tahmin edicinin önyargısı $\hat{\theta}$, $\hat{\theta}$ dağılımının beklenen değeri ile gerçek değer arasındaki fark olarak tanımlanır, yani:

$$\text{Bias}(\hat{\theta}) = E[\hat{\theta}] - \theta$$

Not: $E[\hat{\theta}] = \theta$ olduğunda bir tahmincinin tarafsız olduğu söylenir.

□ **Örnek ortalaması** – Rastgele bir numunenin numune ortalaması, dağılımın gerçek ortalamasını tahmin etmek için kullanılır, genellikle \bar{X} olarak belirtilir ve şöyle tanımlanır:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

□ **Örnek varyansı** – Rastgele bir örneğin örnek varyansı, bir dağılımın σ^2 gerçek varyansını tahmin etmek için kullanılır, genellikle s^2 veya $\hat{\sigma}^2$ olarak elde edilir ve aşağıdaki gibi tanımlanır:

$$s^2 = \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

□ **Merkezi Limit Teoremi** – Ortalama μ ve varyans σ^2 ile verilen bir dağılımın ardından rastgele bir X_1, \dots, X_n örneğine sahip olalım.

$$\bar{X} \underset{n \rightarrow +\infty}{\sim} \mathcal{N}\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

5.2 Doğrusal Cebir ve Kalkülüs

Kadir Tekeli ve Ekrem Çetinkaya tarafından çevrilmiştir

5.2.1 Genel notasyonlar

□ **Vektör** – i -inci elemanı $x_i \in \mathbb{R}$ olmak üzere n elemanlı bir vektör, $x \in \mathbb{R}^n$:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

□ **Matris** – $A_{i,j} \in \mathbb{R}$ i -inci satır ve j -inci sütundaki elemanları olmak üzere m satırlı ve n sütunlu bir matris, $A \in \mathbb{R}^{m \times n}$:

$$A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

Dipnot: Yukarıda tanımlanan x vektörü $n \times 1$ tipinde bir matris olarak ele alınabilir ve genellikle sütun vektörü olarak adlandırılır.

□ **Birim matris** – Birim matris, köşegeni birlerden ve diğer tüm elemanları sıfırlardan oluşan karesel matris, $I \in \mathbb{R}^{n \times n}$:

$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Dipnot: Her $A \in \mathbb{R}^{n \times n}$ matrisi için $A \times I = I \times A = A$ eşitliği sağlanır.

□ **Köşegen matris** – Bir köşegen matris, köşegenindeki elemanları sıfırdan farklı diğer tüm elemanları sıfır olan karesel matris, $D \in \mathbb{R}^{n \times n}$:

$$D = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_n \end{pmatrix}$$

Dipnot: D matrisi $\text{diag}(d_1, \dots, d_n)$ olarak da gösterilir.

5.2.2 Matris işlemleri

Çarpma

□ **Vektör-vektör** – İki çeşit vektör-vektör çarpımı vardır.

- iç çarpım: $x, y \in \mathbb{R}^n$ için:

$$x^T y = \sum_{i=1}^n x_i y_i \in \mathbb{R}$$

- dış çarpım: $x \in \mathbb{R}^m, y \in \mathbb{R}^n$ için:

$$xy^T = \begin{pmatrix} x_1 y_1 & \cdots & x_1 y_n \\ \vdots & & \vdots \\ x_m y_1 & \cdots & x_m y_n \end{pmatrix} \in \mathbb{R}^{m \times n}$$

□ **Matris-vektör** – $A \in \mathbb{R}^{m \times n}$ matrisi ve $x \in \mathbb{R}^n$ vektörünün çarpımları \mathbb{R}^m boyutunda bir vektördür:

$$Ax = \begin{pmatrix} a_{r,1}^T x \\ \vdots \\ a_{r,m}^T x \end{pmatrix} = \sum_{i=1}^n a_{c,i} x_i \in \mathbb{R}^m$$

burada $a_{r,i}^T$ A 'nın vektör satırları ve $a_{c,j}$ A 'nın vektör sütunları ve x_i x vektörünün elemanlarıdır.

□ **Matris-matris** – $A \in \mathbb{R}^{m \times n}$ matrisi ve $B \in \mathbb{R}^{n \times p}$ matrisinin çarpımları $\mathbb{R}^{m \times p}$ boyutunda bir matristir:

$$AB = \begin{pmatrix} a_{r,1}^T b_{c,1} & \cdots & a_{r,1}^T b_{c,p} \\ \vdots & & \vdots \\ a_{r,m}^T b_{c,1} & \cdots & a_{r,m}^T b_{c,p} \end{pmatrix} = \sum_{i=1}^n a_{c,i} b_{r,i}^T \in \mathbb{R}^{m \times p}$$

burada $a_{r,i}^T, b_{r,i}^T$ sırasıyla A ve B 'nin vektör satırları ve $a_{c,j}, b_{c,j}$ sırasıyla A ve B 'nin vektör sütunlarıdır.

Diğer işlemler

□ **Devrik (Transpoze)** – Bir $A \in \mathbb{R}^{m \times n}$ matrisinin devriği, satır ve sütunların yer değiştirmesi ile elde edilir, ve A^T ile gösterilir:

$$\forall i, j, \quad A_{i,j}^T = A_{j,i}$$

Dipnot: Her A, B için $(AB)^T = B^T A^T$ vardır.

□ **Ters** – Tersinir bir A karesel matrisinin tersi, aşağıdaki koşulu sağlayan matristir, ve A^{-1} ile gösterilir:

$$AA^{-1} = A^{-1}A = I$$

Dipnot: Her karesel matris tersinir değildir. Ayrıca, Her tersinir A, B matrisi için $(AB)^{-1} = B^{-1}A^{-1}$ dir.

□ **İz** – Bir A karesel matrisinin izi, köşegenindeki elemanlarının toplamıdır, ve $\text{tr}(A)$ ile gösterilir:

$$\text{tr}(A) = \sum_{i=1}^n A_{i,i}$$

Dipnot: A, B matrisleri için $\text{tr}(A^T) = \text{tr}(A)$ ve $\text{tr}(AB) = \text{tr}(BA)$ vardır.

□ **Determinant** – $A \in \mathbb{R}^{n \times n}$ matrisinin determinantı, $A_{\setminus i, \setminus j}$ gösterimi i -inci satırsız ve j -inci sütunsuz şekilde A matrisi olmak üzere özyinelemeli olarak aşağıdaki gibi ifade edilir, ve $|A|$ ya da $\det(A)$ ile gösterilir:

$$\det(A) = |A| = \sum_{j=1}^n (-1)^{i+j} A_{i,j} |A_{\setminus i, \setminus j}|$$

Dipnot: A tersinirdir ancak ve ancak $|A| \neq 0$. Ayrıca, $|AB| = |A||B|$ ve $|A^T| = |A|$.

5.2.3 Matris özellikleri

Tanımlar

□ **Simetrik ayrışım** – Verilen bir A matrisi simetrik ve ters simetrik parçalarının cinsinden aşağıdaki gibi ifade edilebilir:

$$A = \underbrace{\frac{A + A^T}{2}}_{\text{Simetrik}} + \underbrace{\frac{A - A^T}{2}}_{\text{Ters simetrik}}$$

□ **Norm** – V vektör uzayı ve her $x, y \in V$ için aşağıdaki özellikleri sağlayan $N : V \rightarrow [0, +\infty[$ fonksiyonu bir normdur:

- $N(x + y) \leq N(x) + N(y)$
- Bir a sabiti için $N(ax) = |a|N(x)$
- $N(x) = 0$ ise $x = 0$

$x \in V$ için en yaygın şekilde kullanılan normlar aşağıdaki tabloda verilmektedir.

Norm	Notation	Definition	Use case
Manhattan, L^1	$\ x\ _1$	$\sum_{i=1}^n x_i $	LASSO regularization
Euclidean, L^2	$\ x\ _2$	$\sqrt{\sum_{i=1}^n x_i^2}$	Ridge regularization
p -norm, L^p	$\ x\ _p$	$\left(\sum_{i=1}^n x_i^p\right)^{\frac{1}{p}}$	Hölder inequality
Infinity, L^∞	$\ x\ _\infty$	$\max_i x_i $	Uniform convergence

□ **Doğrusal bağımlılık** – Bir vektör kümesinden bir vektör diğer vektörlerin doğrusal birleşimi (kombinasyonu) cinsinden yazılabiliyorsa bu vektör kümesine doğrusal bağımlı denir.

Dipnot: Eğer bu şekilde yazılabilen herhangi bir vektör yoksa bu vektörlere doğrusal bağımsız denir.

□ **Matris rankı** – Verilen bir A matrisinin rankı, $\text{rank}(A)$, bu matrisinin sütunları tarafından üretilen vektör uzayının boyutudur. Bu ifade A matrisinin doğrusal bağımsız sütunlarının maksimum sayısına denktir.

□ **Pozitif yarı-tanımlı matris** – Aşağıdaki koşulu sağlayan bir $A \in \mathbb{R}^{n \times n}$ matrisi pozitif yarı-tanımlıdır ve $A \succeq 0$ ile gösterilir:

$$A = A^T \quad \text{and} \quad \forall x \in \mathbb{R}^n, \quad x^T A x \geq 0$$

Dipnot: Benzer olarak, pozitif yarı-tanımlı bir A matrisi sıfırdan farklı her x vektörü için $x^T A x > 0$ koşulunu sağlıyorsa A matrisine pozitif tanımlı denir ve $A \succ 0$ ile gösterilir.

□ **Özdeğer, özvektör** – Verilen bir $A \in \mathbb{R}^{n \times n}$ için aşağıdaki gibi bir $z \in \mathbb{R}^n \setminus \{0\}$ vektörü var ise buna özvektör, λ sayısına da A matrisinin öz değeri denir.

$$Az = \lambda z$$

Teorem

□ **Spektral teorem** – $A \in \mathbb{R}^{n \times n}$ olsun. Eğer A simetrik ise, A matrisi gerçel ortogonal $U \in \mathbb{R}^{n \times n}$ matrisi ile köşegenleştirilebilir. $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ olmak üzere:

$$\exists \Lambda \text{ köşegen, } A = U \Lambda U^T$$

□ **Tekil-değer ayrışımı** – $m \times n$ tipindeki bir A matrisi için tekil-değer ayrışımı; $m \times m$ tipinde bir üniter U , $m \times n$ tipinde bir köşegen Σ ve $n \times n$ tipinde bir üniter V matrislerinin varlığını garanti eden bir parçalama tekniğidir.

$$A = U \Sigma V^T$$

Matris kalkülüsü

□ **Gradyan** – $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ bir fonksiyon ve $A \in \mathbb{R}^{m \times n}$ bir matris olsun. f nin A ya göre gradyanı $m \times n$ tipinde bir matristir, ve $\nabla_A f(A)$ ile gösterilir:

$$\left(\nabla_A f(A) \right)_{i,j} = \frac{\partial f(A)}{\partial A_{i,j}}$$

Dipnot: f fonksiyonunun gradyanı yalnızca f skaler döndüren bir fonksiyon ise tanımlıdır.

□ **Hessian** – $f : \mathbb{R}^n \rightarrow \mathbb{R}$ bir fonksiyon ve $x \in \mathbb{R}^n$ bir vektör olsun. f fonksiyonun x vektörüne göre Hessian'ı $n \times n$ tipinde bir simetrik matristir, ve $\nabla_x^2 f(x)$ ile gösterilir:

$$\left(\nabla_x^2 f(x) \right)_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

Dipnot: f fonksiyonunun Hessian'ı yalnızca f skaler döndüren bir fonksiyon ise tanımlıdır.

□ **Gradyan işlemleri** – A, B, C matrisleri için aşağıdaki işlemlerin akılda bulunmasında fayda vardır:

$$\nabla_A \text{tr}(AB) = B^T$$

$$\nabla_{A^T} f(A) = (\nabla_A f(A))^T$$

$$\nabla_A \text{tr}(ABA^T C) = CAB + C^T AB^T$$

$$\nabla_A |A| = |A| (A^{-1})^T$$