Makine Öğrenmesi-2

Akış

- Makine Öğrenmesi nedir?
- Günlük Hayatımızdaki Uygulamaları
- Verilerin Sayısallaştırılması
- **Özellik Belirleme**
  - **Özellik Seçim Metotları**
    - **Bilgi Kazancı (Informaiton Gain-IG)**
    - **Sinyalin Gürültüye Oranı: (S2N ratio)**
    - **Alt küme seçiciler (Wrappers)**
  - **Yeni Özelliklerin Çıkarımı**
    - **Temel Bileşen Analizi (Principal Component Analysis)**
    - **Doğrusal Ayırteden Analizi (Linear Discriminant Analysis)**
- **Sınıflandırma Metotları**
  - **Doğrusal Regresyon**
  - **Karar Ağaçları (Decision Trees)**
  - **Yapay Sinir Ağları**
  - **En Yakın K Komşu Algoritması (k - Nearest Neighbor)**
  - **Öğrenmeli Vektör Kuantalama (Learning Vector Quantization)**
- **Kümeleme Algoritmaları:**
  - **Hiyerarşik Kümeleme**
  - **K-means**
  - **Kendi Kendini Düzenleyen Haritalar (Self Organizing Map -SOM)**
  - **DBscan**
- **Regresyon Algoritmaları**
- **Çok Boyutlu Verilerle Çalışmak**
- **Veri Sızıntısı**
- Pekiştirmeli Öğrenme

Genel Dataset Yapısı
supervised (sınıflandırma / regresyon)

öz1	öz2	…	Öz d	Çıkış
Örnek1 öz1	Örnek1 öz2	…	Örnek 1 öz d	Örnek1 çıkış
Örnek2 öz1	Örnek2 öz2	…	Örnek 2 öz d	Örnek2 çıkış
ÖrnekN öz1	ÖrnekN öz2	…	Örnek N öz d	ÖrnekN çıkış

Özellik tipleri

- Kategorik (K adet ten 1 i)
- Sayı

Özellik Belirleme

- Bir doktor
- Veri: Kişi bilgilerini içeren dosyalar
- Görev: Kimler hasta bul.
- Hangi bilgilere bakılır?
  - Ad soyad
  - Doğum yeri
  - Cinsiyet
  - Kan tahlili sonuçları
  - Röntgen sonuçları
  - vs.

1. Öz	2. Öz	3. Öz	Sınıf
1	3	13	A
2	3	13	B
1	4	13	A
2	3	13	B

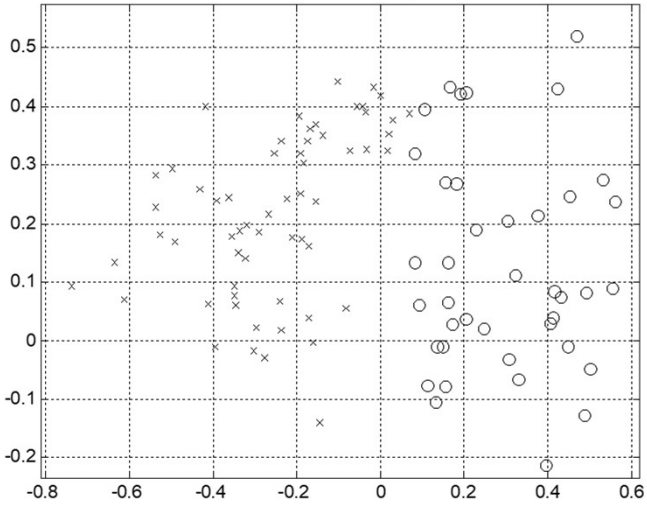
İf 1.öz==1 then çıkış=A
Else çıkış=B (%100)

İf 2.öz==4 then çıkış=A
Else çıkış=B (%) 75

çıkış=B (%50)

Hangi boyut?

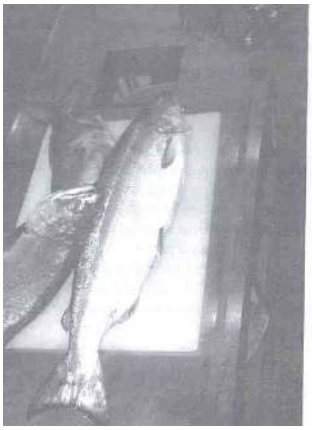
öz2

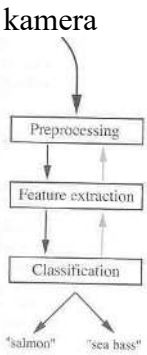


İf öz1>0.1 then çıkış=o else çıkış=x

Balık Hali

- Kayan bant üzerindeki balığın türünü belirlemek
- Salmon / Sea Bass
  Somon / Levrek

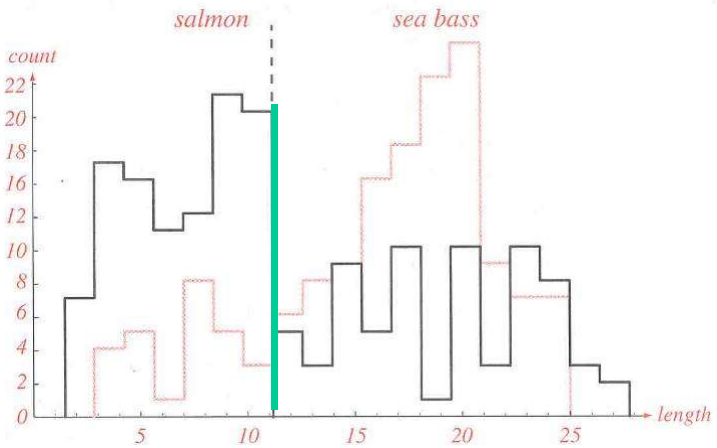




[*] Pattern Classification (2nd Edition) by Richard O. Duda,Peter E. Hart,David G. Stork

Balık Özellikleri: Uzunluk.

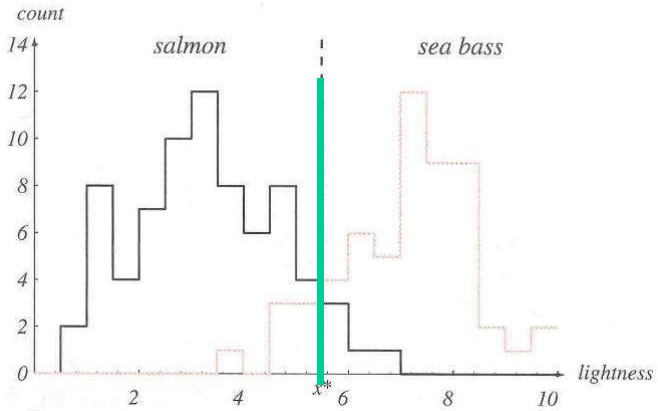
- Salmon lar genelde Sea Bass lardan daha kısalar.



[*] Pattern Classification (2nd Edition) by Richard O. Duda,Peter E. Hart,David G. Stork

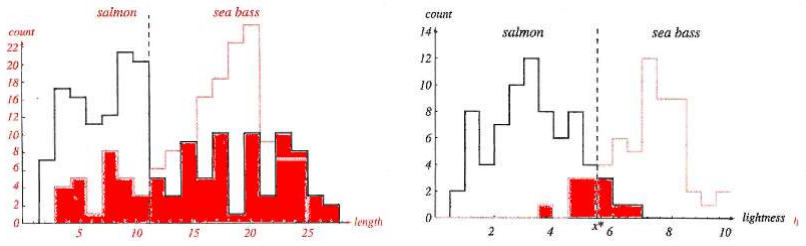
Balık Özellikleri: Parlaklık.

- Sea Bass genelde Salmon lardan daha parlaklar.



[*] Pattern Classification (2nd Edition) by Richard O. Duda,Peter E. Hart,David G. Stork

Hangi Özellik?



Kırmızı bölgeler yapılan hataları gösteriyor.

[*] Pattern Classification (2nd Edition) by Richard O. Duda,Peter E. Hart,David G. Stork

Doktoru yoralım ☺

- Hastalık dosyasında 5000 adet özellik olsaydı? Örneğin kişinin DNA dizisine bakarak hasta olup olmadığına karar verecek olsaydık ne yapardık? Nerelere bakacağımıza nasıl karar verirdik.
- Burada devreye makineleri sokmamız gerekiyor gibi gözükmekte.
- Bu olay bir insanın hesap yapma kabiliyetiyle, bir hesap makinesininkini karşılaştırmaya benziyor.

Özellik seçimi

- Bu problem makinelerle iki farklı metotla çözülebilir.
  - Var olan özelliklerden bazılarını seçmek
    - Özellikleri tek tek değerlendirmek (Filter)
    - Özellik alt kümeleri oluşturup, sınıflandırıcılar kullanıp performanslarını ölçüp, bu alt kümeleri en iyilemek için değiştirerek (Wrapper)
  - Var olan özelliklerin lineer birleşimlerinden yeni özelliklerin çıkarımı

Özellikleri birer birer inceleme (Filters)

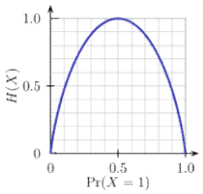
- Eğitim bilgilerindeki her bir özellik teker teker ele alınır.
- Örnek ile ilgili sadece o özellik elimizde olsaydı ne olurdu sorusunun cevabı bulunmaya çalışılır.
- Seçilen özellikle sınıf ya da sonucun birlikte değişimleri incelenir.
- Özellik değiştiğinde sınıf ya da sonuç ne kadar değişiyorsa o özelliğin sonuca o kadar etkisi vardır.

Bilgi Kazancı - Information Gain

S eğitim seti içindeki A özelliğinin

$$Gain(S, A) \equiv Entrophy(S) - \sum_{v \epsilon Value(A)} \frac{|S_v|}{|S|} Entrophy(S_v)$$

N kavramının c farklı değeri varsa N'in entropisi, N'in aldığı her değerin olasılıkları kullanılarak



$$\text{Entropy (N)} = \sum_{i=1}^{c} - p_i \log_2 p_i$$

[*] Machine Learning, Tom Mitchell

Entropy (N)= $\sum_{i=1}^{c} - p_i \log_2 p_i$

Entropi(3.öz) = - 1*log2(1) = 0

C=1
P(13)=1

Entropi(sınıf) = - [(1/2)*log2(1/2)+(1/2)*log2(1/2)] = - [(1/2)*-1+(1/2)*-1] = 1

C=2
P(A)=1/2
P(B)=1/2

3. Öz	Sınıf
13	A
13	B
13	A
13	B

K=A A A B B D için
C=3
Entropi(K)= - (1/2*log2(1/2)+ 1/3*log2(1/3)+1/6*log2(1/6))
Pa=1/2
Pb=1/3
Pd=1/6

daha önceki hava, nem, rüzgar, su sıcaklığı gibi değerlere göre pikniğe gidip gitmeme kararı verilmiş 4 olay

Olay No	Hava	Nem	Rüzgar	Su sıcaklığı	Pikniğe gidildi mi?
1	güneşli	normal	güçlü	ılık	Evet
2	güneşli	yüksek	güçlü	ılık	Evet
3	yağmurlu	yüksek	güçlü	ılık	Hayır
4	güneşli	yüksek	güçlü	soğuk	Evet

Her bir özelliğin piknik kavramı için bilgi kazancını bulalım

- Pikniğe gidildi mi? sorusunun iki cevabı vardır.
- Evet cevabının olasılığı ¾
- Hayır cevabının olasılığı ¼
- Dolayısıyla Pikniğin Entropi'si
- **E(Piknik)** = -(3/4) log2(3/4) - (1/4) log2(1/4) = **0.811 olarak** bulunur.

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- **Gain(Piknik,Hava)**= 0.811- (3/4) (-(3/3) log2 (3/3) – 0) – (1/4) (0 – (1/1) log2 (1/1)) = **0.811**
- Hava özelliğinin IG'si hesaplanırken bulunan rakamların açıklamaları:

0.811 → Pikniğe gitme olayının Entropisi
(3/4) → havanın güneşli olma oranı
(3/3) → hava güneşli iken pikniğe gidilme oranı
0 → hava güneşli iken pikniğe gidilmeme oranı
(1/4) → havanın yağmurlu olma oranı
0 → hava yağmurlu iken pikniğe gidilme oranı
(1/1) → hava yağmurlu iken pikniğe gidilmeme oranı

- **Gain(Piknik,Nem)**= 0.811- (1/4) (-(1/1) log2 (1/1) - 0) – (3/4) (-(2/3) log2(2/3)-(1/3) log2(1/3))
  = 0.811 -0.688= **0.1225**
- **Gain(Piknik,Rüzgar)**= 0.811- (4/4) (-(3/4) log2(3/4) – (1/4) log2(1/4))
  = 0.811 -0.811= **0**
- **Gain(Piknik,SuSıcaklığı)**= 0.811- (3/4) (-(2/3) log2(2/3) – (1/3) log2(1/3)) – (1/4) (-(1/1) log2 (1/1))
  = 0.811 -0.688= **0.1225**
- En büyük bilgi kazancına sahip özellik 'Hava'dır.
- Gerçek uygulamalarda ise yüzlerce özelliğin bilgi kazançları hesaplanır ve en büyük olanları seçilerek kullanılır.

Signal2Noise Ratio (Sinyalin Gürültüye Oranı)

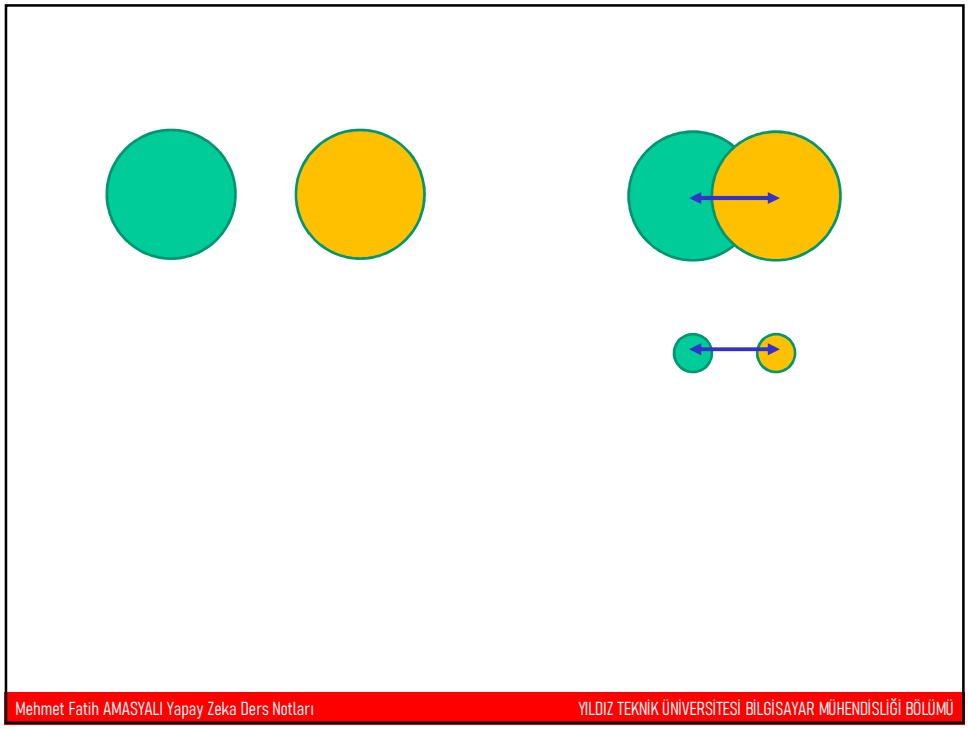
- Sınıflar arası ayrılıkların fazla sınıf içi ayrılıkların az olan özellikler seçilir.

$$S_i = \frac{|m_1 - m_2|}{d_1 + d_2}$$

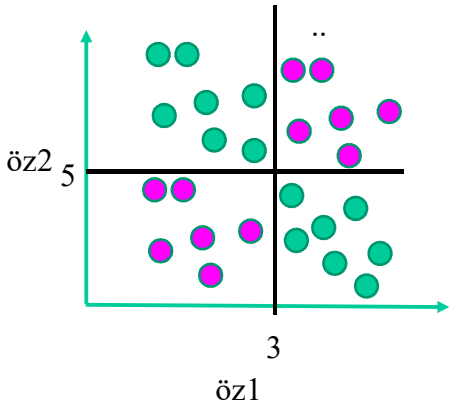
Sinyal

Gürültü

m1→ sınıf1'deki i. özelliklerin ortalaması
m2→ sınıf2'deki i. özelliklerin ortalaması
d1→ sınıf1'deki i. özelliklerin standart sapması
d2→ sınıf2'deki i. özelliklerin standart sapması
S değeri en yüksek olan özellikler seçilerek sınıflandırmada kullanılırlar.

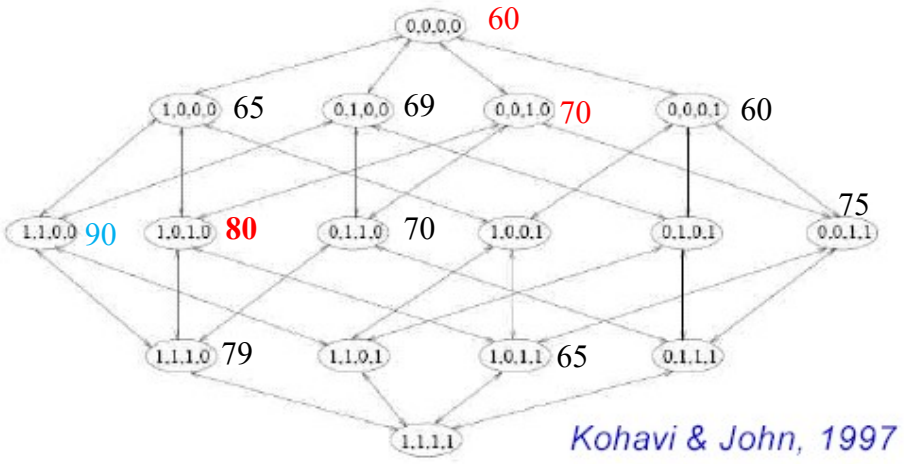


İf öz1<3 and öz2<5 then çıkış=pembe
Else if öz1<3 and öz2>=5 then çıkış= yeşil
Else if ..
..



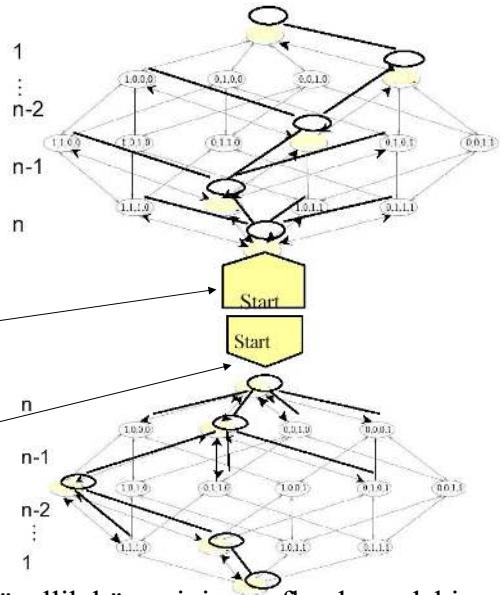
N özellik için olası 2^N özellik alt kümesi = 2^N eğitim

Wrappers (Özellik altkümesi seçiciler)



Özellik altkümesi seçiciler

- *Hızlandırmak için tüm olasılıkları denemek yerine*
  - Hepsiyle başlayıp her seferinde bir tane elemek
  - Tek özellikle başlayıp her seferinde bir tane eklemek



Hangi yoldan gidileceğine o özellik kümesinin sınıflandırmadaki performansına bakılarak karar verilir.

[*] http://clopinet.com/isabelle/Projects/ETH/lecture9.pdf

Yeni Özelliklerin Çıkarımı

- Var olan özelliklerin lineer birleşimlerinden yeni bir özellik uzayı oluşturulur ve veriler bu uzayda ifade edilirler. Yaygın olarak kullanılan 2 metot vardır.

- PCA
- LDA

Orj özellikler: x1, x2, x3, x4
Yeni özellikler:
Y1=w11*x1+w12*x2+w13*x3+w14*x4+w15
Y2=w21*x1+w22*x2+w23*x3+w24*x4+w25
Y3=w31*x1+w32*x2+w33*x3+w34*x4+w35
Y4=w41*x1+w42*x2+w43*x3+w44*x4+w45

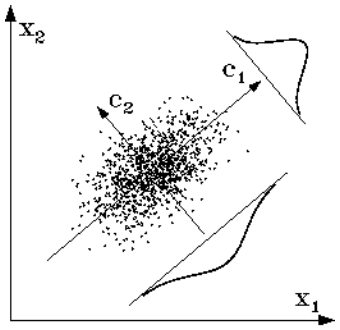
öz1	öz2
3	1
2	5
3	6

Y1	Y2
18	7
25	22
33	27

w11=5 w12=3 w13=0
w21=1 w22=4 w23=0
Y1=w11*öz1+w12*öz2+w13
Y2=w21*öz1+w22*öz2+w23
Y1=5*öz1+3*öz2+0
Y2=1*öz1+4*öz2+0

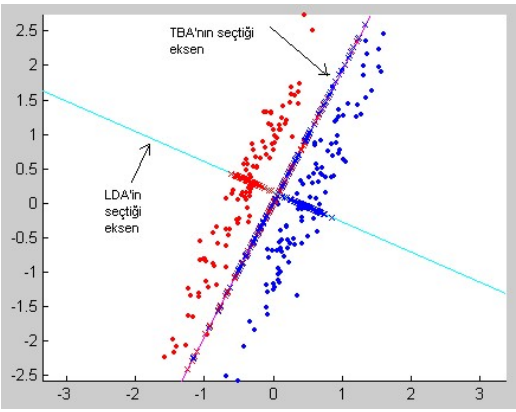
Temel Bileşen Analizi-TBA (Principle Component Analysis - PCA)

- Bu metotta örneklerin en fazla değişim gösterdikleri boyutlar bulunur. Yanda veriler c1 ve c2 eksenlerine izdüşümü yapıldığındaki dağılımları gösterilmiştir.
- C1 eksenindeki değişim daha büyüktür. Dolayısıyla veriler 2 boyuttan bir boyuta C1 ekseni üzerine iz düşürülerek indirgenmiş olur.



Doğrusal Ayırteden Analizi (Linear Discriminant Analysis - LDA)

Yandaki gibi durumlar için LDA önerilmiştir. LDA varyanslara ek olarak sınıf bilgisini de kullanarak boyut indirgene yapar. Sadece varyansa değil sınıflandırabilmeye de bakar.

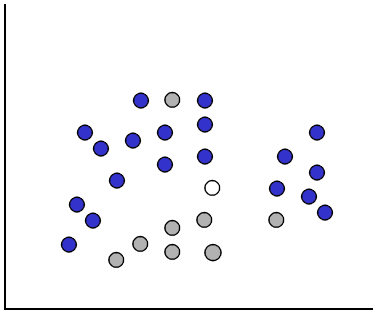


Hangisi

- Niye bu kadar çok metot var?
- Ne zaman hangisini kullanacağız?

Sınıflandırma Metotları

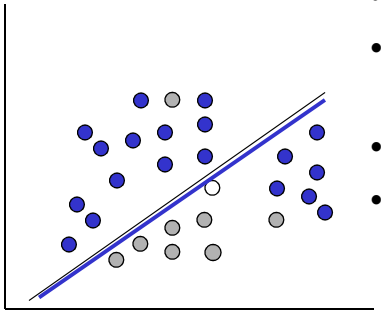
Görev: Önceden etiketlenmiş örnekleri kullanarak yeni örneklerin sınıflarını bulmak



Metotlar:
Regresyon,
Karar Ağaçları,
LVQ,
Yapay Sinir Ağları,
...

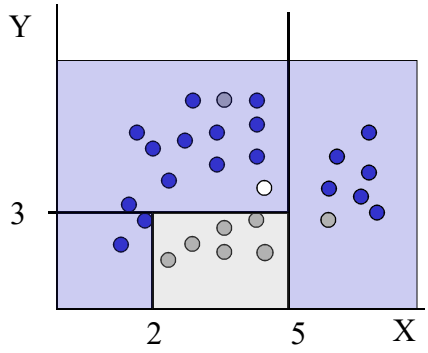
Mavi ve gri sınıftan örnekler ● ●
Beyaz, mavi mi gri mi? ○

Doğrusal Regresyon



- $w_0 + w_1 x1 + w_2 x2 + \ldots + w_d xd >= 0$
- En az hata yapan w_i leri bulmaya çalışır.
- Basit bir model
- Yeterince esnek değil

Karar Ağaçları



Böl ve yönet stratejisi

Nasıl böleceğiz?

if X > 5 then blue
else if Y > 3 then blue
else if X > 2 then green
else blue

Ürettikleri kurallar anlaşılır.
Karar düğümleri ve yapraklardan oluşan hiyerarşik bir yapı.

Karar Ağaçları Oluşturma

- Tüm veri kümesiyle başla.
- Bir özelliğin bir değerlerine göre veri kümesi iki alt kümeye böl. Bölmede kullanılan özellikler ve değerleri karar düğüme yerleştir.
- Her alt küme için aynı prosedür her alt kümede sadece tek bir sınıfa ait örnekler kalıncaya kadar uygula.

Karar ağacı oluşturalım

5	o	x	x	x
4	o	x	x	x
3	o	o	o	x
2	o	o	o	x
1	x	x	o	o
	1	2	3	4

X1 < 1.5 e1
X1 < 2.5 e2
X1 < 3.5 e3
X2 < 1.5 e4
X2 < 2.5 e5
X2 < 3.5 e6
X2 < 4.5 e7

X2 < 1.5 e8
X2 < 2.5 e9
X2 < 3.5 e10
X2 < 4.5 e11

X1 < 2.5 e29
X1 < 3.5 e39

X1 < 2.5 e28
X1 < 3.5 e38
X2 < 2.5 e58

X1 < 2.5 e27
X1 < 3.5 e37
X2 < 1.5 e47
X2 < 2.5 e57

X1 < 2.5 e26
X1 < 3.5 e36
X2 < 1.5 e46
X2 < 2.5 e56
X2 < 3.5 e66
X2 < 4.5 e76

Karar Düğümleri Nasıl Bulunur?

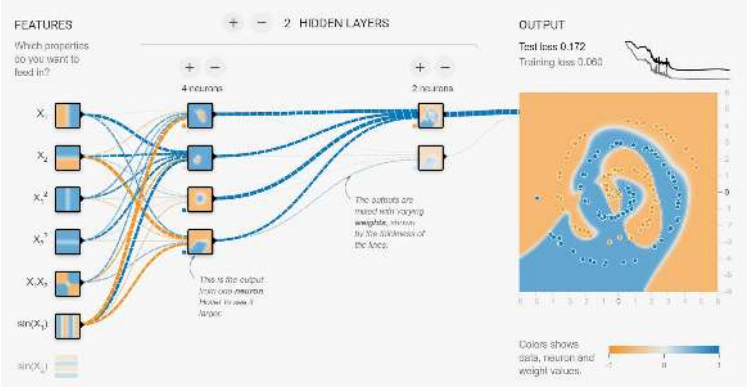
- Karar düğümlerinde yer alan özelliğin ve eşik değerinin belirlenmesinde genel olarak **entropi** kavramı kullanılır.
- Eğitim verisi her bir özelliğin her bir değeri için ikiye bölünür. Oluşan iki alt kümenin entropileri toplanır. En düşük entropi toplamına sahip olan özellik , değer ikilisi karar düğümüne yerleştirilir.

Karar Ağaçlarıyla Sınıflandırma

- En tepedeki kök karar düğümünden başla.
- Bir yaprağa gelinceye kadar karar düğümlerindeki yönlendirmelere göre dallarda ilerle. (Karar düğümlerinde tek bir özelliğin adı ve bir eşik değeri yer alır. O düğüme gelen verinin hangi dala gideceğine verinin o düğümdeki özelliğinin eşik değerinden büyük ya da küçük olmasına göre karar verilir.)
- Verinin sınıfı, yaprağın temsil ettiği sınıf olarak belirle.

Yapay Sinir Ağları

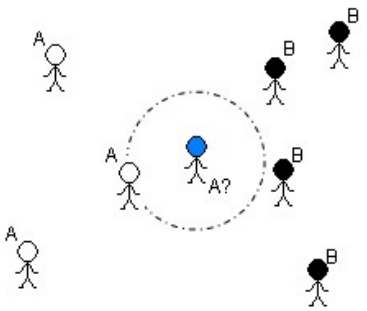
- Daha kompleks karar sınırlar üretebilirler*
- Elimizdeki Eğitim seti Girişler ve Çıkışlar ı içerir. Bu girişler verildiğinde bu çıkışları verecek ağırlık değerlerini (W) bulmaya çalışır.



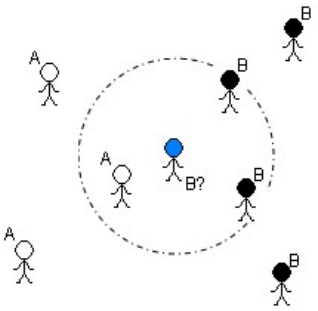
[*] https://playground.tensorflow.org/

En Yakın K Komşu

- Bana Arkadaşını söyle, sana kim olduğunu söyleyeyim.



En yakın 1 komşu



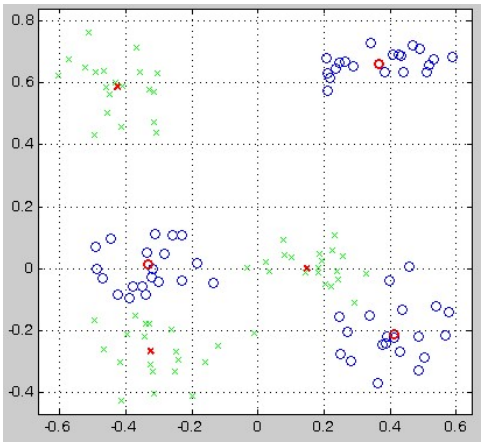
En yakın 3 komşu

En Yakın K Komşu

- Eğitim yok.
- Test verileri en yakınlarındaki K adet komşularının sınıf değerlerine bakılarak sınıflandırılırlar.

1NN'deki Tüm Noktaları kullanmasak?

- Tüm noktalar yerine yeni/ seçilmiş noktalar
- Bu noktalar nasıl belirlenebilir?



Öğrenmeli Vektör Kuantalama (Learning Vector Quantization)

[η] öğrenme oranı

[n] maximum eğitim sayısı

[c] betimleyici vector sayısı

[$\mu_1,\ldots,\mu_c$] betimleyici vektörler (centroids)

[x] eğitim datasından bir örnek

[S(x)] x vektörünün ait oldugu yada betimledigi sınıf

olmak üzere

1. η,n, $\mu_1,\ldots,\mu_c$ için ilk değer atamalarını gerçekleştir
2. Aşağıdaki işlemleri n defa tekrar et
   2.1 X eğitim datasını al
   2.2 X e en yakın betimleyici vektörü bul

$(\mu_k): k \leftarrow \arg\min_j \|x-\mu_j\|$ j=1..c

2.3 μ_k nın güncellenmesi:

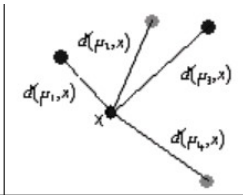
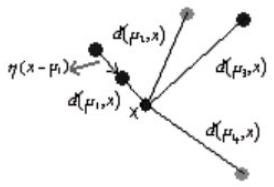
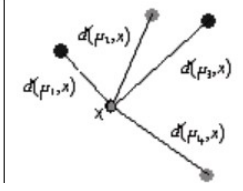
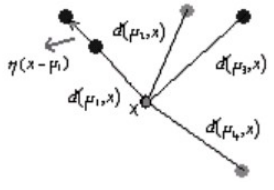
Eğer x doğru sınıfsa (s(x)=s(μ_k) sınıfları aynı ise)

$\mu_k \leftarrow \mu_k + \eta(x-\mu_k)$ ödüllendir x'e yaklaştır

değilse

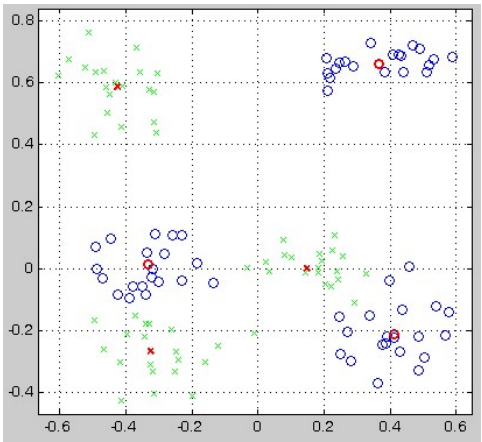
$\mu_k \leftarrow \mu_k - \eta(x-\mu_k)$ cezalandır x'den uzaklaştır

LVQ'da eğitim

LVQ'da ödüllendirme Kazanan vektörle, örnek aynı sınıftan (ikisi de siyah sınıftan)		
LVQ'da cezalandırma Kazanan vektörle, örnek farklı sınıflardan (kazanan siyah, örnek gri sınıftan)		

LVQ- Test İşlemi

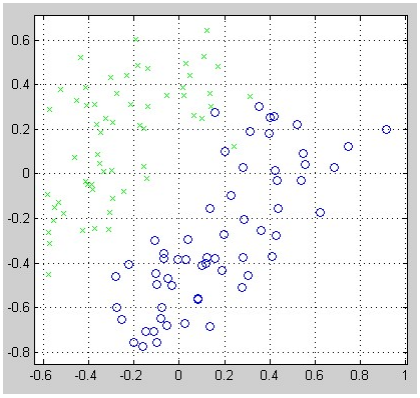
- Eğitim sonucu bulunan 2 sınıfa ait 3'er betimleyici vektör.
- Test işlemi, test örneğinin bu 6 vektörden en yakın olanının sınıfına atanmasıdır.



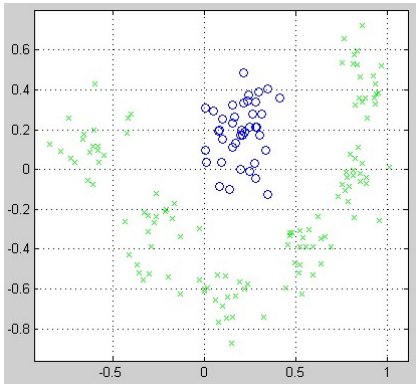
- LVQ ne zaman lineer/doğrusal karar sınırı üretir?

Hangisi?

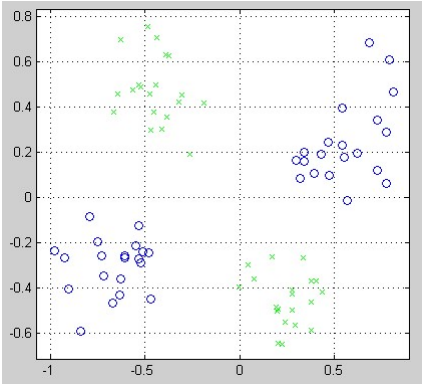
Lineer Regresyon
Karar Ağaçları
KNN
YSA
LVQ



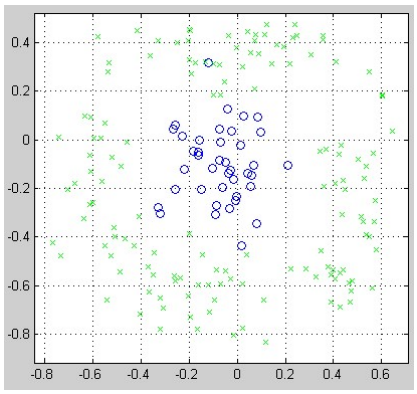
Lineer Regresyon
Karar Ağaçları
KNN
YSA
LVQ



Lineer Regresyon
Karar Ağaçları
KNN
YSA
LVQ



Başka bir model?



Eğitim kümesi üzerindeki performans neyi ifade eder?

- Borsa oynayan maymunlar*
- 1000 kişiye hisse senedi tahminleri göndermek*

- * Hatasız Düşünme Sanatı'ndan

Hata nasıl ölçülür? Çapraz Geçerleme

—Tüm dataseti eşit boyutlu N gruba böl

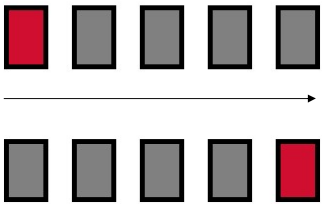
—

—

—Bir grubu test için geriye kalanların hepsini eğitim için kullan

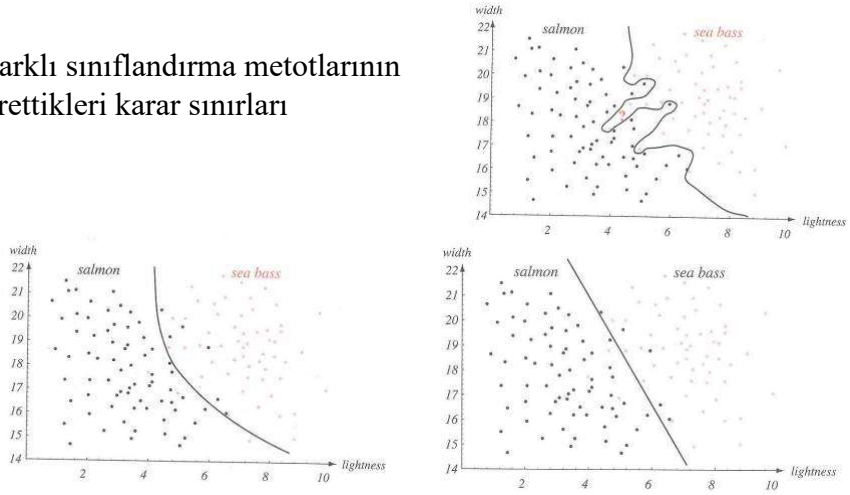
—

—N defa tekrar et



Sınıflandırma Metotları- Sonuç

Farklı sınıflandırma metotlarının ürettikleri karar sınırları



[*] Pattern Classification (2nd Edition) by Richard O. Duda,Peter E. Hart,David G. Stork

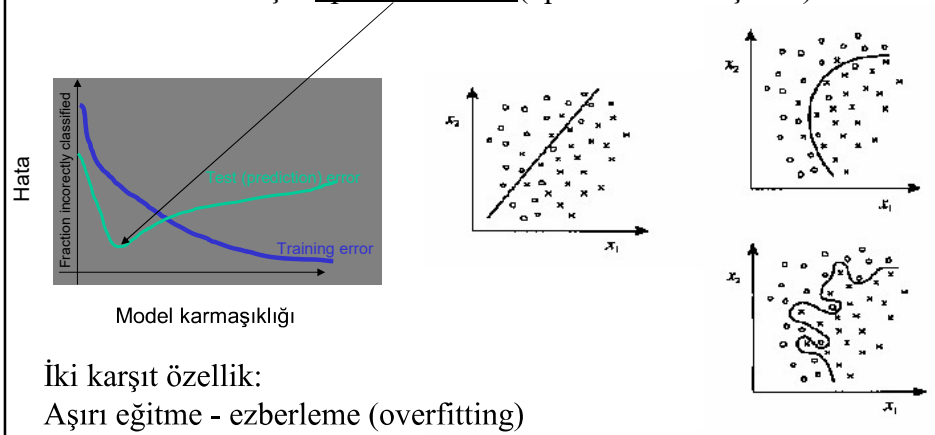
Sınıflandırma Metotları- Sonuç

- Neden bu kadar çok algoritma var?
- Ne zaman hangisini seçeceğiz?

dataset	amlall	ann	bi75ds3	derma	gkanser	Hava
Özellik sayısı	7129	21	470	34	30	34
Sınıf sayısı	2	3	9	6	2	2
Örnek sayısı	72	3772	315	286	456	281
NB	**97,14**	95,55	68,49	77,97	94,29	89,31
SVM	92,86	93,74	62,11	**79,37**	**96,26**	86,48
1NN	94,29	93,4	63,19	76,26	**96,26**	89,72
C45	83,39	**99,58**	65,01	75,2	93,62	91,82
RF	95,71	99,5	**72**	76,96	95,38	**95,02**

Modelin karmaşıklığı arttığında eğitim kümesindeki hata sürekli Düşerken, test kümesindeki hata bir noktadan sonra yükselir.

Her veri kümesi için optimum nokta (optimum karmaşıklık) farklıdır.



İki karşıt özellik:
Aşırı eğitme - ezberleme (overfitting)
Genelleştirebilme

Kümeleme Algoritmaları

- Kümeleme algoritmaları eğiticisiz öğrenme metotlarıdır.
- Örneklere ait sınıf bilgisini kullanmazlar.
- Temelde verileri en iyi temsil edecek vektörleri bulmaya çalışırlar.
- Verileri temsil eden vektörler bulunduktan sonra artık tüm veriler bu yeni vektörlerle kodlanabilirler ve farklı bilgi sayısı azalır.
- Bu nedenle birçok sıkıştırma algoritmasının temelinde kümeleme algoritmaları yer almaktadır.

Kümeleme Algoritmaları

- Bir boyutlu (özellikli) 10 örnek içeren bir veri
  12-15-13-87-4-5-9-67-1-2
- Bu 10 farklı veriyi 3 farklı veriyle temsil etmek istersek:
  12-12-12-77-3-3-3-77-3-3
- şeklinde ifade edebiliriz.
- Kümeleme algoritmaları bu 3 farklı verinin değerlerini bulmakta kullanılırlar.
- Gerçek değerlerle temsil edilen değerler arasındaki farkları minimum yapmaya çalışırlar.

Yukarıdaki örnek için 3 küme oluşmuştur.

- 12-15-13 örnekleri 1. kümede
- 87-67 örnekleri 2. kümede
- 4-5-1-2-9 örnekleri 3. kümede yer almaktadır.

Renk Kümeleme

256 Farklı renkten … 2 renge dönüşüm

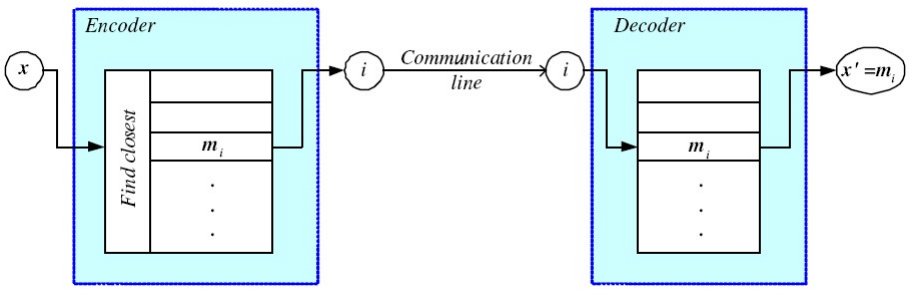
[*] http://fourier.eng.hmc.edu/e161/lectures/digital_image/node2.html

Resim Kümeleme

10*10 luk blokları ifade eden vektörler kümelenmiş

Nasıl Kullanılır?

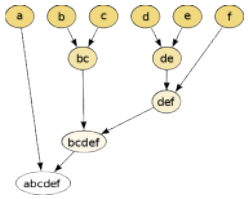
Bulunan (renkleri yada blokları temsil eden) küme merkezlerinden bir kod kitabı oluşturulur. Bu kitap her iki merkeze verilir. Vektörlerin kendileri yerine sadece indisler kullanılır. İndisin maximum büyüklüğü kodlanması için gereken bit sayısını arttırır.
Bu yüzden farklı vektör sayısının az olması istenir.

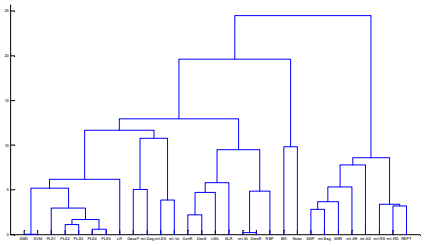


[*] ETHEM ALPAYDIN © The MIT Press, 2004

Hiyerarşik Kümeleme

- Tek kümeden çok kümeye (bölmeli)
- Çok kümeden tek kümeye (eklemeli)





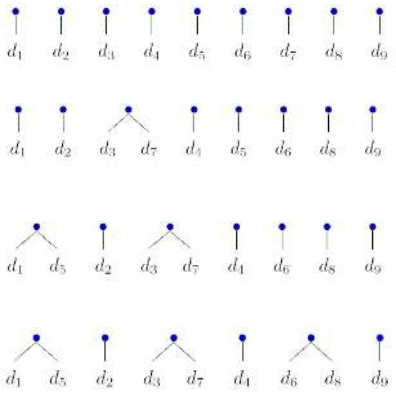
Algoritmaların başarılarına göre kümelenmesi

[*] https://en.wikipedia.org/wiki/Hierarchical_clustering

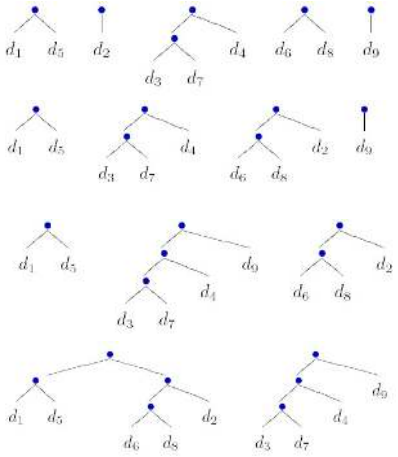
Hiyerarşik Eklemeli Kümeleme

- Birbirine en benzeyen iki kümeyi birleştir
- Tekrar et

Başlangıçta küme sayısı = örnek sayısı



Hiyerarşik Eklemeli kümeleme



Sonuçta küme sayısı=2

Kümelerin birbirine benzerliği

- İki kümenin benzerliği
  - En benzer elemanları (Single link)
  - En benzemeyen elemanları (Compete link)
  - Ortalamaları (Group average)

kullanılarak bulunabilir.

Bölmeli Hiyerarşik Kümeleme

- Tek bir kümeyle başla
- Küme içinde birbirine en az benzeyen iki elemanı bul.
- Kümeyi bu iki elemana yakınlığa göre böl.
- Oluşan her alt küme için bu işlemi tekrar et.

Ne zaman duracağız

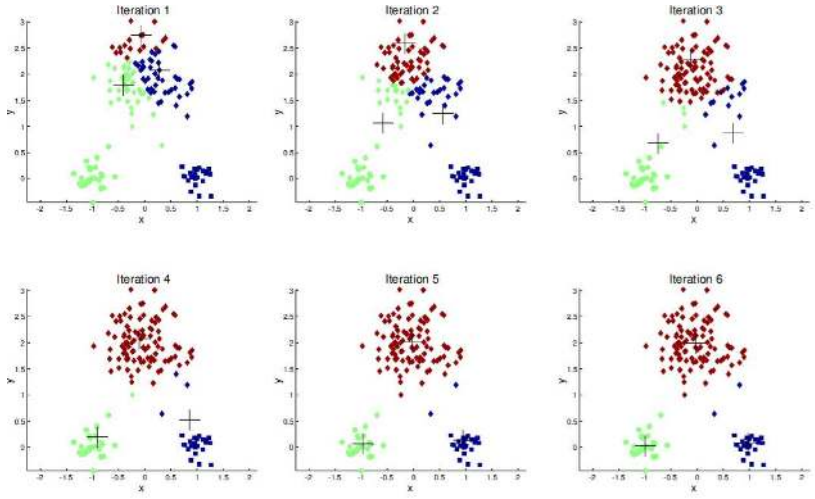
- İstenen küme sayısına ulaşınca
- Önceden belirlenmiş bir toplam benzerlik eşik değerine göre

K-means

Sadece sayısal veriyle çalışır

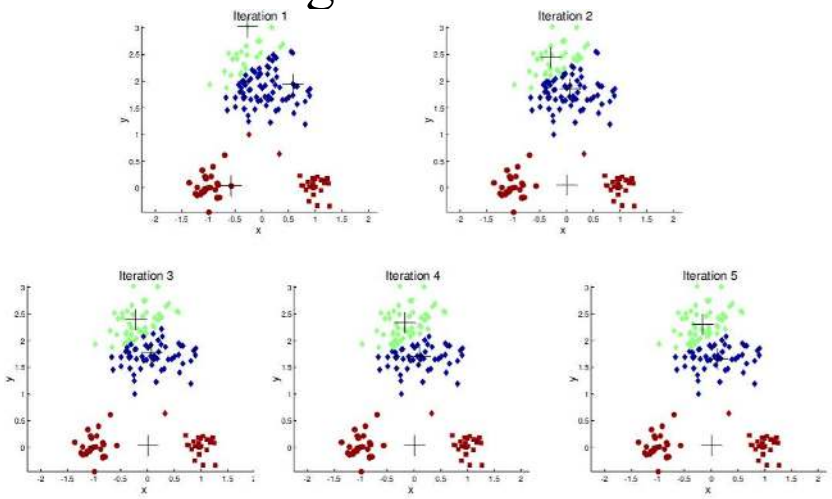
1) Rasgele K adet küme merkezi ata
2) Her örneği en yakınındaki merkezin kümesine ata
3) Merkezleri kendi kümelerinin merkezine ata
4) 2. ve 3. adımları küme değiştiren örnek kalmayıncaya kadar tekrar et.

Adım Adım K-means



[*] https://cs.wmich.edu/alfuqaha/summer14/cs6530/lectures/ClusteringAnalysis.pdf

İlk değerlerin önemi



[*] https://cs.wmich.edu/alfuqaha/summer14/cs6530/lectures/ClusteringAnalysis.pdf

K-means Hata Fonksiyonu

- k adet küme:C1,C2,..Ck
- ci: i. küme merkezi
- x: örnekler
- Farklı ilk değerlerin sonuçları bu hataya göre karşılaştırılır.

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} (x - c_i)^2$$

[*] https://cs.wmich.edu/alfuqaha/summer14/cs6530/lectures/ClusteringAnalysis.pdf

Tekli (Online) vs. Toplu (Batch) K-means

- Önceki örnekte yapılan işlem toplu (batch) k-means
- Eğer her bir örnek için mean'ler güncellenirse → tekli k-means

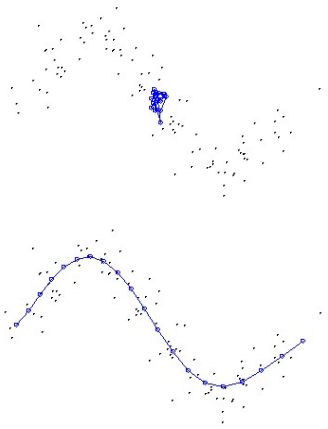
Kendi Kendini Düzenleyen Haritalar Self Organizing Maps*

- Kmeans algoritmasında merkez noktalar arasında herhangi bir ilişki yoktur. SOM'da ise merkez noktalar 1 ya da 2 boyutlu bir dizi içinde yer alırlar. Buna göre birbirlerine 1 ya da 2 boyutlu uzayda komşudurlar.
- Kmeans algoritmasında sadece kazanan (en yakın) merkez güncellenirken SOM'da bütün merkezler kazanan nörona komşuluklarına göre güncellenir. Yakın komşular uzak komşulara göre daha fazla hareket ederler (güncellenirler).
- Merkezlerin birbirlerine bağlı oluşu verinin 1 ya da 2 boyutlu uzaydaki yansımasının da elde edilmesini sağlar.

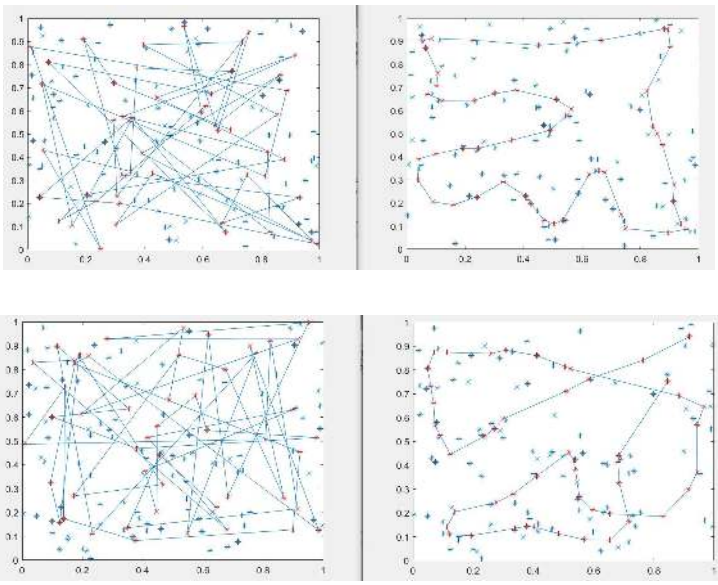
[*] Kohonen, Teuvo. "The self-organizing map." *Proceedings of the IEEE* 78.9 (1990): 1464-1480.

SOM

- Örnekte SOM merkezleri 1 boyutlu bir dizide birbirlerine komşudurlar. Başlangıçtaki durumları rasgele atandığı için bir yumak şeklindedirler. Eğitim tamamlandığında ise SOM merkezleri verinin şeklini almıştır.

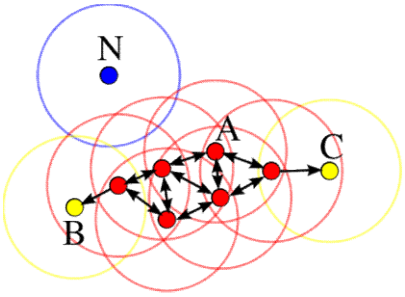


Kod: my_som.m



DBscan

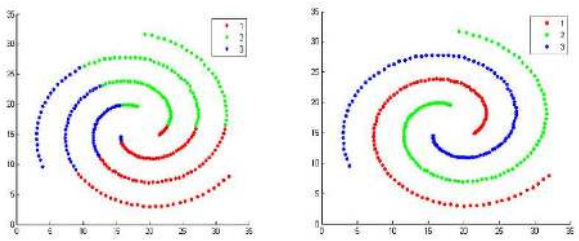
- Yoğunluk tabanlı kümeleme
- Hiper-parametreleri: eps, min_p
- Önce tüm noktaları çekirdek, sınır, gürültü olarak belirle
- Çekirdek (kırmızı): eps komşuluğunda en az min_p adet nokta bulunan noktalar
- Gürültü (mavi): eps komşuluğunda hiç nokta bulunmayan noktalar
- Sınır (sarı): eps komşuluğunda min_p'den az nokta bulunan noktalar



[*] https://en.wikipedia.org/wiki/DBSCAN

DBscan

- Noktalar etiketlendikten sonra gürültüler elenir
- Çekirdeklerden birbirine eps mesafede olanlar birleştirilir.
- X (hiper parametrelere bağlı) adet küme oluşur.
- Sınır noktalar en yakın çekirdeğin kümesine atanır



[*] https://www.researchgate.net/figure/Results-of-a-k-means-b-DBSCAN-on-Jain-dataset_fig2_291831757

Kümeleme algoritmalarında

- Ne zaman dururuz?
- Ölü nöron kavramı (SOM, K-means)
- Başarı ölçütü nedir?
- **Küme sayısı nasıl belirlenir?**
  - Mecburiyet (verilmiştir)
  - ?

Regresyon Algoritmaları

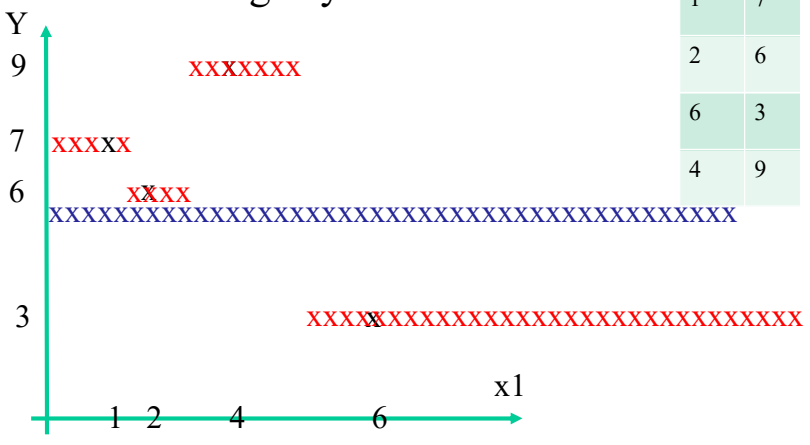
- Basit lineer regresyon
- kNN
- Regresyon ağaçları
- YSA

Basit lineer regresyon

X: girişler (n*d)
Y: çıkışlar (n*1)
Model: Y=XW
W: katsayılar
Amaç W yi bulmak
Her 2 tarafı X^{-1} ile çarpalım.
$X^{-1}Y=X^{-1}XW$ $(X^{-1}X=I)$
$W=X^{-1}Y$ (tamam, ama X kare matris değilse?)
$X^TY=X^TXW$ (X^TX her zaman kare matristir)
$(X^TX)^{-1}(X^TY)=(X^TX)^{-1}(X^TX)W$ $[(X^TX)^{-1}(X^TX)=I]$
$W=(X^TX)^{-1}(X^TY)$

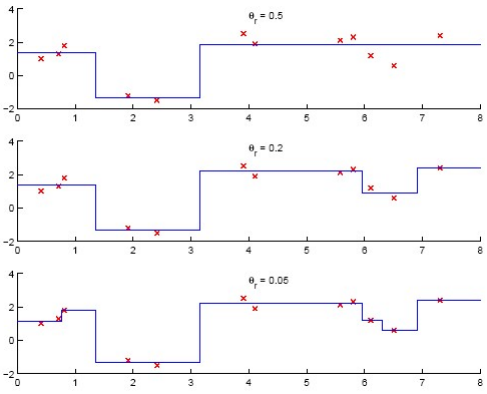
KNN ile regresyon

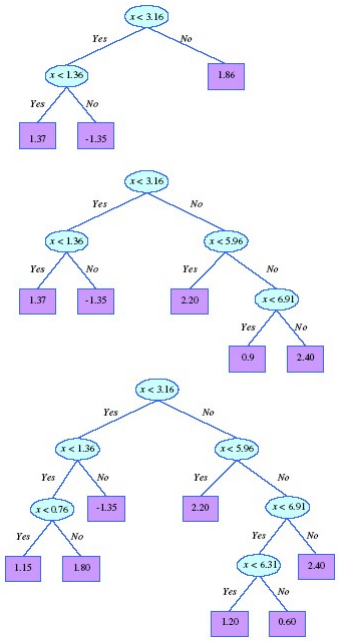
x1	Y
1	7
2	6
6	3
4	9



K=1
K=4

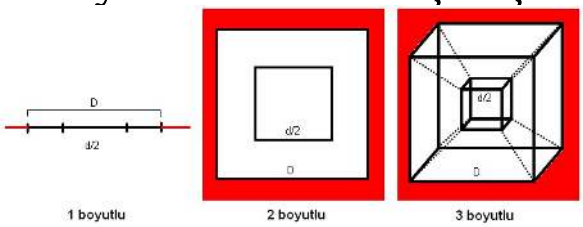
Regresyon ağaçlarında
kabul edilen hata değerinin etkisi





ETHEM ALPAYDIN © The MIT Press, 2004

Çok Boyutlu Verilerle Çalışmak-1



Boyut Sayısı	Merkeze daha yakın noktaların oranı (%)
1	50
2	25
3	12,50
…	…
P	$(½)^P$

Boyut sayısı arttığında verilerin çok büyük bir kısmı sınıfları ayıran sınırlara çok yakın yerlerde bulunacağından sınıflandırma yapmak zorlaşmaktadır.

Çok Boyutlu Verilerle Çalışmak-2

- Tek boyutlu uzayda [0,1] aralığı temsil eden 10 nokta
- Rastgele bir noktanın, uzayı temsil eden noktalardan en yakın olanına ortalama uzaklığı = 0.5
- İki boyutlu uzayda rasgele bir noktanın en yakın noktaya olan ortalama uzaklığının düşey ya da dikey (manhattan) 0.5 olması için gerekli temsilci nokta sayısı = 100

Boyut Sayısı	Gerekli temsil eden nokta sayısı
1	10
2	100
3	1000
…	…
p	10^p

Doğru sınıflandırma yapmak için gereken örnek sayısı artıyor.

Veri Sızıntısı (Data Leakage)

- Tahmin sonuçlarının çok iyi görünmesine sebep olabilir, ama gerçekler uygulamada ortaya çıkar
- Test kümesindeki bilgilerin eğitim sürecine karışması
  - Verileri normalize ederken, özellik seçimi yaparken test kümesini de kullanmak
- Test zamanı elde olmayacak özelliklerin kullanılması
  - x(t)=f(x(t-1),x(t+1))
- Zaman serisi sınıflandırmada eğitim ve test kümelerini oluşturmada hata
  - Rasgele seçim yapılmamalı, bir t anından öncesi eğitim, sonrası test olmalı ki ardışık süreçleri değil sınıfı tanısın.
- Verilerde çıkışla korelasyonu çok yüksek olan özelliklerin olması (kişi tanırken id, telefon no vb.)

Sonuç olarak

- Makineler insanlığın işgücüne sağladıkları katkıyı, makine öğrenmesi metotları sayesinde insanlığın beyin gücüne de sağlamaya başlamışlardır.

Bir gün bilgisayarlar bütün bunları mükemmel bir şekilde yaparlarsa Nasıl bir dünya

- Bir sürü işsiz bilgisayar mühendisi ☺
- Bir sürü işsiz insan
- ???

Kaynaklar

- Alpaydın E. (2004) “Introduction to Machine Learning”, The MIT Press, 3-6
- https://www.autonlab.org/resources/tutorials/information-gain
- http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf
- http://www.kernel-machines.org
- T.Kohonen," Self-Organization and associative Memory",3d ed, 1989, Berlin :Springer-Verlag.
- http://www.willamette.edu/~gorr/classes/cs449/Classification/perceptron.html
- O. T. Yıldız, E. Alpaydın, Univariate and Multivariate Decision Trees, Tainn 2000
- http://mathworld.wolfram.com/K-MeansClusteringAlgorithm.html
- Pattern Classification (2nd Edition) by Richard O. Duda,Peter E. Hart,David G. Stork
- Machine Learning, Tom Mitchell, McGraw Hill, 1997

Weka

Copyright: Martin Kramer (mkramer@wxs.nl)