

Probabilistic Classification: Naïve Bayes

Denetimli Öğrenme Teknikleri

- Uzaklık Temelli (K-Nearest-Neighbor) yöntemler
- Karar Ağacı(Decision Tree) yöntemleri
- **Olasılık temelli yöntemler (Naïve Bayes)**
- **Support Vector Machines**
- Yapay Sinir Ağları (Artificial Neural Networks)
 - ANN, CNN, RNN
- Ensemble Yöntemler (Bagging)
 - Random Forest
- Ensemble Yöntemler (Boosting)
 - AdaBoost, Gradient Boosting, XGBoost

Bayes Yöntemi ile Sınıflandırma

Bayes Sınıflandırma

- Sınıflandırma Problemleri için olasılık temelli bir yöntem

- Koşullu Olasılık
(Conditional Probability):

$$P(C | A) = \frac{P(A, C)}{P(A)}$$

$$P(A | C) = \frac{P(A, C)}{P(C)}$$

- Bayes teoremi:

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

Bayes Teoremi Örnekleri

- Verilen:
 - Bir doktor menanjitin 50% oranında boyun tutulmasına neden olduğunu biliyor
 - Her hangi bir hastanın geçmiş verilere göre menanjit olma ihtimali 1/50,000 oranındadır
 - Her hangi bir hastanın geçmiş verilere göre boyun tutulması oranı 1/20 diye biliniyor
- Eğer bir hastanın boyun tutulması şikayeti varsa Menanjit olma ihtimali nedir?

$$P(M | S) = \frac{P(S | M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Sınıflandırıcı

- Her attribute ve Sınıf random değişken olarak kabul edilir
- (A_1, A_2, \dots, A_n) attributeleri olan bir kayıt verilsin
 - Amaç bu kayıta ait olduğu sınıf C 'yi bulmaktır
 - Spesifik olarak, $P(C | A_1, A_2, \dots, A_n)$ olasılığını maksimize edecek C sınıfını bulma görevidir
- $P(C | A_1, A_2, \dots, A_n)$ olasılık değerini verinin kendisinden bulabilir miyiz?

Naïve Bayes Sınıflandırıcı

- A_i attributleri arasında bağımsızlık olduğunu varsayar:
 - $P(A_1, A_2, \dots, A_n | C) = P(A_1 | C_j) P(A_2 | C_j) \dots P(A_n | C_j)$
 - Böylece her bir A_i ve C_j değeri için $P(A_i | C_j)$ olasılığı bulunur.
 - $P(C_j)$, $P(A_i | C_j)$ olasılığını maksimize edecek şekilde her yeni nokta C_j sınıfına tahsis edilir

Olasılık Değerleri Veriden Nasıl Elde edilir?

<i>Tid</i>	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Sınıf: $P(C) = N_c / N$
 - e.g., $P(\text{No}) = 7/10$,
 $P(\text{Yes}) = 3/10$
- Sayılabilir değişkenler için:
 $P(A_i | C_k) = |A_{ik}| / N_{ck}$
 - $|A_{ik}|$ C_k sınıfına ait olup A_i özelliği içeren değerlerin sayısı
 - Örnekler
 - $P(\text{Status}=\text{Married} | \text{No}) = 4/7$
 - $P(\text{Refund}=\text{Yes} | \text{Yes})=0$

Olasılık Değerleri Veriden Nasıl Elde edilir?

- Sürekli Değerler İçin:
 - **Belli aralıklardan** oluşan ayrıklaştırma(Discretization) yapılır
 - bir ordinal attribute için bir aralık
 - bağımsızlık varsayımını engeller
 - **İkiye bölme:** $(A < v)$ or $(A > v)$
 - yeni bir attribute olarak ikiye bölecek bir değer seçme
 - **Olasılık yoğunluk tahmini**(Probability dense estimation):
 - normal dağılım(Bell Curve) olduğunu varsayıp
 - veriyi kullanıp dağılım parametrelerini belirle (ortalama ve standart sapma gibi)
 - Olasılık dağılımı belirlendikten sonra conditional probability $P(A_i | c)$ hesap edilebilir

Olasılık Değerleri Veriden Nasıl Elde edilir?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal Dağılım:

$$P(A_i | c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

– Her bir(A_i, c_i) değeri için

- (Income, Class=No) için:

– Eğer Class=No

- örnek ortalaması = 110
- örnek variance değeri= 2975

$$P(\text{Income} = 120 | \text{No}) = \frac{1}{\sqrt{2\pi (54.54)}} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Naïve Bayes Sınıflandırıcı Örnek

Verilmiş bir

Test Kayıdı için;

$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$

naive Bayes Classifier:

$$P(\text{Refund}=\text{Yes}|\text{No}) = 3/7$$

$$P(\text{Refund}=\text{No}|\text{No}) = 4/7$$

$$P(\text{Refund}=\text{Yes}|\text{Yes}) = 0$$

$$P(\text{Refund}=\text{No}|\text{Yes}) = 1$$

$$P(\text{Marital Status}=\text{Single}|\text{No}) = 2/7$$

$$P(\text{Marital Status}=\text{Divorced}|\text{No}) = 1/7$$

$$P(\text{Marital Status}=\text{Married}|\text{No}) = 4/7$$

$$P(\text{Marital Status}=\text{Single}|\text{Yes}) = 2/7$$

$$P(\text{Marital Status}=\text{Divorced}|\text{Yes}) = 1/7$$

$$P(\text{Marital Status}=\text{Married}|\text{Yes}) = 0$$

For taxable income:

If class=No: sample mean=110

sample variance=2975

If class=Yes: sample mean=90

sample variance=25

$$\begin{aligned} P(X|\text{Class}=\text{No}) &= P(\text{Refund}=\text{No}|\text{Class}=\text{No}) \\ &\quad * P(\text{Married}|\text{Class}=\text{No}) \\ &\quad * P(\text{Income}=120\text{K}|\text{Class}=\text{No}) \\ &= 4/7 * 4/7 * 0.0072 = 0.0024 \end{aligned}$$

$$\begin{aligned} P(X|\text{Class}=\text{Yes}) &= P(\text{Refund}=\text{No}|\text{Class}=\text{Yes}) \\ &\quad * P(\text{Married}|\text{Class}=\text{Yes}) \\ &\quad * P(\text{Income}=120\text{K}|\text{Class}=\text{Yes}) \\ &= 1 * 0 * 1.2 * 10^{-9} = 0 \end{aligned}$$

$P(X|\text{No})P(\text{No}) > P(X|\text{Yes})P(\text{Yes})$ olduğundan

$P(\text{No}|X) > P(\text{Yes}|X)$ olur

$\Rightarrow \text{Class} = \text{No}$

Naïve Bayes Sınıflandırma

- Eğer conditional probability değerlerinden biri sıfır olursa tüm eşitsizlik sonucu sıfır olur

- Olasılık Tahmini:

$$\text{Original : } P(A_i | C) = \frac{N_{ic}}{N_c}$$

c: sınıfların sayısı

$$\text{Laplace : } P(A_i | C) = \frac{N_{ic} + 1}{N_c + c}$$

p: önceki olasılık değeri

m: parametre

$$\text{m - estimate : } P(A_i | C) = \frac{N_{ic} + mp}{N_c + m}$$

Naïve Bayes Sınıflandırma Örneği

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

A: özellikler

M: memeli hayvan

N: memeli-olmayan hayvan

$$P(A | M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A | N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A | M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A | N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

$$P(A|M)P(M) > P(A|N)P(N)$$

=> Memeli hayvan

Naïve Bayes (Özet)

- İzole noise(ayrık) değerlere karşı dirençlidir.
- Kayıp değerleri de olasılık hesabında görmezden gelerek ele alır
- Alakasız attributlere karşı dirençlidir
- Naive Bayes'in hesaplama maliyeti düşüktür.
- Büyük bir veri kümesi üzerinde verimli bir şekilde çalışabilir.
- Ayrıca, metin analizi sorunları durumunda da iyi performans gösterir.
- Bazı attributler için bağımsızlık varsayımı doğru olmayabilir

Support Vector Machines(SVM)

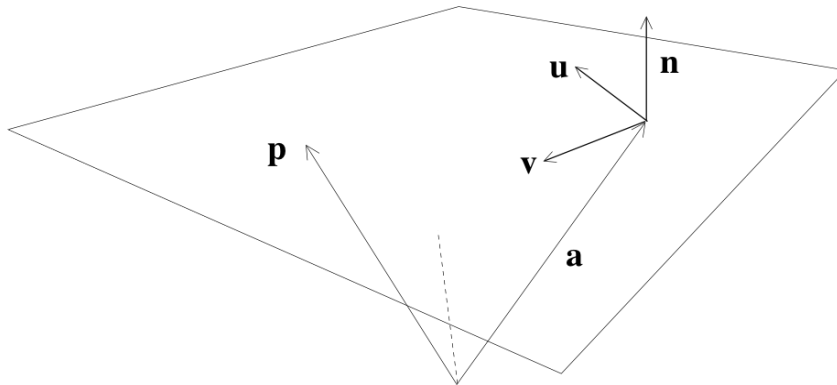
Doğrusal Ayırıcılar

Bir düzlem noktalar kümesi olarak ifade edilebilir:

$$\mathbf{p} = \mathbf{a} + s\mathbf{u} + t\mathbf{v}, \quad (s, t) \in \mathcal{R}.$$

vektör
orijin ile düzlem arasındaki her hangi bir
nokta

Düzlemdeki paralel olmayan iki doğru



Alternatif gösterimi şöyle:

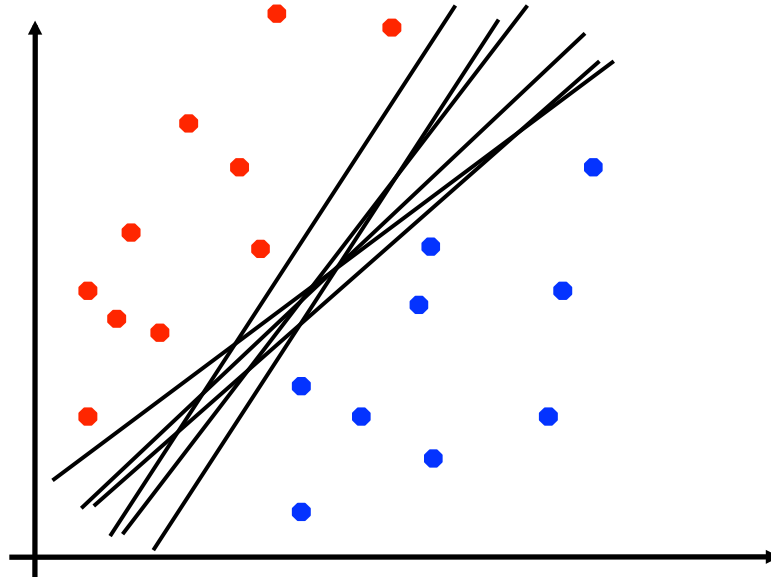
$$(\mathbf{p} - \mathbf{a}) \cdot \mathbf{n} = 0 \Leftrightarrow \mathbf{p} \cdot \mathbf{n} = \mathbf{a} \cdot \mathbf{n}$$

Normal vektör

Sadece bu vektör çarpımını bulmamız yeterli

Doğrusal Ayırıcılar

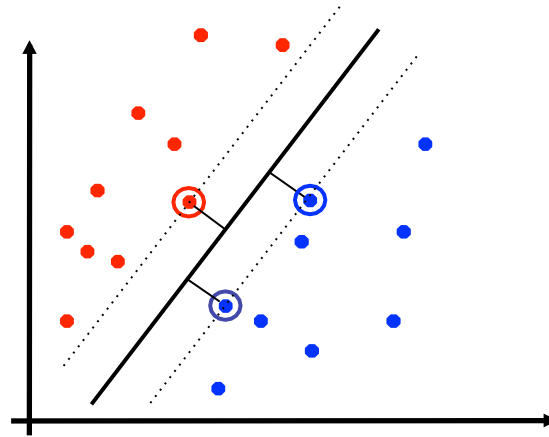
- Training verileri doğrusal olarak ayrılabilirse, perceptronun belirli doğrusal ayırıcıları bulması garanti edilir
- Peki hangisi en **optimum**?



Support Vector Machine (SVM)

- SVMs (Vapnik, 1990'lar) **en geniş** doğrusal ayrımı sağlayacak doğrunun belirlenmesi

Outlier
değerlere
dayanıklı

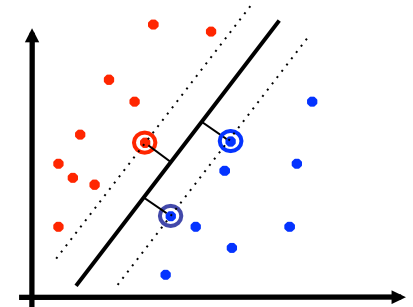


V. Vapnik

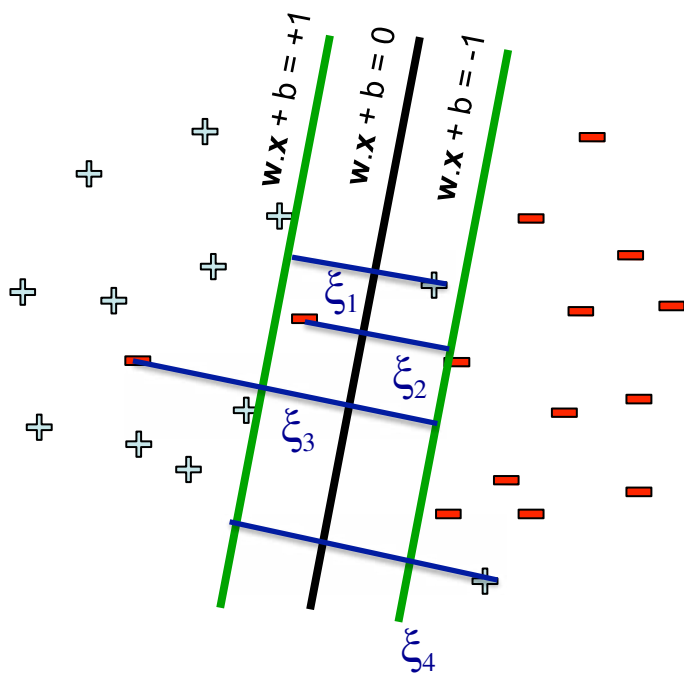
- Teori ve pratikte iyi çalışan bir yöntem
- SVM görüntü işleme verilerindeki başarısı nedeniyle ünlendi. El yazısı tanıma işlemlerinde neural-network ile karşılaştırılabilir bir düzeyde başarılı sonuçlar veriyor.

Support vector machines: 3 önemli fikir

1. **Optimizasyon** kullanarak az hata veren çözüm bul (örneğin bir hyperplane)
2. En **geniş margin** ayracı belirleyerek genelleme yapmayı imkanını geliştir
3. **Kernel** yöntemi kullanarak çok sayıda özellik içerecek şekilde vektörel alanı genişlet



Anahtar Fikir #1: Esnekliğe izin ver “Soft margin” SVM



$$\text{minimize}_{w, b, \xi} \quad \sum_j \xi_j$$
$$(w \cdot x_j + b) y_j \geq 1 - \xi_j, \forall j \quad \xi_j \geq 0$$

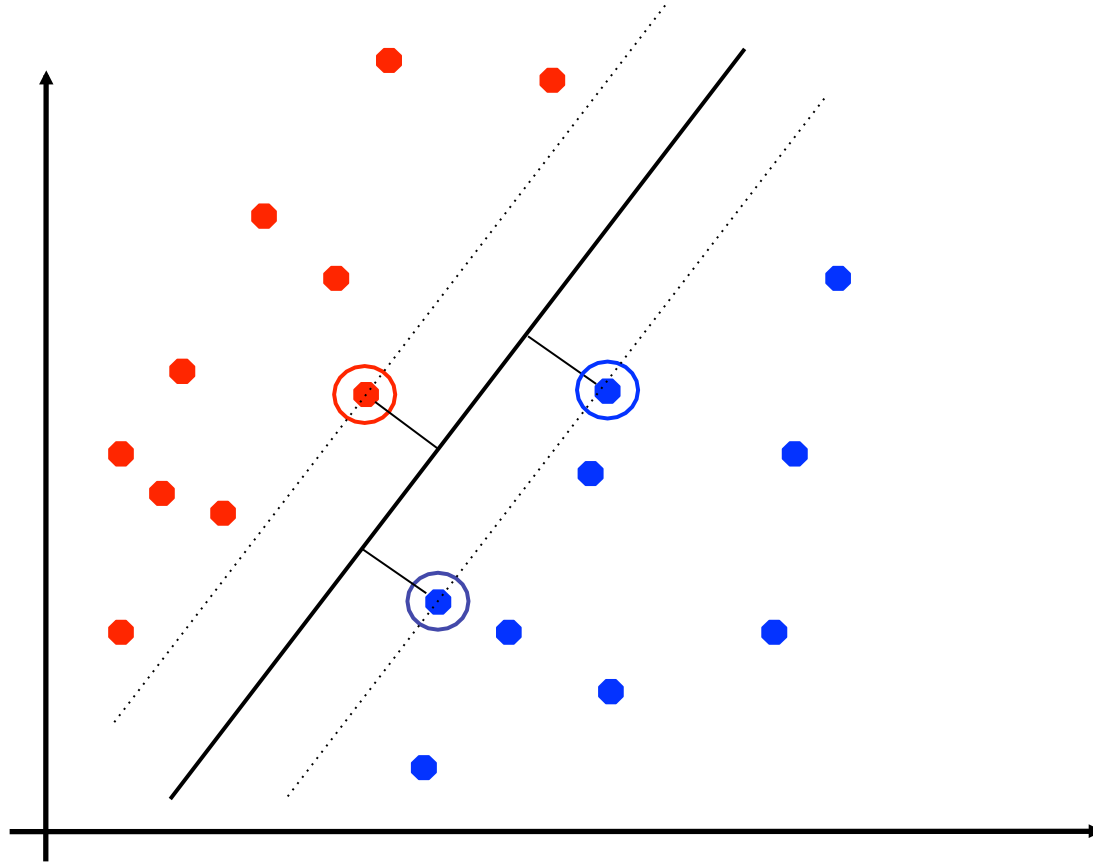
↑
“slack variables”

Yeniden bir doğrusal programlama problemi var ve en verimli çözüm bulunabilir

Her veri noktası için:

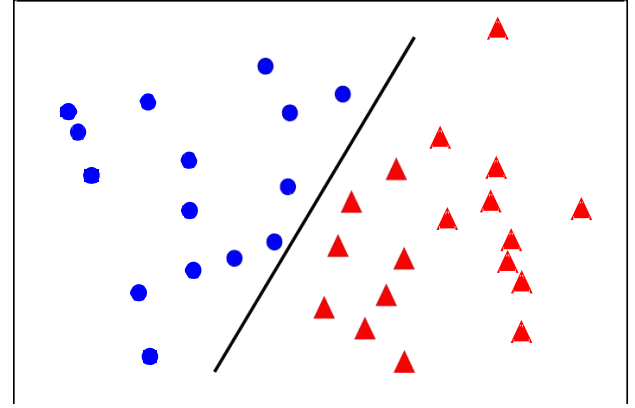
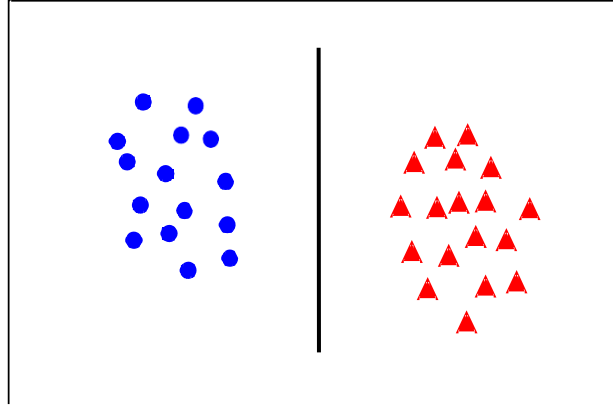
- eğer margin değeri ≥ 1 , önemli değil
- eğer margin değeri < 1 , doğrusal ceza uygula

Anahtar Fikir #2: geniş margin belirle

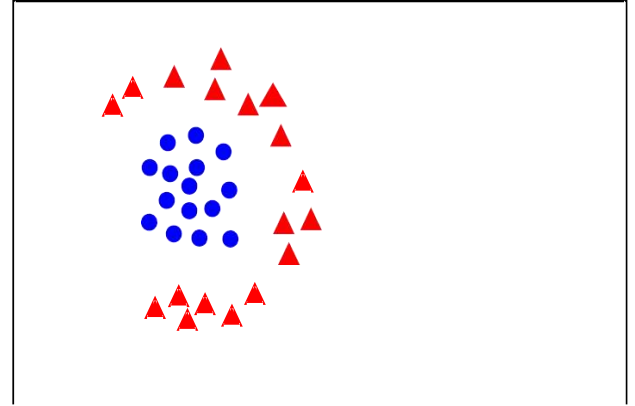
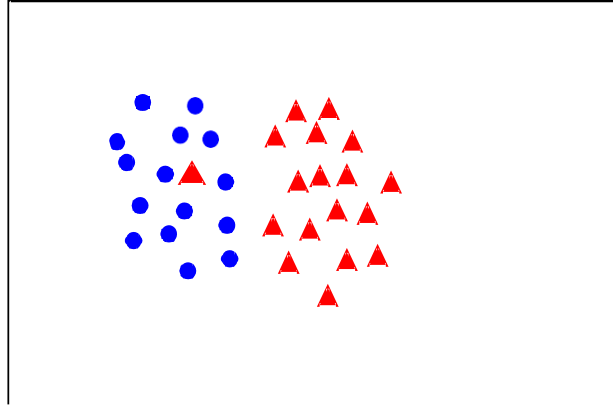


Doğrusal Ayrılabilirlik

Doğrusal
ayrılabilir



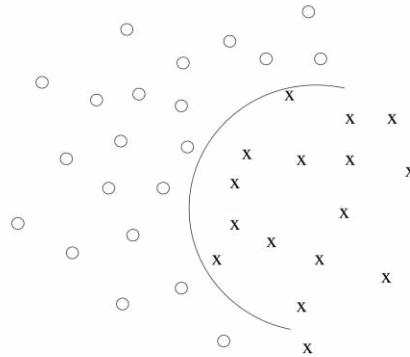
Doğrusal
olarak
ayrılmaz



Anahtar Fikir #3: Kernel Kullanımı

Veri doğrusal olarak ayrılmazsa kernel numarası kullanılıp çözüm bulunabilir

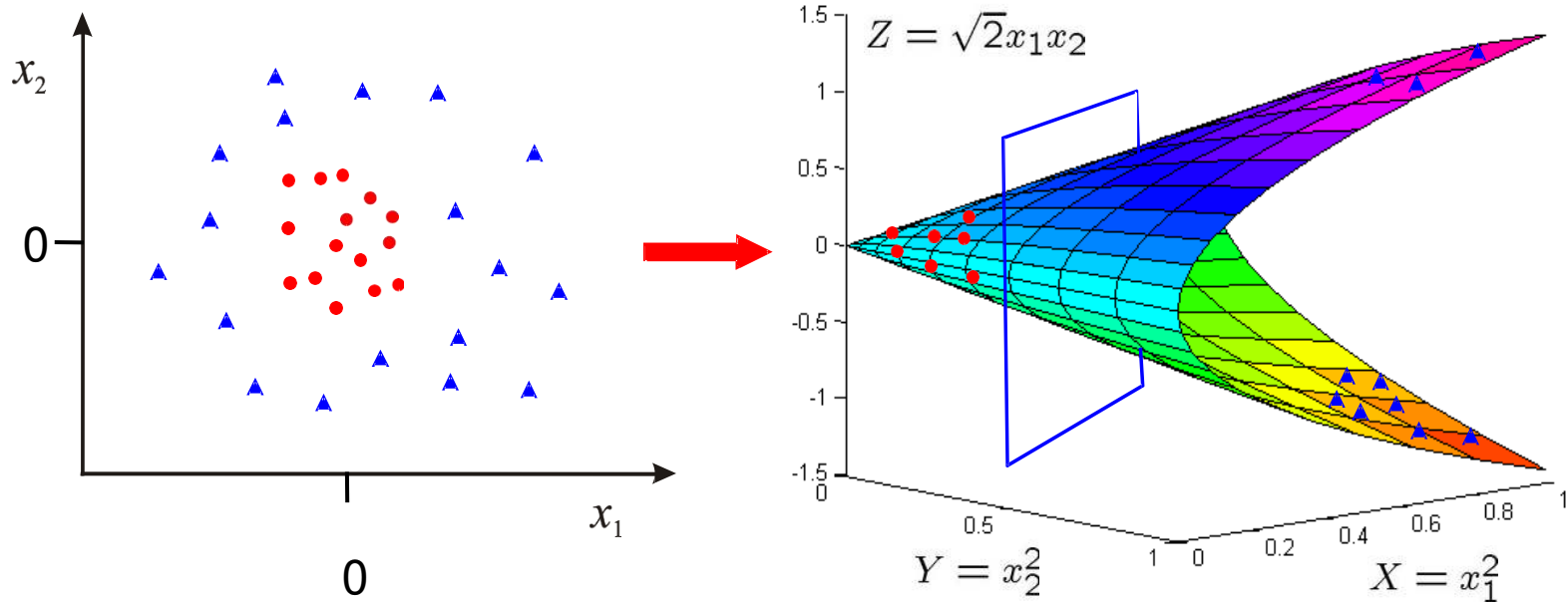
Quadratic kernel



Non-linear separator in the **original x-space**

Kernel ile veriyi daha yüksek boyuta taşımak

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \quad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$



- Veri 3D boyutta doğrusal olarak bölünebilir
- Böylece problem hala doğrusal sınıflandırma ile çözülebilir

Sıklıkla kullanılan kernel'lar

- d boyutlu polinomlar

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v})^d$$

- d'ye kadar boyutlu polinomlar

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^d$$

- Gaussian kernel

$$K(\vec{u}, \vec{v}) = \exp\left(-\frac{\|\vec{u} - \vec{v}\|_2^2}{2\sigma^2}\right) \longleftarrow \text{Öklid uzaklığı}$$

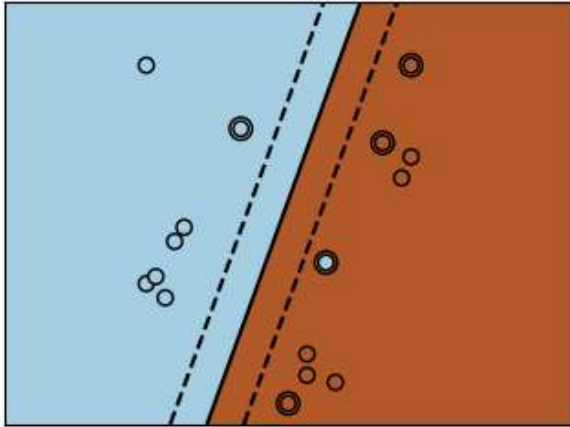
- Sigmoid

$$K(\mathbf{u}, \mathbf{v}) = \tanh(\eta \mathbf{u} \cdot \mathbf{v} + \nu)$$

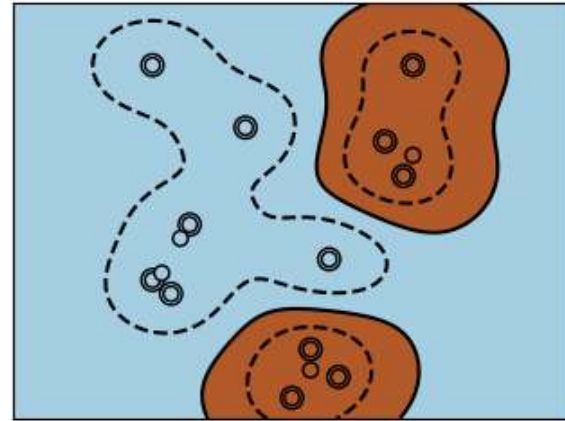
- Radial basis ve daha bir çok kernel çeşidi mevcut. Aktif araştırma alanı.

Kernel

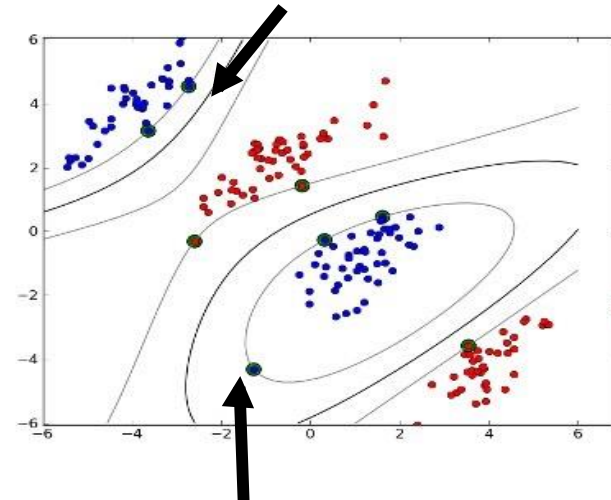
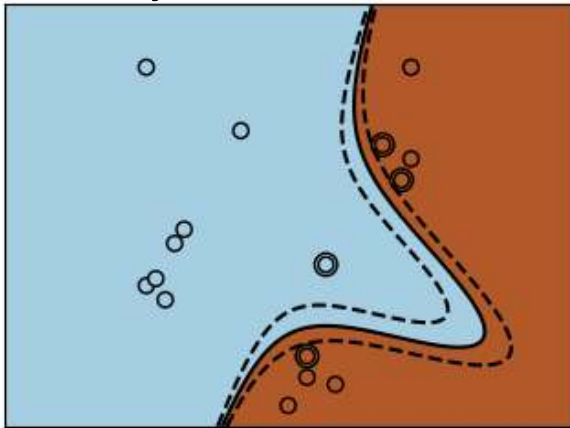
Linear Kernel



RBF Kernel



Polynomial Kernel

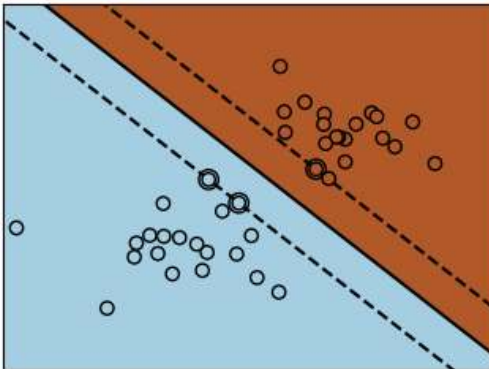


Support vectors

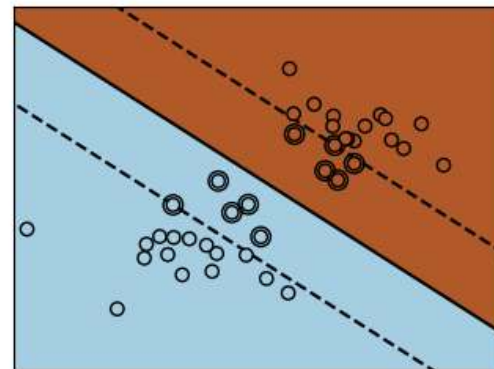
SVM Regularization

- ***C(Cost)***, yanlış sınıflandırmayı temsil eden ceza parametresidir. SVM optimizasyonuna ne kadar hatanın katlanılabilir olduğunu söyler.
- ***Büyük C değeri***, temel olarak modelimize, verilerimizin dağılımına çok fazla güvenimiz olmadığını ve yalnızca ayırım çizgisine yakın noktaları dikkate alacağımızı söyler.
- ***Küçük C değeri***, gözlemlerin çoğunu / tümünü içerir ve bu, bölgedeki tüm veriler kullanılarak kenar boşluklarının hesaplanmasına izin verir.

Unregularized SVM



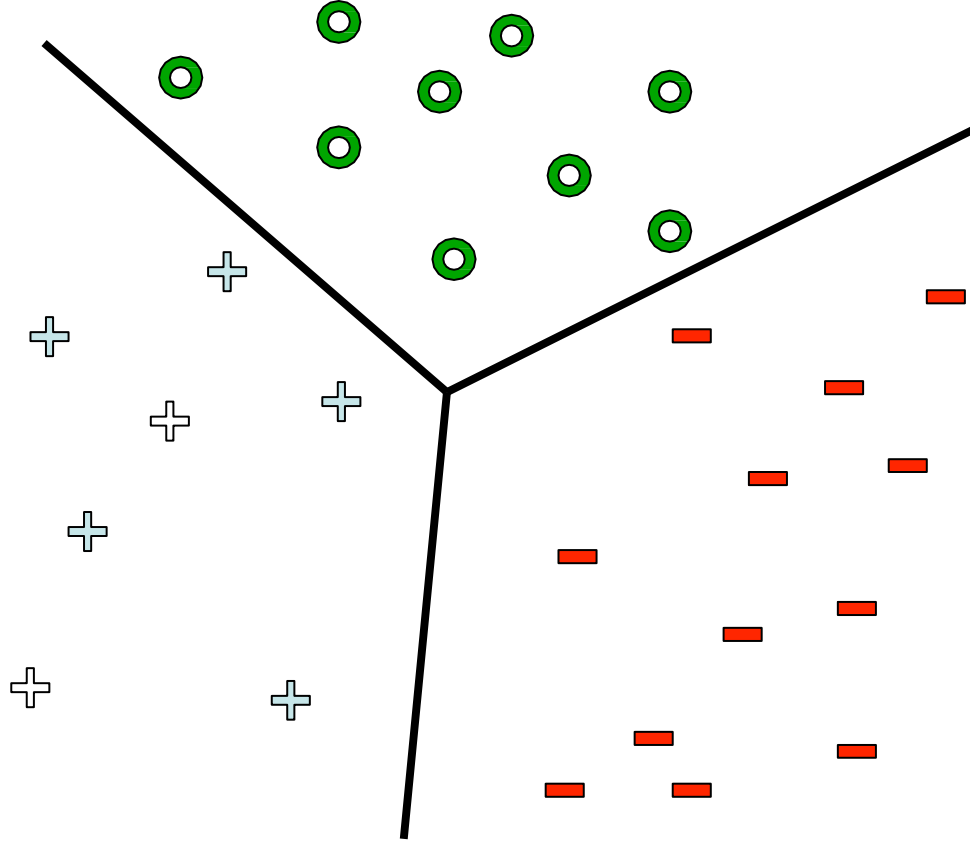
Regularized SVM (default)



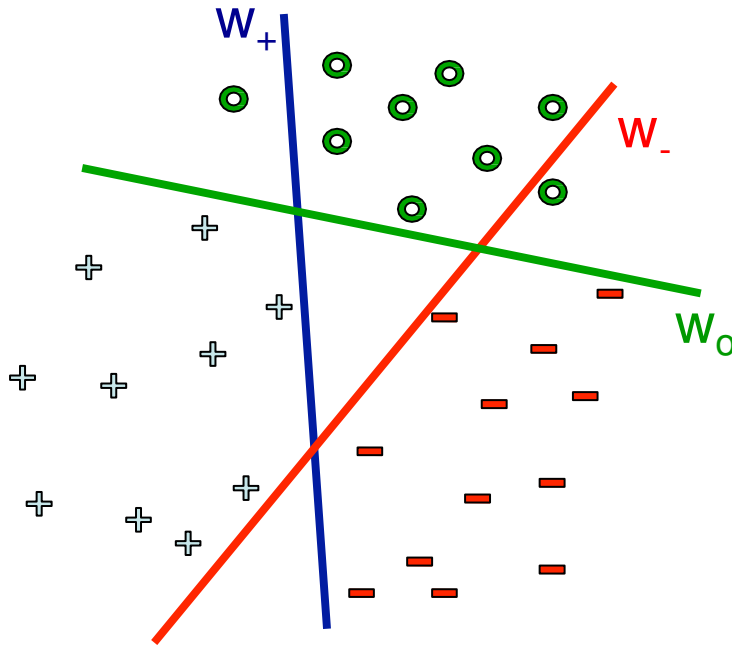
RBF Kernel Hiperparametre

- ***RBF Kernel'in Gamma*** parametresi, model tarafından destek vektörleri olarak seçilen örneklerin etki yarıçapının tersi olarak görülebilir.
- Modelin davranışı gama parametresine çok duyarlıdır.
- ***Gama çok büyükse***, destek vektörlerinin etki alanının yarıçapı sadece destek vektörünün kendisini içerir ve C ile hiçbir düzenleme miktarı overfittingi önleyemez.
- ***Gama çok küçükse***, model çok sınırlıdır ve verilerin karmaşıklığını veya "şeklini" yakalayamaz. Seçilen herhangi bir destek vektörünün etki bölgesi, tüm eğitim setini içerecektir.

Çok-sınıfllı sınıflandırma nasıl yapılır?



One vs all sınıflandırma



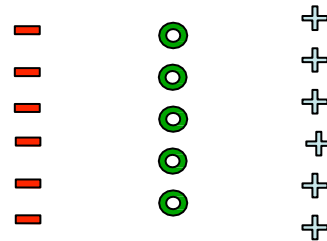
3 sınıflandırıcı eğitimi:

- - vs {o,+}, weights w_-
- + vs {o,-}, weights w_+
- o vs {+,-}, weights w_o

Kategori tahmini:

$$\hat{y} \leftarrow \arg \max_k w_k \cdot x + b_k$$

Potansiyel
problem
nedir?



SVM (Özet)

- SVM, Naïve Bayes algoritmasına göre iyi doğruluk sunar ve daha hızlı tahmin gerçekleştirir.
- Ayrıca karar aşamasında bir alt eğitim noktası kullandıkları için daha az bellek kullanırlar.
- SVM, açık bir ayırma marjı ve yüksek boyutsal boşlukla iyi çalışır.
- SVM, yüksek eğitim süresi nedeniyle büyük veri kümeleri için pek uygun olmayabilir ve Naïve Bayes'e kıyasla eğitimde daha uzun sürer.
- Çakışan sınıflarla kötü çalışır ve ayrıca kullanılan çekirdek türüne duyarlıdır.