# Destek Vektör Makineleri

Dr. Caner Erden cerden@sakarya.edu.tr

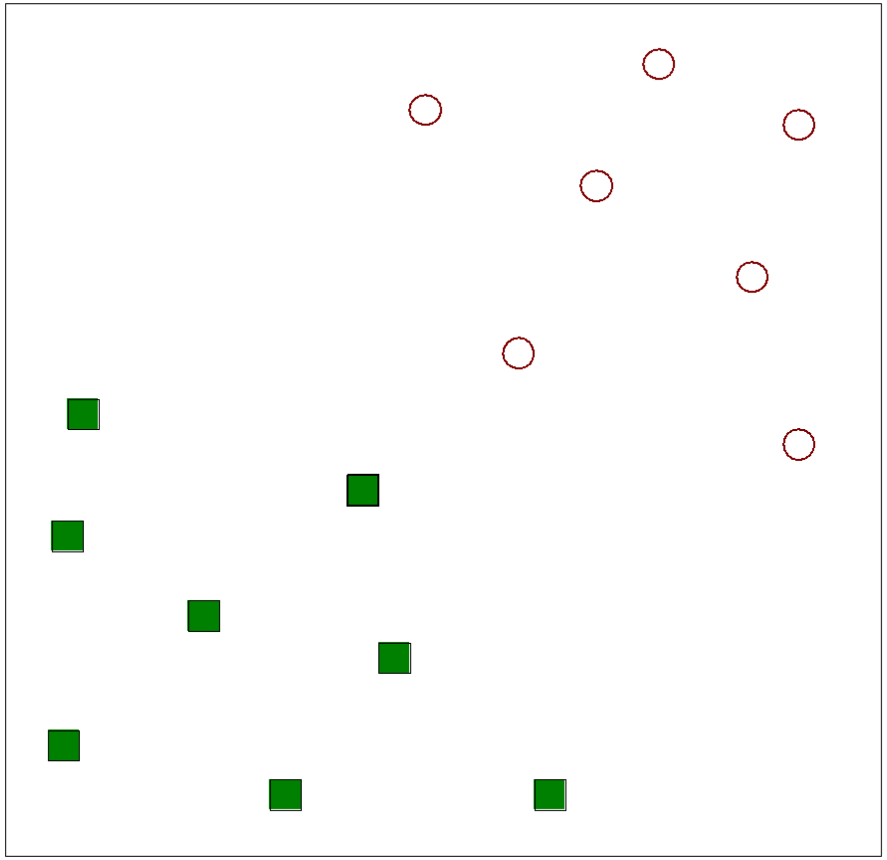
Sakarya Üniversitesi, Endüstri Mühendisliği Bölümü

# Destek Vektör Makineleri(DVM)

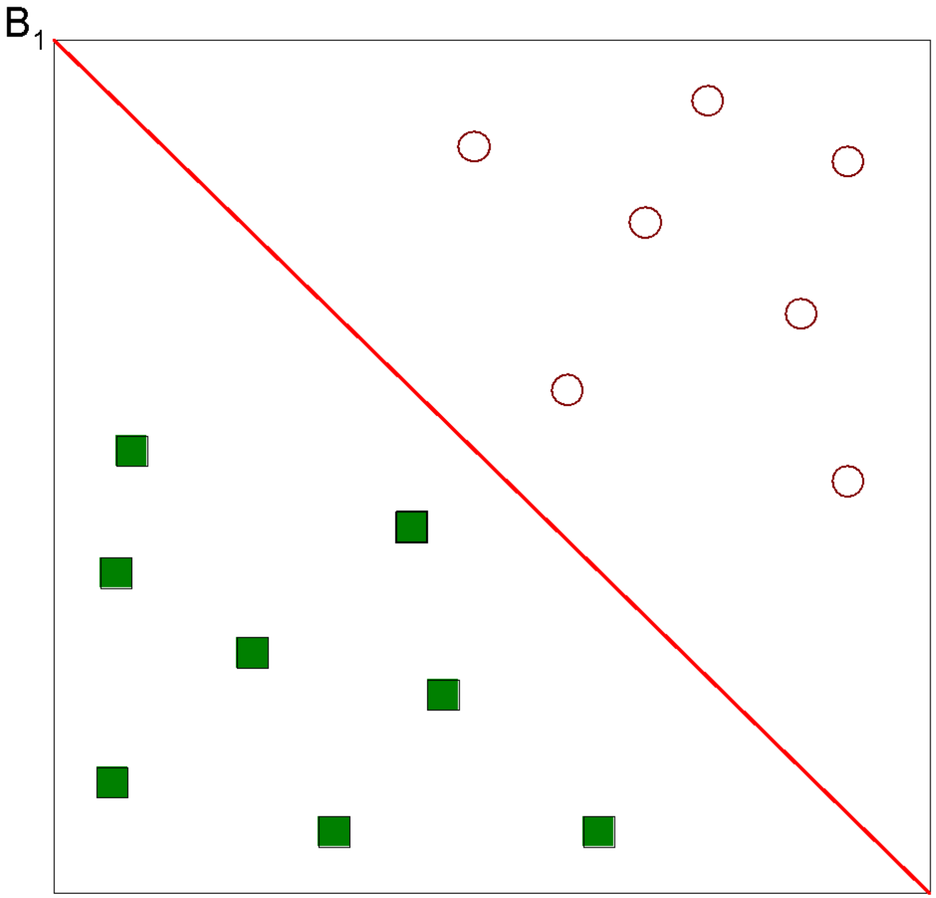
- Bu yöntem sınıflandırmayı bir doğrusal ya da doğrusal olmayan bir fonksiyon yardımıyla yerine getirir.

- Veriyi birbirinden ayırmak için en uygun fonksiyonun tahmin edilmesi esasına dayanır.

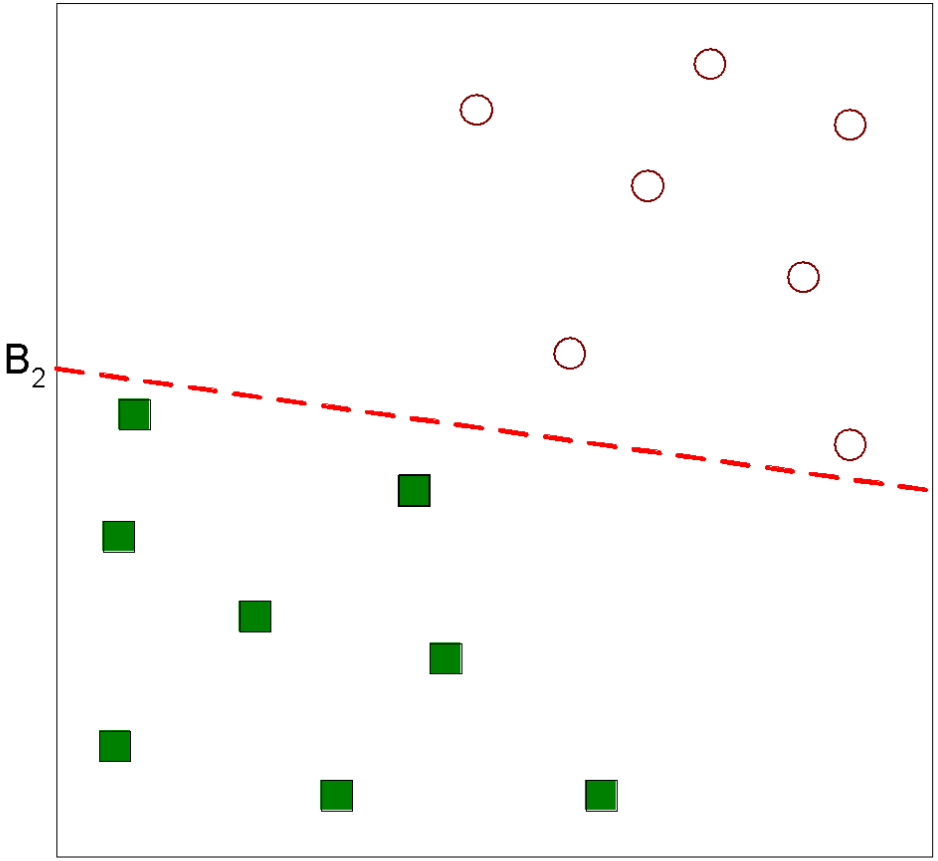
- Verileri birbirinden ayıran bir çizgi bulalım



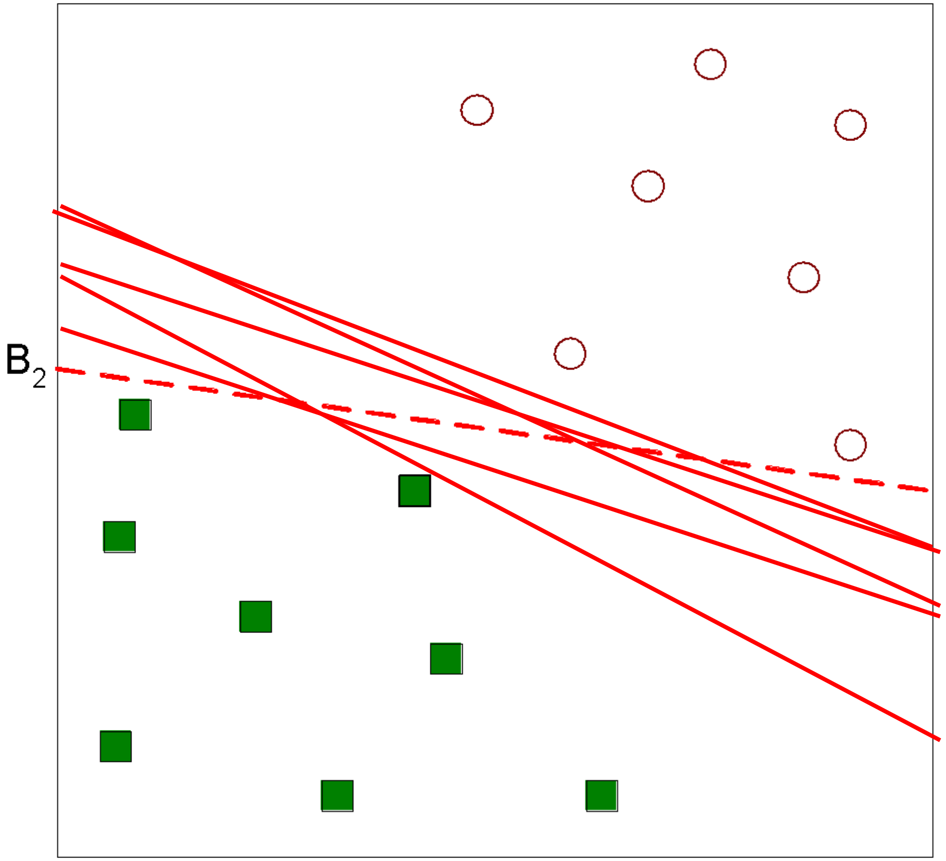
- Muhtemel Bir Çözüm



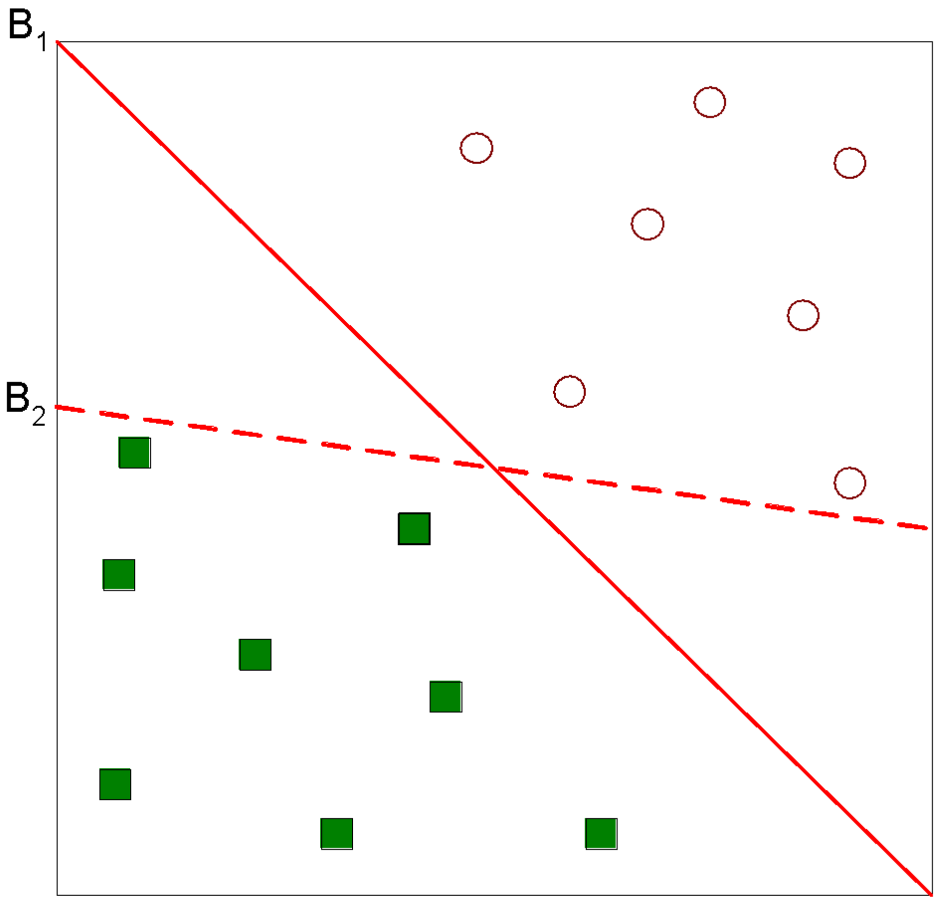
- Diğer Muhtemel Bir Çözüm



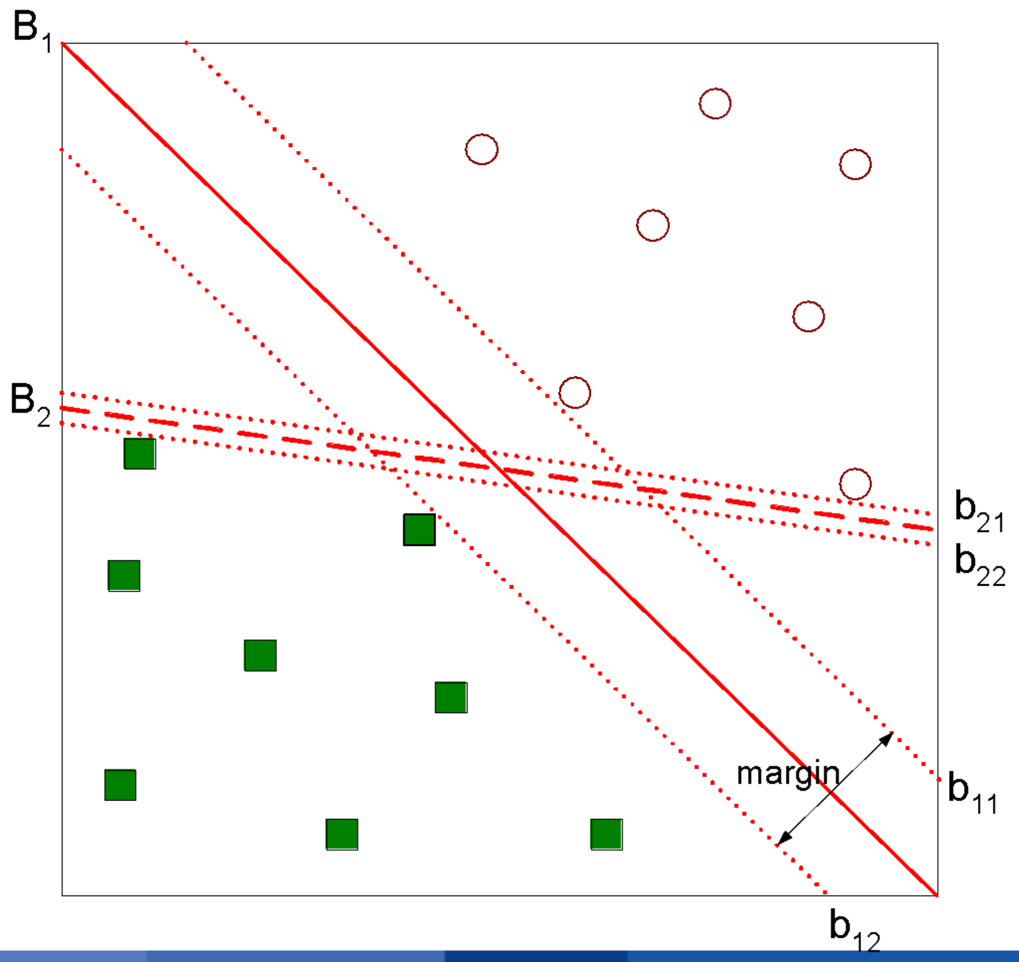
- Birçok muhtemel çözüm



# B1 ya da B2'den hangisi daha iyi? Daha iyiyi nasıl tanımlıyoruz.

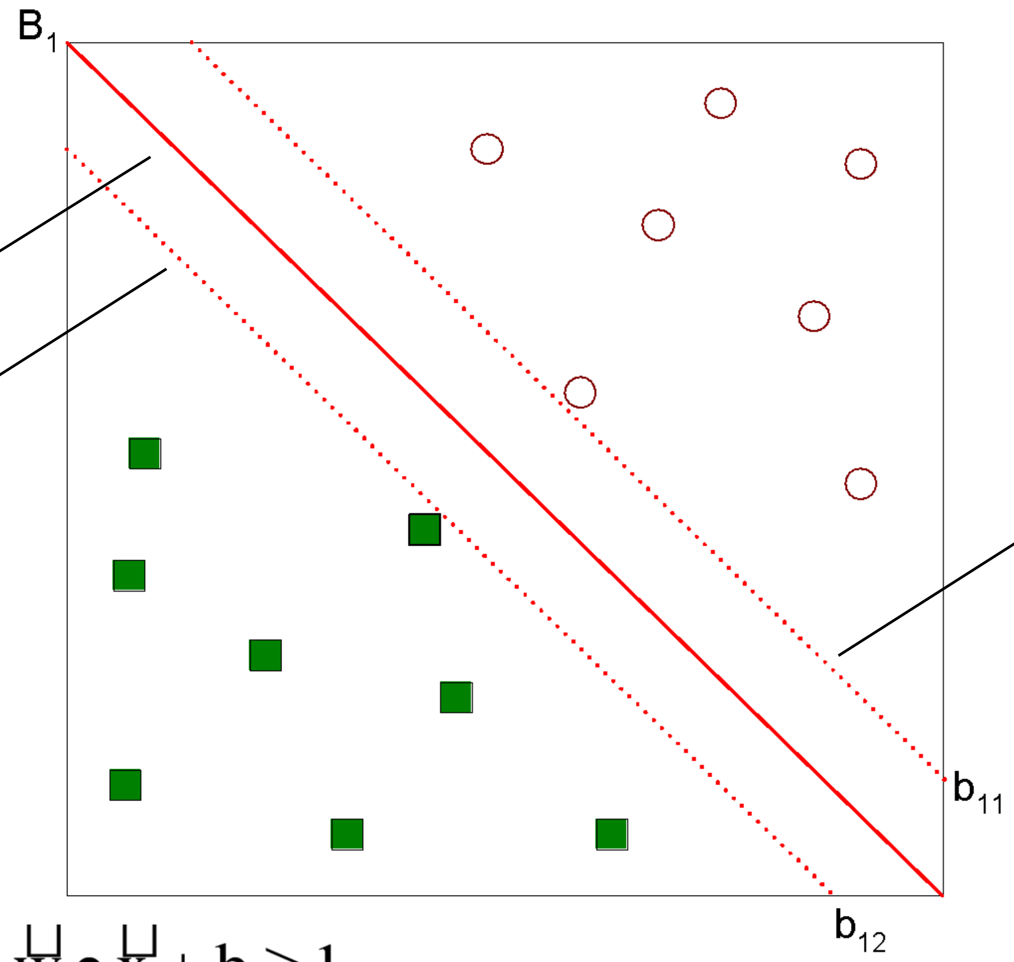


Buradaki aralığı maksimize eden çubuğu bulalım => B1 B2’den iyidir.



$$\vec{w} \bullet \vec{x} + b = 0$$

$$\vec{w} \bullet \vec{x} + b = -1$$




$$\vec{w} \bullet \vec{x} + b = +1$$

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\|\vec{w}\|}$$

Hiper düzlemin üst tarafında kalan noktalar :

$$W^T X + b > 0, \quad y_1 = +1$$

Hiper düzlemin alt tarafında kalan noktalar ise

$W^T X + b < 0, \quad y_2 = -1$ eşitsizliğine uyar.

Eşitsizlikleri brleştirilerek tek bir eşitsizliğe dönüşür:

$$y_i (W^T X + b) - 1 \geq 0$$

- Bir destek vektör ile $H_0$ : $W^T X + b = 0$ ile gösterilen hiper düzlemi arsındaki uzaklık, P hiper düzlemi üzerindeki bir nokta olmak üzere:
- d= $\frac{|WX'_p \pm b|}{\|w\|} = \frac{|w_1 x_{1P+} w_2 x_{2P+\cdots} + w_n x_{nP} + b|}{\sqrt{w_1^2 + w_2^2 + \cdots \ldots + w_n^2}}$
- O halde $X_1$ destek vektörü ile $H_2$ hiper düzlemi arasındaki uzaklık:
- d= $\frac{|W^T X_1 + b|}{\|w\|} = \frac{1}{\|w\|}$ biçiminde hesaplanır.

Bu durumda m= 2d= $\frac{2}{\|w\|}$ olur.

# Lineer DVM

- Lineer Model: $$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

- Modelin öğrenmesi $\vec{w}$ and $b$ iki değerin belirlenmesi ile gerçekleşir
- Bu değerler $\vec{w}$ and $b$ nasıl belirlenecek?

Amaç Maksimizasyon: $\text{Margin} = \frac{2}{\|\vec{w}\|}$

Burada minizasyonu yapılacak: $L(\vec{w}) = \frac{\|\vec{w}\|^2}{2}$

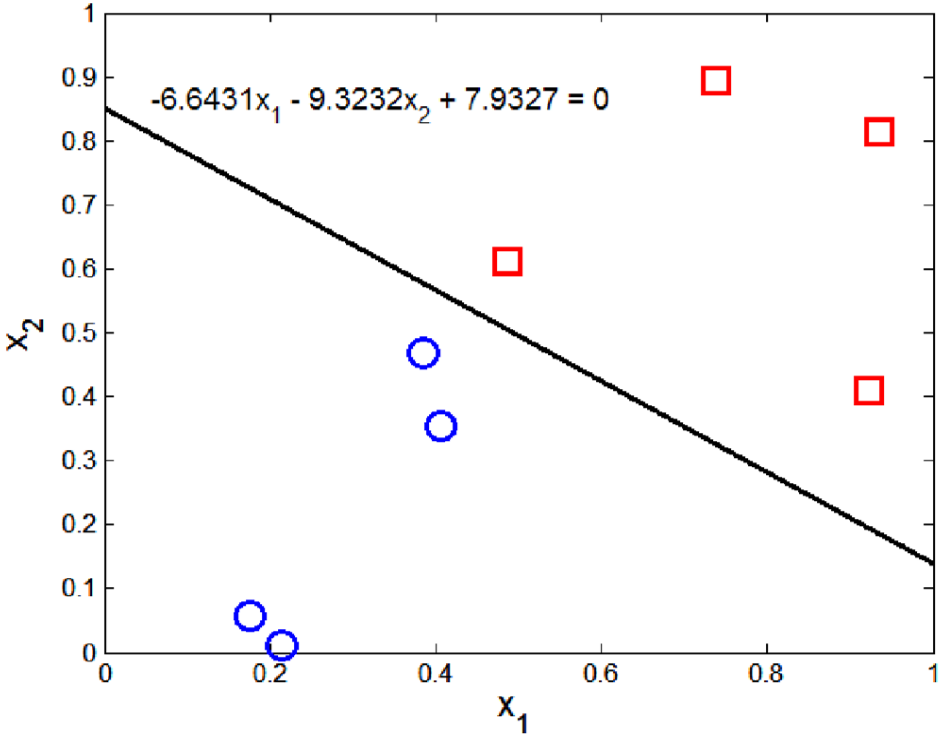
Kısıtlar şu şekildedir:

ya da $y_i = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_\mathrm{i} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_\mathrm{i} + \mathrm{b} \leq -1 \end{cases}$

$$y_i(\mathbf{w} \bullet \mathbf{x}_i + b) \geq 1, \quad i = 1,2,\ldots,N$$

Bu problem kısıtlı optimizasyon problemidir.

# Örnek bir çalışma

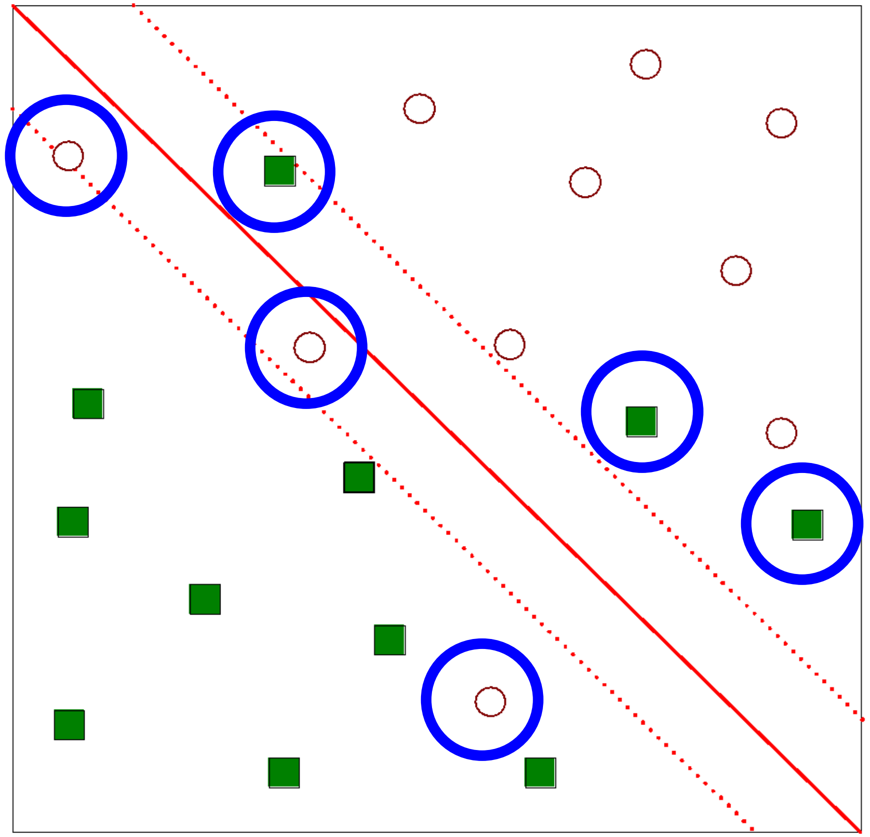



**Destek Vektörleri**

| x1 | x2 | y | λ |
|---|---|---|---|
| 0.3858 | 0.4687 | 1 | 65.5261 |
| 0.4871 | 0.611 | -1 | 65.5261 |
| 0.9218 | 0.4103 | -1 | 0 |
| 0.7382 | 0.8936 | -1 | 0 |
| 0.1763 | 0.0579 | 1 | 0 |
| 0.4057 | 0.3529 | 1 | 0 |
| 0.9355 | 0.8132 | -1 | 0 |
| 0.2146 | 0.0099 | 1 | 0 |

# DVM

- Peki ya problem tam doğrusal olarak ayrılmazsa?



- O zaman slack değişkenler devreye girer
  - Yani problem şuna döner
    - Minimizasyon:

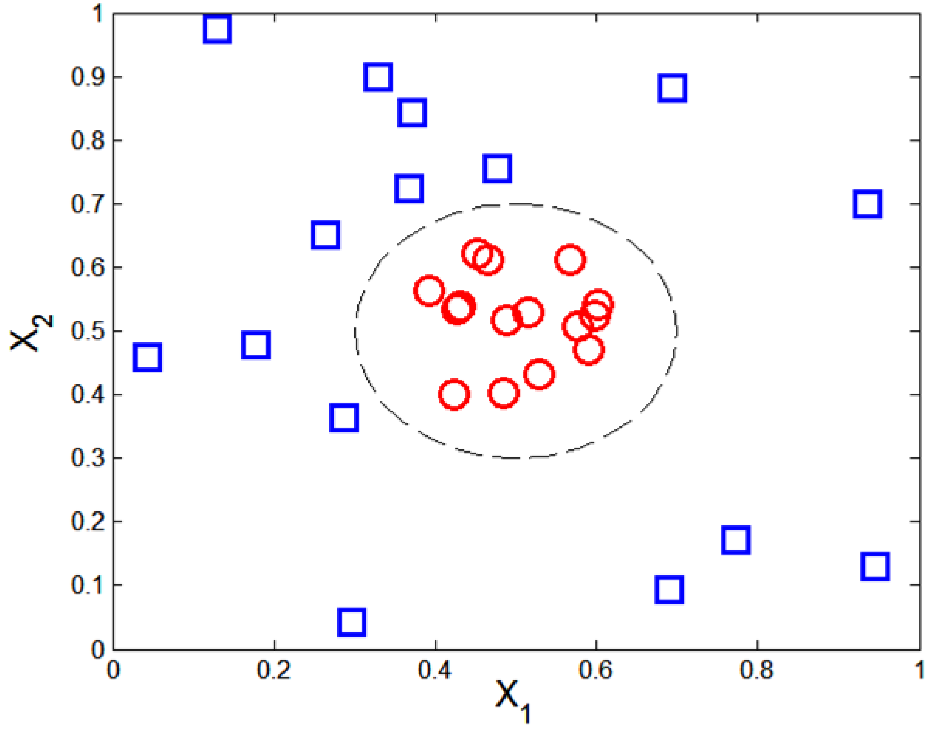
$$L(w) = \frac{\| \vec{w} \|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i^k\right)$$

    - Kısıtlar:

$$y_i = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

# Lineer Olmayan DVM

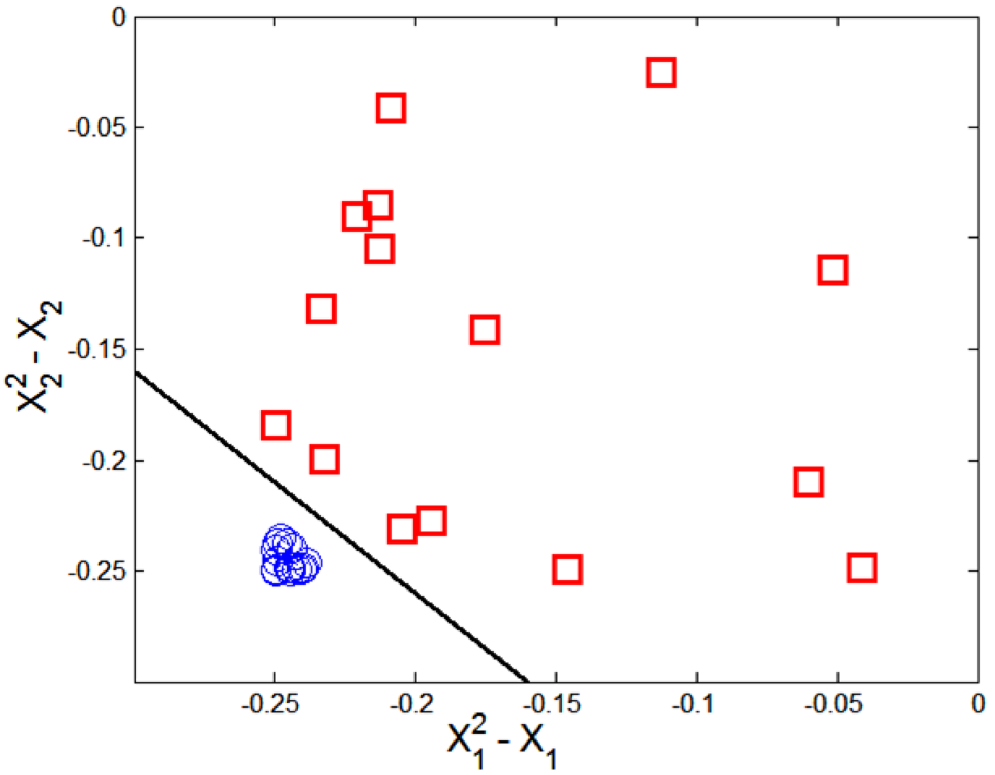
- Peki ya sınırlar lineer değilse?




$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2 \\ -1 & \text{otherwise} \end{cases}$$

# Lineer Olmayan SVM

- O zaman veriler daha büyük bir veri uzayına taşınır.




$$x_1^2 - x_1 + x_2^2 - x_2 = -0.46.$$

$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

$$w_4 x_1^2 + w_3 x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0.$$

**Sınırlar:**

$$\vec{w} \bullet \Phi(\vec{x}) + b = 0$$

# Lineer Olmayan DVM

- Optimizasyon Problemi:

$$\min_{w} \frac{\|\mathbf{w}\|^2}{2}$$

$$\textit{subject to} \quad y_i(w \cdot \Phi(x_i) + b) \geq 1, \ \forall\{(x_i, y_i)\}$$

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

$$\mathbf{w} = \sum_{i} \lambda_i y_i \Phi(\mathbf{x}_i)$$

$$\lambda_i \{y_i (\sum_{j} \lambda_j y_j \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i) + b) - 1\} = 0,$$

$$f(\mathbf{z}) = sign(\mathbf{w} \cdot \Phi(\mathbf{z}) + b) = sign(\sum_{i=1}^{n} \lambda_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{z}) + b).$$

# Kaynaklar


- Akküçük, Ulaş. "Veri madenciliği: kümeleme ve sınıflama algoritmaları". *İstanbul: Yalın Yayıncılık* 18 (2011).
- Han, Jiawei, Jian Pei, ve Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.
- Kantardzic, Mehmed. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.
- Sumathi, Sai, ve S. N. Sivanandam. *Introduction to data mining and its applications*. C. 29. Springer, 2006.
- Tan, Pang-Ning, Michael Steinbach, ve Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.
- Towards Data Science. "Towards Data Science". Erişim 29 Mart 2020. https://towardsdatascience.com/.
- VanderPlas, Jake. *Python Data Science Handbook. OReilly Media*. Inc, 2017.