Support Vector Machines

Atil Samancioglu

1 Giriş

Support Vector Machines (SVM), hem sınıflandırma hem de regresyon problemlerinde kullanılan güçlü bir makine öğrenimi algoritmasıdır. Bu belgede, SVM'in temel mantığı, geometrik yorumu, matematiksel temelleri, varyantları (SVC, SVR), kernel triklerinin kullanımı ve optimizasyon yaklaşımı detaylı bir şekilde ele alınacaktır.

2 Vektörel Gösterim: $\mathbf{w}^\top \mathbf{x} + b = 0$ Ne Anlama Geliyor?

Önceki bölümlerde, doğrusal modelleri tanımlarken genellikle şu tür basit gösterimler kullandık:

$$y = \theta_0 + \theta_1 x \quad \text{veya} \quad ax + b = 0$$

Ancak Support Vector Machine (SVM) gibi daha gelişmiş algoritmalarda ve genel vektör uzaylarında, bu doğrusal model şu şekilde gösterilir:

$$\mathbf{w}^\top \mathbf{x} + b = 0$$

Bu gösterim şu anlama gelir:

- $\mathbf{w}^\top$ — Ağırlıklar vektörünün transpozesi. Örneğin iki boyutlu bir problemde: $\mathbf{w}^\top = [w_1 \quad w_2]$
- $\mathbf{x}$ — Girdi özelliklerinin vektörü. Örneğin $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
- b — Sabit terim veya bias.

İki boyutlu bir örnekte:

$$\mathbf{w}^\top \mathbf{x} = w_1 x_1 + w_2 x_2$$

Böylece denklemin tamamı şu hale gelir:

$$w_1 x_1 + w_2 x_2 + b = 0$$

Bu, iki boyutlu bir düzlemde bir doğruyu, üç boyutlu bir uzayda bir düzlemi, daha yüksek boyutlarda ise bir hiperdüzlemi tanımlar. SVM algoritmasının amacı, bu hiperdüzlemi en uygun şekilde yerleştirerek sınıflar arasındaki mesafeyi maksimize etmektir.

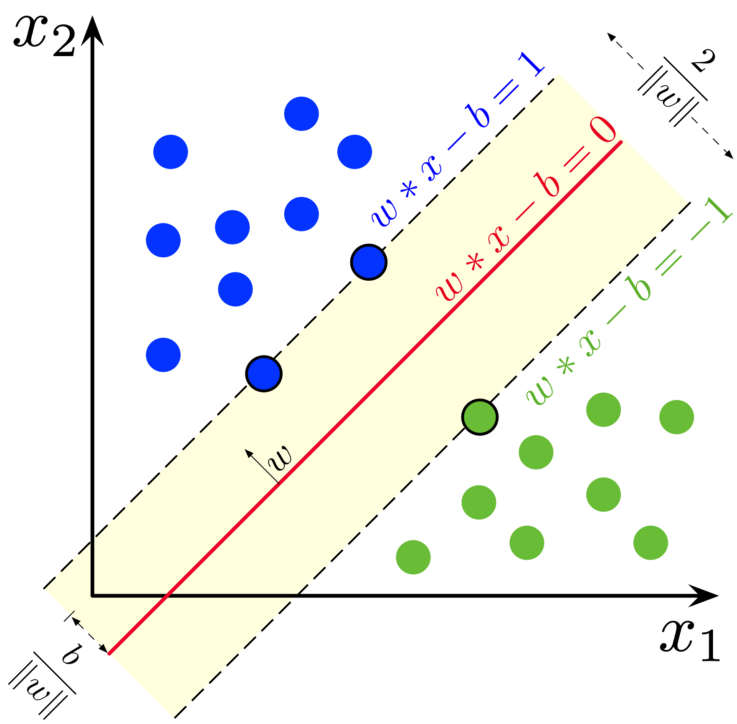

Figure 1: SVM Wikipedia Görseli

3 Support Vector Classifier (SVC)

3.1 Amaç

Veri noktalarını bir düzlem veya hiper düzlem yardımıyla ayrıştırmak. Bu düzlem:

$$\mathbf{w}^\top \mathbf{x} + b = 0$$

olarak ifade edilir.

3.2 Marjinal Düzlemler

Veri setini iki sınıfa ayıran bu düzlemden eşit uzaklıktaki iki düzlem:

$$\mathbf{w}^\top \mathbf{x} + b = +1 \quad \text{ve} \quad \mathbf{w}^\top \mathbf{x} + b = -1$$

Bu iki düzlem arasındaki mesafe (margin):

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

3.3 Optimizasyon Problemi

Amaç: margin'i maksimize etmek $\Rightarrow \|\mathbf{w}\|$ minimizasyonu.

Kısıt: Tüm doğru sınıflandırılmış noktalar için:

$$y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 \quad \forall i$$

Optimizasyon problemi:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1$$

3.4 Hard Margin vs Soft Margin

Support Vector Machine (SVM), verileri en iyi şekilde ayıran bir hiperdüzlem bulmayı hedefler. Ancak gerçek dünyada veriler mükemmel bir şekilde ayrılmayabilir. Bu durumda **soft margin** yaklaşımı devreye girer.

Soft margin SVM, verileri doğru sınıflandırırken aynı zamanda marjin dışına çıkan hatalı örnekleri cezalandırmak için **hinge loss** fonksiyonunu kullanır.

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned}$$

Burada:

- C: ceza katsayısı
- ξ_i: kısıt ihlalini temsil eden slack değişkeni

4 Support Vector Regression (SVR)

Support Vector Regression (SVR), Support Vector Machine algoritmasının regresyon problemlerine uyarlanmış halidir. Sınıflandırma yerine sürekli (reel sayı) değerleri tahmin etmek için kullanılır. Ancak SVR'nin yaklaşımı klasik regresyon tekniklerinden farklıdır.

SVR vs SVC

SVC (Support Vector Classifier) sınıflar arasında **en geniş marjini ayıran bir hiper düzlem** bulmaya çalışır. SVR ise aynı yaklaşımı kullanarak **tahmin eğrisi etrafında toleranslı bir tüp** tanımlar. Amaç, bu tüp içinde kalan tahminlerin hatasını yok saymak ve yalnızca dışındaki sapmaları cezalandırmaktır.

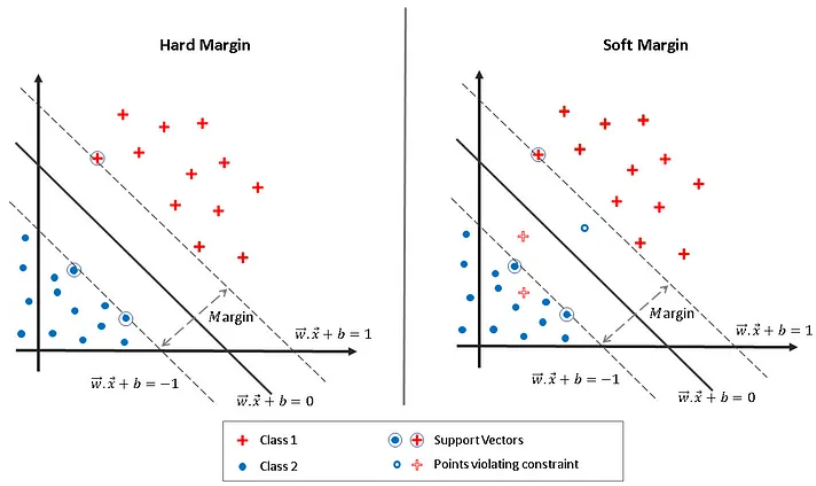

Figure 2: Suraj Donthi Hard Margin Görseli

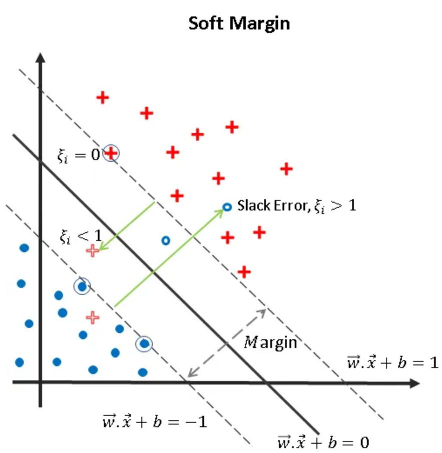

Figure 3: Suraj Donthi Soft Margin Görseli

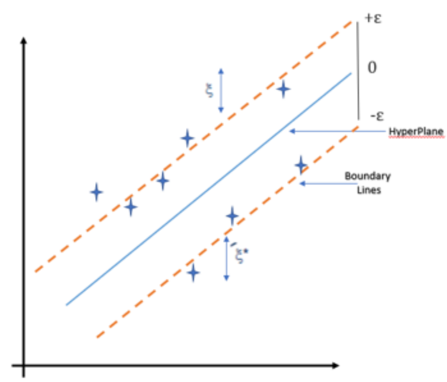

Figure 4: SVR Epsilon Tunnel

SVR vs Lojistik veya Lineer Regresyon

Lineer regresyon, her bir noktanın **kare hatasını** minimize etmeye çalışır. Yani her tahminin ne kadar saptığını önemser. SVR ise daha esnektir: Tahmin tüpü içinde kalan sapmaları görmezden gelir. Bu sayede **aşırı uç değerlere karşı daha dayanıklı** olur.

Temel Fikir

SVR, tahmin eğrisi etrafında ϵ genişliğinde bir tüp oluşturur. Bu tüpün içinde kalan tahminler "yeterince iyi" kabul edilir ve ceza uygulanmaz. Tüp dışında kalan tahminlere ceza verilir.

$$\hat{y} = \theta^\top x + b$$

SVR'nin farkı, bu tahminin çevresinde iki tolerans çizgisi (marjin) belirlemesidir:

$$\hat{y} + \epsilon \tag{1}$$

$$\hat{y} - \epsilon \tag{2}$$

Buradaki ϵ, modelin göz ardı edebileceği maksimum hata miktarını gösterir.

SVR Kayıp Fonksiyonu (Loss Function)

Epsilon-insensitive loss fonksiyonu olarak adlandırılır. Eğer gerçek değer ile tahmin arasındaki fark ϵ'den küçükse ceza verilmez.

$$L(y, \hat{y}) = \begin{cases} 0 & \text{if } |y_i - \hat{y}_i| \leq \epsilon \\ |y_i - \hat{y}_i| - \epsilon & \text{otherwise} \end{cases} \tag{3}$$

Bu kayıp fonksiyonu, modeli yalnızca "tüp dışına" çıkan tahminler için cezalandırır.

Maliyet Fonksiyonu (Cost Function)

SVR'nin optimize etmeye çalıştığı hedef şudur:

$$J(\theta) = \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{4}$$

- $\frac{1}{2}\|\theta\|^2$: Eğrinin düz (yumuşak) olmasını sağlar.
- C: Ceza katsayısı – marjin dışındaki hatalara ne kadar tolerans gösterileceğini belirler.
- ξ_i, ξ_i^*: Yukarı ve aşağı yönde marjin dışına çıkan veriler için ceza değişkenleri.

SVR'nin Gücü

SVR özellikle şu durumlarda faydalıdır:

- Aykırı değerlere karşı dayanıklı bir regresyon modeli gerektiğinde
- Küçük toleranslarla (epsilon) hatasız tahminler istendiğinde
- Marjin kontrolü sayesinde sadeleştirilmiş, aşırı öğrenmeye karşı dirençli bir model tercih edildiğinde

5 Kernel Trikleri ve Lineer Olmayan Ayırımlar

Lineer olarak ayrıştırılamayan veriler için SVM, veri uzayını daha yüksek boyutlara dönüştürerek ayırımı mümkün kılar. Bu işlem **kernel trick** kullanılarak yapılır.

5.1 Kernel Fonksiyonları

- **Lineer Kernel:** $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$
- **Polinomial Kernel:** $K(\mathbf{x}, \mathbf{x}') = (\gamma \mathbf{x}^\top \mathbf{x}' + r)^d$
- **RBF (Gaussian):** $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$
- **Sigmoid Kernel:** $K(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \mathbf{x}^\top \mathbf{x}' + r)$

6 Özet

Support Vector Machines güçlü sınıflandırma ve regresyon algoritmalarıdır. Margin'i maksimize ederek ve kernel trikleri kullanarak doğrusal olmayan veri setlerinde yüksek başarı sağlayabilir.

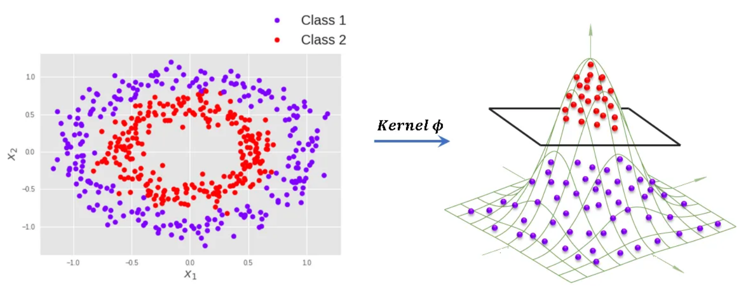

Figure 5: Suraj Donthi Kernel Görseli