

• Yeni loss'ın w ve b ye göre kısmi türevlerini alacağım.

5) Ağırlıkların güncellenmesi: Gradient descent algoritması ile w ve b güncellenir.

$$\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1} \rightarrow \frac{\partial L}{\partial w_1} \text{ e göre türevini bulacağım. (Geri yayılım)}$$

▼ Hatırlatma

$$\frac{\partial \log x}{\partial x} = \frac{1}{x} \text{ 'dır.}$$

$$L(\hat{y}, y) = - (y \log \hat{y} + (1-y) \log (1-\hat{y})) \rightarrow L(\hat{y}, y)$$

$$\frac{\partial L}{\partial \hat{y}} = \frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$$

$$\sigma = \frac{1}{1+e^{-z}} \rightarrow \frac{\partial \hat{y}}{\partial z}$$

$$\frac{\partial z}{\partial w_1} = x_1$$

Çaptık

$$\frac{\partial L}{\partial w_1} = (\hat{y} - y) x_1$$

$$\frac{\partial L}{\partial b} = (\hat{y} - y)$$

St gradient descent:

$$w_1 \leftarrow w_1 - \alpha (\hat{y} - y) x_1$$

$$b \leftarrow b - \alpha (\hat{y} - y)$$

adım büyüklüğü \leftarrow hyperparametre hızla okula

Lojistik Regresyon Hyperparametreleri

• Penalty (cezalandırma): Overfitting önlemeye yardımcıdır. Cezalı terimleri $L_1 \rightarrow$ Geresiz katsayıları '0' a cezalar. Değişken seçimi yapar. $L_2 \rightarrow$ Bütün katsayıları küçültür ama sıfıra sezmek.

• (cezalandırma katsayısı) : Regularizasyonun şiddetini kontrol eder. Deha yazar C daha az regularizasyon anlamına gelir. Modelin optim. verisine uyma neden olur.

- Solver (Çözücü) : Modelin optimize edilmesinde kullanılan algoritmayı belirler.
- max iterations : Maksimum iterasyonlar.
- Class Weight : Dengesiz sınıfların olmasına bağlı olarak sınıflara farklı parametreler verilebilir.

Bilgi \rightarrow LR 'a NonLinear katmanlar eklersek bu yöntem derin öğrenmeye girer.

Zayıf Yöntemler:

• Lineer modeldir, kuramsal ilişkileri modelleyemez.

• Basit özellikler ve yazar. Bağıtli veri setleriyle eğitildiğinde aşırı uydurmaya eğilimlidir.

• Sınıflandırma problemi çözer. Regresyon çözenler.

Güçlü Yöntemler:

• Basit ve yazar kuramları birleştirir.

• Regularizasyon ile overfitting (Lineer ile kontrol edilebilir).

• Sınıf olasılıkları ve tahminleri sağlar.

Destek Vektör Makinesi

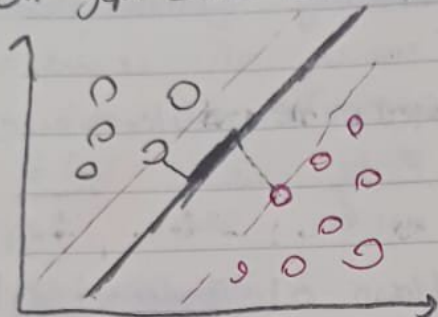
SVM (Support Vektör M)

• SVM, sınıflandırma ve regresyon problemlerini çözmek için güçlü bir algoritmadır.

• SVM, veri uzayını sınıflar arasında en iyi şekilde ayıran bir hiperdüzlemi bulmaya çalışır.

• LR, lineer olduğu için kuramsal verileri ayıramaz. SVM, doğrusal olmayan kuramsal verileri ayırmak için de etkilidir.

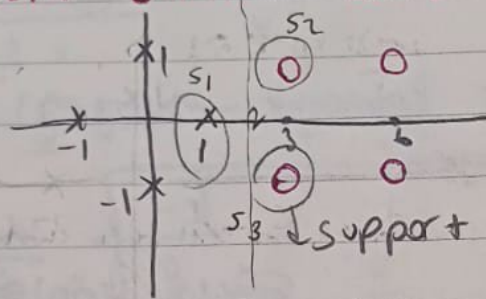
Görünüşte kernel yöntemi kullanılarak veriyi daha yavaşça boyutlu uzaylara dönüştürerek doğrusal olarak ayrılabilir bir yapı elde edilebilir.



• En yakın 2 sınıf veriler arasındaki mesafeyi en çok yakmayla ediliyor.

etiket

SVM Durumu = 2 (Cakışması) $0 \rightarrow p \rightarrow +1$
 $x \rightarrow n \rightarrow -1$



Support vektör.

$$s_1 = (1, 0) \quad s_2 = (3, 1) \quad s_3 = (3, -1)$$

$$\tilde{s}_1 = (1, 0, 1) \quad \tilde{s}_2 = (3, 1, 1) \quad \tilde{s}_3 = (3, -1, 1)$$

$[\alpha_1, \alpha_2, \alpha_3] \rightarrow$ Bunlar parametre hesaplamak için.

$$\alpha_1 \cdot \tilde{s}_1 \cdot \tilde{s}_1 + \alpha_2 \cdot \tilde{s}_2 \cdot \tilde{s}_1 + \alpha_3 \cdot \tilde{s}_3 \cdot \tilde{s}_1 = -1$$

$$\alpha_1 \cdot \tilde{s}_1 \cdot \tilde{s}_2 + \alpha_2 \cdot \tilde{s}_2 \cdot \tilde{s}_2 + \alpha_3 \cdot \tilde{s}_3 \cdot \tilde{s}_2 = +1$$

$$\alpha_1 \cdot \tilde{s}_1 \cdot \tilde{s}_3 + \alpha_2 \cdot \tilde{s}_2 \cdot \tilde{s}_3 + \alpha_3 \cdot \tilde{s}_3 \cdot \tilde{s}_3 = +1$$

$$\rightarrow \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

$$= 2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$$

$$\rightarrow = 4\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1$$

$$\rightarrow = 4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1$$

$$\begin{bmatrix} 2 & 4 & 4 \\ 4 & 11 & 9 \\ 4 & 9 & 11 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

$$\alpha_1 = -3.5 \quad \alpha_2 = 0.75 \quad \alpha_3 = 0.75$$

$$w = \sum_{i=1}^3 \alpha_i \tilde{S}_i = -3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}$$

$\begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix} \rightarrow b$ w ağırlığımız line $x=0$ noktasından -2 birim aşağıdır.

Destek Vektör makineleri hiperparametreleri

* C : Doğruluk ve aşırı uyum arasındaki dengeyi kontrol eder.

* Kernel tipi : Eğer elimizdeki veriler doğrusal değilse (non linear) ise doğrusal olmamıza problem oluşturur ve non linear yapıya ihtiyacımız varsa kernel tipini doğrusal dan non lineere değiştirebiliriz.

Uygulama Alanı

Tıp, Finans, Görüntü işleme, Biyoinformatik, Metin ve Dil işleme

Zayıf Yöntemler
Overfitting olabilir (veri seti çok büyükse)

Güçlü Yöntemler
Doğrusal ve doğrusal olmayan sınıflandırma problemlerinde etkilidir