Makine Öğrenmesi

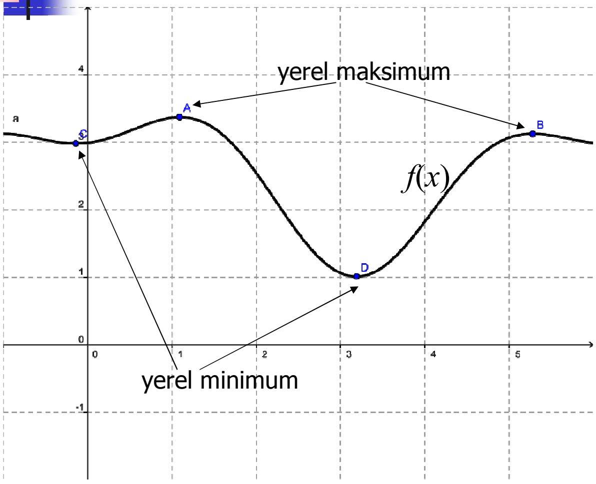
10. hafta

- Lagrange Optimizasyonu
- Destek Vektör Makinesi (SVM)
- Karesel (Quadratic) Programlama

Optimizasyon

Bilimsel terim olarak dilimize geçmiş olsa da bazen "en iyileme" terimiyle karşılık bulur. Matematikteki en basit kullanımıyla bir fonksiyonun alabileceği en büyük ve en küçük değerleri metodolojik olarak bulabilme işlemidir. Yerel minimum ve maksimumu bulabilmek için bilinen en temel yöntem, fonksiyonun birinci dereceden türevinin alınmasıdır.

Yerel Minimum-Maksimum



$$f' = \frac{\partial f}{\partial x} = 0$$

Yerel Minimum-Maksimum

Her yerel minimum ve maksimum noktasında birinci dereceden türev sıfırdır fakat türevin sıfır olduğu her nokta yerel minimum veya maksimum olmayabilir. Bunun için ikinci derece türev incelenmelidir.

Eğer $f''(x) < 0$ yerel maksimum

Eğer $f''(x) > 0$ yerel minimum

Lagrange Optimizasyonu

Bir fonksiyonun en büyük veya en küçük değerlerini bir kısıta bağlı olarak bulmak gerektiğinde uygulanan en temel yöntemdir. Lagrange fonksiyonu, optimize edilecek olan amaç (objective) $f(x)$ fonksiyonuna kısıt (constraint) $g(x)$ teriminin bir α katsayısı oranında eklenmesiyle hazırlanır.

$$L(x,\alpha) = f(x) + \alpha\, g(x)$$

Lagrange Optimizasyonu

Tek değişkenli fonksiyonlarda yerel min-max değerlerinin bulunmasında olduğu gibi Lagrange fonksiyonunun da bağlı değişkenlere göre türevleri sıfıra eşitlenerek çözüm aranır.

$$\frac{\partial L(x,\alpha)}{\partial x} = 0$$

$$\frac{\partial L(x,\alpha)}{\partial \alpha} = 0$$

Örnek

$f(x,y)=25-x^2-y^2$ denkleminin $4-x-y=0$ şartını sağlayan minimum değerini bulalım.

$$L(x,y,\alpha)=25-x^2-y^2+\alpha(4-x-y)$$

$$\frac{\partial L(x,y,\alpha)}{\partial x}=-2x+\alpha=0$$

$$\frac{\partial L(x,y,\alpha)}{\partial y}=-2y+\alpha=0$$

$$\frac{\partial L(x,y,\alpha)}{\partial \alpha}=4-x-y=0$$

Örnek

$f(x,y)=25-x^2-y^2$ denkleminin $4-x-y=0$ şartını sağlayan minimum değerini bulalım.

$$x=0.5\alpha \qquad \alpha=4$$

$$y=0.5\alpha \qquad x=2$$

$$x+y=4 \qquad y=2$$

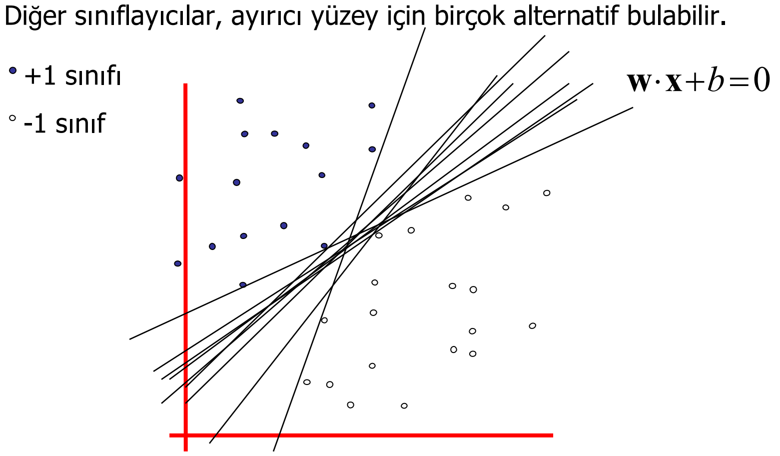
$f(x=2,y=2)=25-2^2-2^2=17$ bulunur.

Destek Vektör Makinesi (SVM)

Destek Vektör Makinesi (support vector machine - SVM) veriyi sınıflandırırken sınıfların birbirlerine en yakın örneklerini bularak bu örneklerin (iki sınıfı ayıracak olan) ayırıcı yüzeye dik uzaklıklarını maksimize etmeyi amaçlar. Ayırıcı yüzeyin, veri kümesi üzerindeki başarısı değişmeden birçok farklı alternatifi olabilir. SVM sayesinde ayırıcı yüzey her iki sınıfa da aynı mesafede ve maksimum uzaklıktadır.

Destek Vektör Makinesi (SVM)

Diğer sınıflayıcılar, ayırıcı yüzey için birçok alternatif bulabilir.

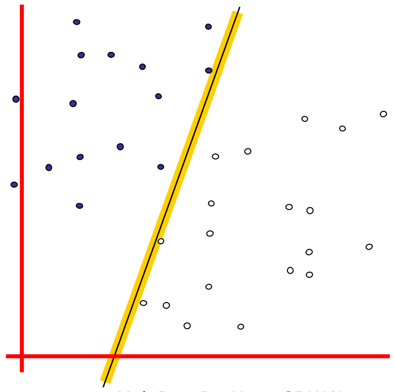


Destek Vektör Makinesi (SVM)

Bunların bir kısmı sınıflar arası uzaklığı maksimize edemez.

- +1 sınıfı
- -1 sınıf

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

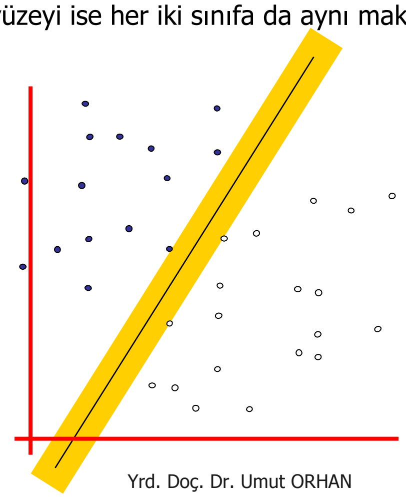


Destek Vektör Makinesi (SVM)

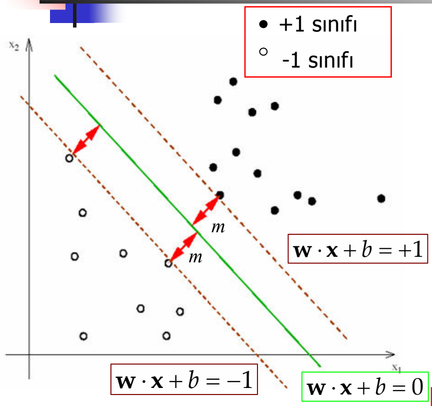
SVM ayırıcı yüzeyi ise her iki sınıfa da aynı maksimum uzaklıktadır.

- +1 sınıfı
- -1 sınıf

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$



Destek Vektör Makinesi (SVM)



$$d_i = +1 \text{ için} \qquad w \cdot x_i + b \geq +1$$
$$d_i = -1 \text{ için} \qquad w \cdot x_i + b \leq -1$$

$$m = \frac{2}{\sqrt{w \cdot w}} \rightarrow f_{\min}(w) = \frac{w \cdot w}{2}$$
$$d_i(w \cdot x_i + b) \geq 1, \quad 1 \leq i \leq N$$

$$\boxed{L(w,b,\alpha) = \frac{w.w}{2} - \sum_i \alpha_i \left(d_i (w \cdot x_i + b) - 1\right)}$$

Destek Vektör Makinesi (SVM)

$$L(w,b,\alpha) = \frac{w.w}{2} - \sum_i \alpha_i \left(d_i (w \cdot x_i + b) - 1\right)$$

Yukarıdaki Lagrange denkleminin bazı veri kümeleri için geleneksel çözümü belki de çok kolay olabilir. Fakat bu denkleme bilgisayar destekli bir çözüm üretebilmek için farklı bir yol izlenmelidir.

Destek Vektör Makinesi (SVM)

$$\frac{\partial L(w,b,\alpha)}{\partial w}=0 \qquad \frac{\partial L(w,b,\alpha)}{\partial b}=0 \qquad \frac{\partial L(w,b,\alpha)}{\partial \alpha}=0$$

Lagrange denkleminin geleneksel uygulamasına göre yukarıdaki üç eşitliği ortak bir hesaplamada değerlendirmek gerekmektedir. Bunun yerine teorik olarak ortak bir ilişki keşfedilerek çözüm yöntemindeki üç değişkene bağımlılık bire indirgenebilir.

Destek Vektör Makinesi (SVM)

Klasik çözüm uygulanarak Lagrange denklemini tek değişkene bağımlı hale getirmeliyiz.

$$\frac{\partial L(w,b,\alpha)}{\partial w}=0 \rightarrow \qquad w=\sum_i \alpha_i d_i x_i$$

$$\frac{\partial L(w,b,\alpha)}{\partial b}=0 \rightarrow \qquad \sum_i \alpha_i d_i = 0$$

bulunur. Bu ifadeler Lagrange denkleminde yerine yazılır.

Destek Vektör Makinesi (SVM)

Lagrange denklemi bu sayede tek değişkene bağımlı hale getirilir.

$$L(w,b,\alpha)=\frac{w.w}{2}-\sum_i \alpha_i\left(d_i\left(w\cdot x_i+b\right)-1\right)$$

$$L(\alpha)=\sum_i \alpha_i-\frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j d_i d_j x_i x_j$$

Haritalama

Doğrusal olmayan veri kümeleri seçilen bir çekirdek (kernel) fonksiyonu ile bir üst uzaya haritalandıktan sonra doğrusal SVM çözümü uygulanabilir.

$$L(\alpha)=\sum_i \alpha_i-\frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j d_i d_j K(x_i,x_j)$$

Haritalama

En sık kullanılan çekirdek fonksiyonları doğrusal (linear), polinom (polynomial) ve RBF fonksiyonlarıdır.

$$K(x_i, x_j) = \begin{cases} x_i x_j & linear \\ (1 + x_i x_j)^p & polynomial \\ \exp\left(-\frac{1}{2}\|x_i - x_j\|^2\right) & RBF \end{cases}$$

Karesel (Quadratic) Programlama

Tek değişkene bağlı Lagrange denklemini çözmek için geriyayılım, benzetimli tavlama ve EM gibi yöntemler kullanılabilir. Yine de en çok tercih edilen çözüm Karesel (Quadratic) Programlamadır.

Karesel (Quadratik) terim

$$\min_u \left(\frac{u^T R u}{2} + d^T u + c \right)$$

Karesel Programlama

Karesel Programlama optimizasyon tekniğini tek değişkenli Lagrange denklemine uyarlanırsa;

$$L(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j d_i d_j x_i x_j$$

$$R \approx -D^T D\, X^T X \qquad \text{ve} \qquad u \approx \alpha_i$$

Karesel Programlama

Karesel Programlamanın temeli aslında geri-yayılıma benzeyen iterasyonal bir yaklaşımdır. Rastsal başlangıç değerlerini sürekli güncelleyerek optimum *u* matrisi bulunmaya çalışılır. Kısıtlar sayesinde geri-yayılımdaki başlangıç değerlerine bağımlılık kaldırılmıştır.

$$\min_u \left(\frac{u^T R u}{2} + d^T u + c \right)$$

MATLAB Uygulaması

```
>edit SVM_ornek.m
```

Hazırlanmış olan farklı datasetler yüklenerek Destek Vektör Makinesi yardımıyla sınıflandırılmaktadır. Matlab `svmtrain` ve `svmclassify` komutlarını kullanarak farklı örnekler yapılmalıdır.

ÖDEV

Karesel Programlama yerine geri-yayılım tekniğini kullanarak herhangi bir veri kümesi üzerinde tek değişkenli Lagrange denklemini minimize eden α_i değerlerini bulunuz.