Hafta 05 - Destek Vektör Makinesi ve Dengesiz Sınıf Dağılımı

SİB 552 - Siber Güvenlik İçin Veri Madenciliği

Bilgisayar Mühendisliği

Siber Güvenlik Yüksek Lisans Programı

Dr. Ferhat Özgür Çatak

ozgur.catak@tubitak.gov.tr

Gebze Teknik Üniversitesi

2019 - Bahar

İçindekiler

İçindekiler

Destek Vektör Makinesi

Support Vector Machine

Destek Vektör Makinesi

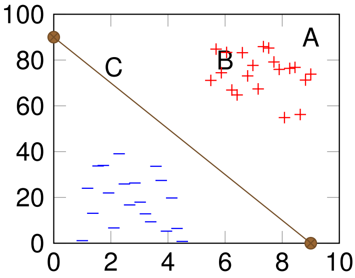
- Destek Vektör Makinesi (Support Vector Machine - SVM), en iyi denetimli öğrenme algoritması
- Ayrılabilir durumda **büyük marjlı bir sınıflandırıcı** (large-margin classifier) tanımlanması
- Doğrusal olmayan durumlarda **yumuşak kenarlı SVM** (soft-margin SVM) uygulamak için **çekirdek hilesi** (kernel-trick) kullanılır.

Doğrusal Ayrıştırılabilir Durum

The linearly separable case

Doğrusal Sınıflandırıcı

- $\mathcal{D}$ doğrusal ayrıştırılabilir ise $+$ ve $-$ örnekleri ayrıştıran bir doğru olacaktır.
- $\mathbf{w}^T\mathbf{x} = 0$ ayrıştırıcı hipredüzlem (separating hyperplane).
- **Hiperdüzlem** [1]: Boyutu, ortam uzayından **bir adet** daha az olan bir alt uzaydır.



Güven - Confidence

- Logistic regression: $p(y = 1|\mathbf{x}, \mathbf{w})$ olasılığı $h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x})$ ile modellenir.
- $h(\mathbf{x}) \geq 0.5 \Rightarrow \hat{y} = 1$
- $h(\mathbf{x}) \gg 0.5 \Rightarrow \hat{y} = 1$ veya $h(\mathbf{x}) \ll 0.5 \Rightarrow \hat{y} = 0$ ifadelerinin güveni yüksektir.

Karar sınırı (Decision Boundary) doğru ve güvenilir sonuçlar elde etmektedir.

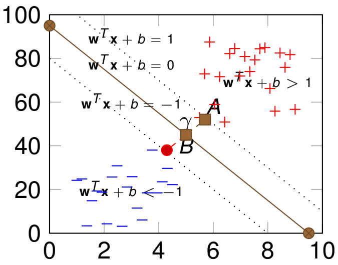
[1] http://mathworld.wolfram.com/Hyperplane.html

Marjin

Margin of a linear classifier

Tanım

- $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \cdots, (\mathbf{x}^{(m)}, y^{m}) | \mathbf{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \{-1, +1\}\}$
- **Fonksiyonel marjin** (Functional margin) $\gamma^{(i)} = y^{(i)}(\mathbf{w}^T\mathbf{x} + b)$ ve $y^{(i)}(\mathbf{w}^T\mathbf{x} + b) > 0$
  - Eğer $y^{(i)} = +1$, Fonk. marjin'in güvenilir olması için $(\mathbf{w}^T\mathbf{x} + b)$ ifadesi **büyük bir pozitif sayı** olmalı
  - Eğer $y^{(i)} = -1$, Fonk. marjin'in güvenilir olması için $(\mathbf{w}^T\mathbf{x} + b)$ ifadesi **büyük bir negatif sayı** olmalı



Karar Fonksiyonu ve Karar Sınırı

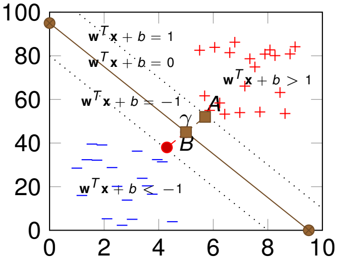
- Karar fonksiyonu (decision function)
  $h(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b)$
  - $\mathbf{w}^T\mathbf{x} + b > 0 \Rightarrow y_i = +1$
  - $\mathbf{w}^T\mathbf{x} + b < 0 \Rightarrow y_i = -1$

Marjin

Margin of a linear classifier

Tanım

- $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \cdots, (\mathbf{x}^{(m)}, y^{m}) | \mathbf{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \{-1, +1\}\}$
- **Fonksiyonel marjin** (Functional margin) $\gamma^{(i)} = y^{(i)}(\mathbf{w}^T\mathbf{x} + b)$ ve $y^{(i)}(\mathbf{w}^T\mathbf{x} + b) > 0$
  - Eğer $y^{(i)} = +1$, Fonk. marjin'in güvenilir olması için $(\mathbf{w}^T\mathbf{x} + b)$ ifadesi **büyük bir pozitif sayı** olmalı
  - Eğer $y^{(i)} = -1$, Fonk. marjin'in güvenilir olması için $(\mathbf{w}^T\mathbf{x} + b)$ ifadesi **büyük bir negatif sayı** olmalı



Hard-Margin

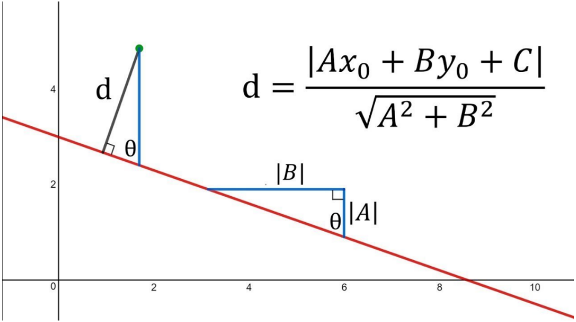
- Eğitim kümesi doğrusal ayrıştırılabilir (linearly separable) ise, aralarında uzaklık en fazla olacak şekilde birbirine paralel iki hiperdüzlem (hyperplane) seçebiliriz.
- İki hiperdüzlem arasında bulunan bölge: margin
  - $\mathbf{w}^T\mathbf{x} + b = +1$
  - $\mathbf{w}^T\mathbf{x} + b = -1$

Karar Sınırlarının Uzaklığı I

Hiperdüzlem uzaklığı

- Bir noktanın, (x_0, y_0), bir doğruya, $Ax + By + c = 0$, olan uzaklığı $\frac{|Ax_0+By_0+c|}{\sqrt{A^2+B^2}}$
- H_0 ve H_1 arasında uzaklık $\frac{|\mathbf{w}^T\mathbf{x}+b|}{||w||} = \frac{1}{||w||}$
- Bu durumda H_1 ve H_2 arasında uzaklık: $\frac{2}{||w||}$

Karar Sınırlarının Uzaklığı II



$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

Karar Sınırlarının Uzaklığı III

Optimizasyon

- Marjin maksimizasyonu için $||\mathbf{w}||$ minimize edilmesi gerekmektedir.
- H_1 ve H_2 arasında veri olmadığı kabul edersek

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x} + b &\geq +1 \, if \;\; y_i = +1 \\ \mathbf{w} \cdot \mathbf{x} + b &\leq -1 \, if \;\; y_i = -1 \end{aligned} \tag{1}$$

$$y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq +1 \;\forall\; \mathbf{x}_i \tag{2}$$

Optimizasyon Problemi I

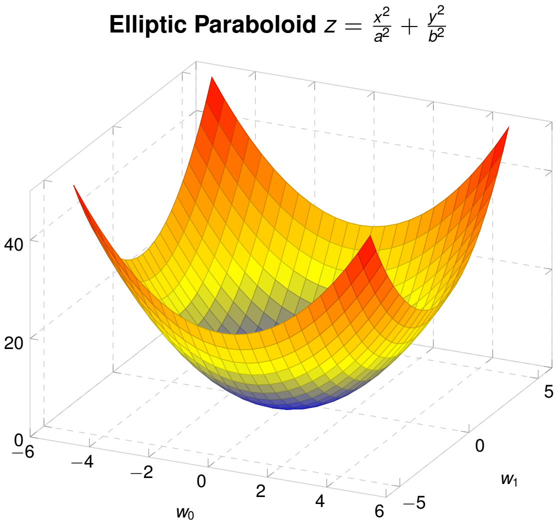
Optimizasyon Problemi

- Optimizasyon problemini değiştirelim. Yeni *quadratic optimizasyon* problemi

$$\begin{aligned} &\underset{\mathbf{w},b}{arg\ min} \frac{1}{2}||\mathbf{w}||^2 \\ &s.t:\ y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, i = 1, \cdots, m \end{aligned} \tag{3}$$

- **Hedef**: *convex quadratic*
  - Yüzey paraboloid, Tek çözüm mevcut.
- **Kısıtlar**: *linear*.
- **Çözüm**: *Lagrangian multipler method*

Optimizasyon Problemi II



Lagrange Fonksiyonu I

Lagrange multiplier

Bir fonksiyonun, $f(x, y, \cdots)$, eşitlik kısıtlarına, $g(x, y, \cdots) = c$, bağlı olarak local maxima ve minima noktalarının bulunması için bir stratejidir.

Örnek optimizasyon problemi:

$$\min_{x,y} f(\mathbf{w})$$

$$\textit{subject to } h(\mathbf{w}) = 0$$

Adım 1 Yeni değişken λ ve yeni fonksiyon $\mathcal{L}$ olmak üzere *Lagrange Fonksiyonu*

$$\mathcal{L}(\mathbf{w}, \beta) = f(\mathbf{w}) + \sum_{i=1}^{l} \beta_i h_i(\mathbf{w}) \tag{4}$$

- β_i değerleri *Lagrange çarpanlarıdır*.

Lagrange Fonksiyonu II

- $\mathbf{w}$ ve β değerlerini çözebilmek için $\mathcal{L}$'nin kısmi türevlerini alıp 0'a eşitlenmelidir.

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0;\ \frac{\partial \mathcal{L}}{\partial \beta_i} = 0 \tag{5}$$

- Dual form optimizasyon problemi

$$\max_{\alpha} \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \alpha) \tag{6}$$

Lagrange Fonksiyonu - Örnek I

Lokal mimimum ve maksimum değerlerinin bulunması
$f(x, y) = xy$ ve Eğri $3x^2 + y^2 = 6$

Çözüm
$\nabla f =< y, x >$ ve $g(x, y) =< 6x, 2y >$.
$y = 6\alpha x$, $x = 2\alpha y$, $3x^2 + y^2 = 6$
1. ve 2. eşitlik birleştirilirse

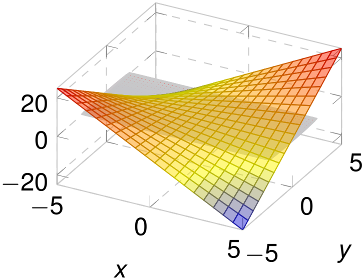
$$y = 6\alpha(2\alpha y) = 12\alpha^2 y \Rightarrow 12\alpha^2 = 1$$

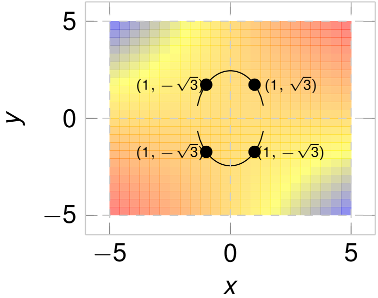
. Bu sonucu $3x^2 + y^2 = 6$ ifadesinde yerine koyarsak;

$$\begin{aligned} 6 &= 3x^2 + (6\alpha x)^2 \\ &= 3x^2 + 36\alpha^2 x^2 \\ &= 3x^2 + 3(12\alpha^2)x^2 \\ &= 3x^2 + 3x^2 \end{aligned}$$

sonuç: $x = \pm 1$ ve $y = \pm\sqrt{3}$: Kritik noktalar:
$(1, \sqrt{3}), (1, -\sqrt{3}), (-1, \sqrt{3}), (-1, -\sqrt{3})$,

Lagrange Fonksiyonu - Örnek II





SVM - Lagrange Fonksiyonu I

Destek Vektör Makinesi

$$\begin{aligned} & \min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \\ & s.t.\ y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b \geq 1, i = 1, \cdots, m \end{aligned} \tag{7}$$

Kısıt fonksiyonu yeni form

$$g_i(\mathbf{w}) = -y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) + 1 \leq 0$$

SVM - Lagrange Fonksiyonu II

SVM optimizasyon probleminin Lagrange fonksiyonu

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i \left[y^{(i)} \left(\mathbf{w}^T \mathbf{x}^{(i)} + b \right) - 1 \right] \quad (8)$$

$\mathcal{L}$ fonksiyonunun **w**'ye göre türevi

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \quad (9)$$

Bu durumda

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)} \quad (10)$$

$\mathcal{L}$ fonksiyonunun b'ye göre türevi

$$\frac{\partial}{\partial b} \mathcal{L}(\mathbf{w}, b, \alpha) = \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \quad (11)$$

Lagrange fonksiyonuna bu sonuçları birleştirirsek

SVM - Lagrange Fonksiyonu III

$$\begin{aligned}
\mathcal{L} &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \left(\mathbf{x}^{(i)}\right)^T \mathbf{x}^{(j)} - b \sum_{i=1}^{m} \alpha_i y^{(i)} \\
\mathcal{L} &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \left[y^{(i)} y^{(j)} \alpha_i \alpha_j \left(\left(\mathbf{x}^{(i)}\right)^T \mathbf{x}^{(j)} \right) \right]
\end{aligned} \tag{12}$$

Optimizasyon problemi

$$\begin{aligned}
\max_{\alpha} W(\alpha) &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \left[y^{(i)} y^{(j)} \alpha_i \alpha_j \left(\left(\mathbf{x}^{(i)}\right)^T \mathbf{x}^{(j)} \right) \right] \\
s.t.\ & \alpha_i \geq 0,\ i = 1, \cdots, m \\
& \sum_{i=1}^{m} \alpha_i y^{(i)} = 0
\end{aligned} \tag{13}$$

Python

Python

```
class sklearn.svm.SVC(C=1.0, kernel='rbf', degree=3, gamma=0.0,
coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200,
class_weight=None, verbose=False, max_iter=-1, random_state=None)
```

Lab

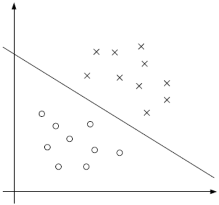
Lab - 1

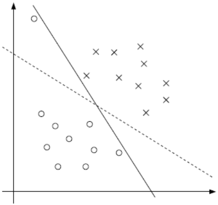
İçindekiler

Yumuşak Marjin Hiperdüzlem I
Soft Margin Hyperplane

Doğrusal Ayrıştırılmama Durumu

- Eğer veri kümemiz doğrusal ayrıştırılabilir değilse, *Hard Margin SVM* algoritması uygun olmayacaktır.
- Veya outliers olması durumunda hassasiyetin azaltılması gerekebilir.
- Bu durumu çözmek için, bir sınıfa ait bir örneği sınırın diğer tarafında yer almasına izin verilebilir.





Yumuşak Marjin Hiperdüzlem II

Soft Margin Hyperplane

Çözüm: Slack Variables

$$
\begin{aligned}
&\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m} \epsilon_i \\
&s.t.\ y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - \epsilon_i,\ i = 1, \cdots, m \\
&\epsilon_i \geq 0, i = 1, \cdots, m
\end{aligned}
\tag{14}
$$

Yeni Lagrange fonksiyonu

$$
\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m} \epsilon_i - \sum_{i=1}^{m} \alpha_i \left[y^{(i)}(\mathbf{x}^T\mathbf{w} + b) - 1 + \epsilon_i \right] - \sum_{i=1}^{m} r_i \epsilon_i \tag{15}
$$

α_i ve r_i ifadeleri Lagrange çarpanlarıdır(kısıtları ≥ 0)

Yumuşak Marjin Hiperdüzlem III

Soft Margin Hyperplane

Optimizasyon problemi

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \left[y^{(i)} y^{(j)} \alpha_i \alpha_j \left(\left(\mathbf{x}^{(i)} \right)^T \mathbf{x}^{(j)} \right) \right]$$

$$s.t.\ \alpha_i \geq 0 \geq C,\ i = 1, \cdots, m \tag{16}$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

Hard Margin vs Soft Margin

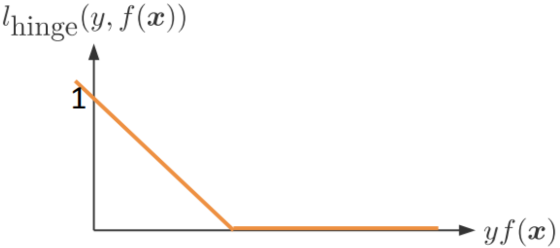
- Hard margin: $0 \leq \alpha_i$
- Soft margin: $0 \leq \alpha_i \leq C$
- C
  - Yüksek C: hata sayısı az, dar margin
  - Düşük C: hata sayısı çok, geniş margin

Hinge Loss

Hinge Loss

- Amaç: bütün örnekler için $y^{(i)} h(\mathbf{x}^{(i)}) \geq 1$
- Hinge loss fonksiyonu

$$\begin{aligned}\ell_{hinge}(u, y) &= max(1 - uy, 0) \\ &= \begin{cases} 0, & if \;\; yu \geq 1 \\ 1 - yu, & otherwise \end{cases}\end{aligned} \tag{17}$$



Soft Margin SVM

Soft Margin SVM

- Large margin ve hataların optimizasyonu.

$$\min_{\mathbf{w},b}\left(\frac{1}{2}||\mathbf{w}||^2 + C \times error(h)\right)$$

- error:

$$\begin{cases} 0, & if\ y(\mathbf{w}^T\mathbf{x}+b) \geq 1 \\ 1 - y(\mathbf{w}^T\mathbf{x}+b) & otherwise \end{cases}$$

- Soft-margin SVM

$$\underset{\mathbf{w},b}{argmin}\left(\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m} max\left(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}+b)\right)\right)$$

Lab - 2

Lab-2

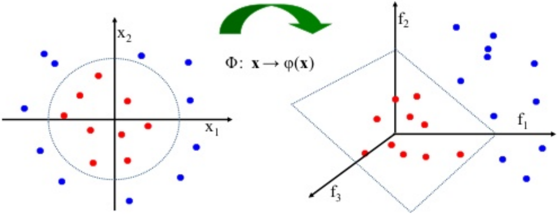
İçindekiler

Doğrusal Olmayan Karar Sınırı

Nonlinear decision boundary

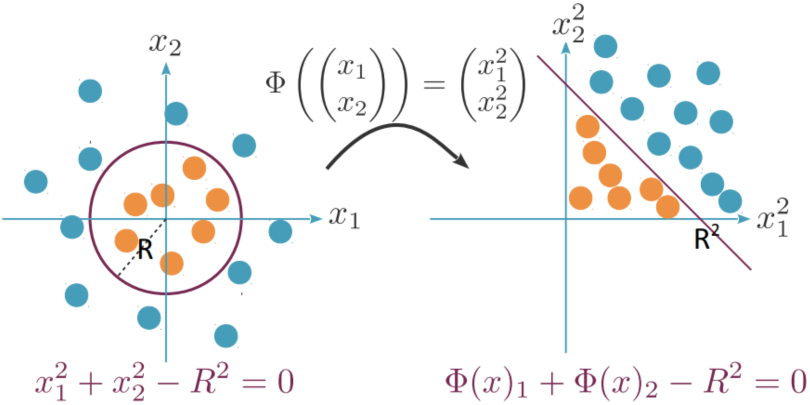
Doğrusal Olmayan Karar Sınırı

- Veri kümesi içinde yer alan örnekler, $\mathbf{x}_i$, daha yüksek boyutlu nitelik uzayına (Feature Space) haritalanarak doğrusal ayırştırılabilir hale getirilebilir.
- $\mathbf{x}_i$ örnekleri $\phi(\mathbf{x}_i)$ kullanılarak yüksek boyutlu uzay haritalaması (higher-dimensional space mapping).



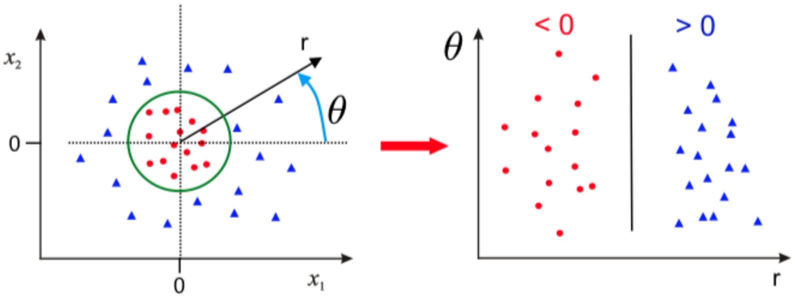
Bir özellik alanına doğrusal olmayan eşleme I

Non-linear mapping to a feature space



Bir özellik alanına doğrusal olmayan eşleme II

Non-linear mapping to a feature space

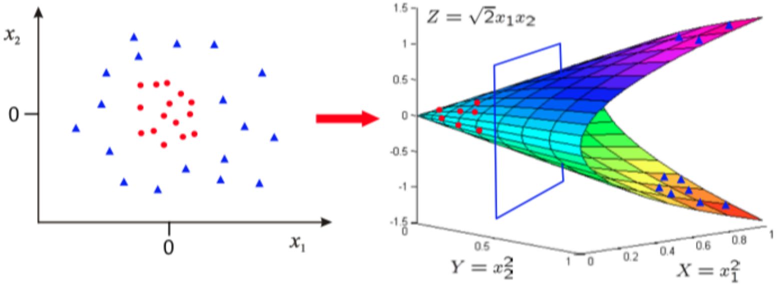


Şekil: polar coordinates

Bir özellik alanına doğrusal olmayan eşleme III

Non-linear mapping to a feature space

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



Çekirdek Fonksiyonları I

Kernel Functions

Çekirdek Hilesi

- Yüksek boyutlu veri kümelerinde $\phi(\mathbf{x}_i)^T \mathbf{x}_i)$ hesaplanması zorlu olabilir.
- Bunun yerine bir çekirdek fonksiyonu kullanılarak ϕ nitelik haritalaması tanımlanabilir.
$$K(x,z) = \phi(x)^T \phi(z)$$
- $K(x,z)$ hesaplaması $\phi(x)$ hesaplamasına göre daha kolay olabilir.

Karar Sınırı

$$\mathbf{w}^T \phi(\mathbf{x}) - b = 0 \tag{18}$$

Çekirdek Fonksiyonları II

Kernel Functions

SVM in the feature space

$$\max_{\alpha \geq 0} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \right)$$
$$\max_{\alpha \geq 0} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right) \tag{19}$$

Doğrusal Olmayan SVM Çekirdek Fonksiyonları

- Polynomial: $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$
- Radial basis function: $K(\mathbf{x}, \mathbf{y}) = exp^{\left(-\frac{||\mathbf{x}-\mathbf{y}||^2}{2\sigma^2} \right)}$
- Sigmoid: $K(\mathbf{x}, \mathbf{y}) = tanh(\mathbf{x} \cdot \mathbf{y} - s)$

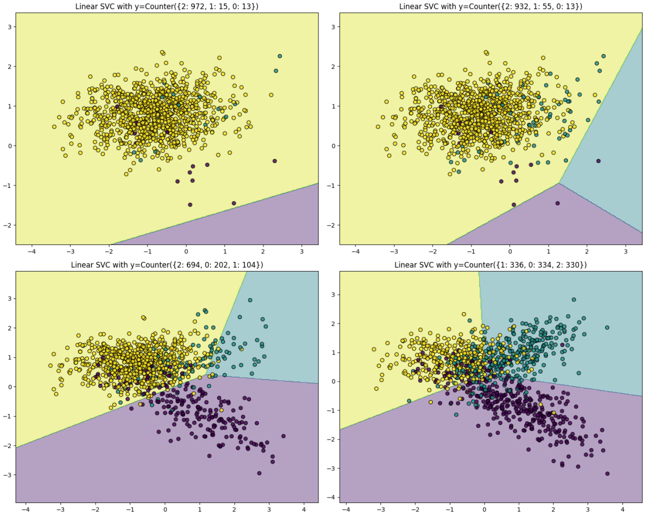
Lab

Lab-3

İçindekiler

Dengesiz Sınıf Dağılımı I

Class imbalance



Dengesiz Sınıf Dağılımı II

Class imbalance

Dengesiz Sınıf Dağılımı

- Dengesiz sınıflarla başa çıkmak başlı başına bir sorundur.
- Genel olarak iki çözüm yolu vardır:
  - **Undersampling**: örnek sayısını azaltmak için fazla temsil edilen sınıf (ları) örneklemek anlamına gelir.
    - Rassal örnek alma : bazı veri kümelerinde veri kayıbına neden olur
    - Sampling strategy : Diğer örneklere çok yakın olan örneklerin kaldırılması. Fazla örnek içeren sınıf için k-means clustering kullanılarak *high-density centroid* temizlenmesi
  - **Oversampling :** Azınlık sınıfları için sentetik veri örnekleri oluşturma
    - SMOTE
    - ADASYN

Dengesiz Sınıf Dağılımı III

Class imbalance

```
>>> print(pd.Series(train_Y).value_counts())
benign 67343
dos 45927
probe 11656
r2l 995
u2r 52
>>> from imblearn.over_sampling import SMOTE
>>> # Apply SMOTE oversampling to the training data
>>> sm = SMOTE(ratio='auto', random_state=0)
>>> train_x_sm, train_Y_sm = sm.fit_sample(train_x, train_Y)
>>> print(pd.Series(train_Y_sm).value_counts())
benign 67343
dos 67343
probe 67343
u2r 67343
r2l 67343
```

Örnek Oluşturma

Sample generation

Örnek Oluşturma

- Hem `SMOTE` hem de `ADASYN`, yeni örnekler üretmek için aynı algoritmayı kullanır.
- $\mathbf{x}_i$ şeklinde bir örnek olsun. Bu örnek kullanılarak oluşturulacak olan yeni örnek $\mathbf{x}_{new}$ ise *k neareast-neighbors* kullanılarak oluşturulacaktır.

$$\mathbf{x}_{new} = \mathbf{x}_i + \lambda \times (\mathbf{x}_{zi} - \mathbf{x}_i) \tag{20}$$

  λ is a random number in the range [0, 1]
- `ADASYN`: Not linearly correlated. Adds a little more variance