

Hafta 09 -Topluluk Yöntemleri - Boyut Azaltma - Anomali Tespiti

BGM 565 - Siber Güvenlik için Makine Öğrenme Yöntemleri Bilgi Güvenliği Mühendisliği Yüksek Lisans Programı

Dr. Ferhat Özgür Çatak
ozgur.catak@tubitak.gov.tr

İstanbul Şehir Üniversitesi
2018 - Bahar

İçindekiler

1 Topluluk Yöntemleri (Ensemble)

- Giriş
- Bagging meta-estimator
- Random Forest
- AdaBoost
- AdaBoost

2 Boyut Azaltma

- Temel Bileşen Analizi
- Doğrusal diskriminant analizi

3 Anomali Tespiti

- Gaussian Distribution
- One-class SVM

İçindekiler

1 Topluluk Yöntemleri (Ensemble)

- Giriş
- Bagging meta-estimator
- Random Forest
- AdaBoost
- AdaBoost

2 Boyut Azaltma

- Temel Bileşen Analizi
- Doğrusal diskriminant analizi

3 Anomali Tespiti

- Gaussian Distribution
- One-class SVM

Topluluk Yöntemleri

Ensemble Methods

Topluluk Yöntemleri

- Birden fazla modeli birleştirerek makine öğrenimi sonuçlarını iyileştirmektedir.
- Tek bir modele kıyasla daha iyi tahmin performansının üretilmesini sağlar.
- Topluluk yöntemleri, çeşitli makine öğrenme tekniklerini tek bir tahmin modelinde birleştiren **meta algoritmalar**dır.

Bagging Meta-Estimator I

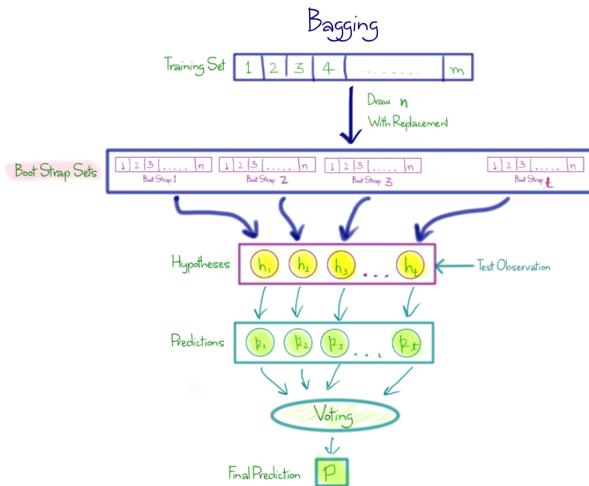
Bootstrap Aggregation

Bagging

- ▶ Orjinal veri kümesinden rassal örnekler alıp farklı sınıflandırıcılar oluşturulması.
- ▶ Temel bir sınıflandırma algoritması kullanılır.
- ▶ **Bootstrap**: Aynı örneklerin kullanılması
- ▶ **bootstrap_features**: Aynı niteliklerin kullanılması
- ▶ `sklearn.ensemble.BaggingClassifier`

Bagging Meta-Estimator II

Bootstrap Aggregation



Şekil: Bagging - <http://manish-m.com/?p=794>

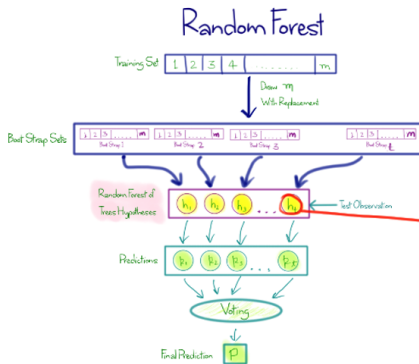
Lab-1

Random Forest I

Random Forest

- ▶ A random subset of the features.
- ▶ Farklı başlangıç düğümleri oluşturulmaktadır.
- ▶ *feature bagging*
- ▶ **sklearn.ensemble.RandomForestClassifier**
- ▶ Varsayılan nitelik sayısı: \sqrt{p}
- ▶ Diğerleri
 - ▶ log2: $\log_2(n_features)$
 - ▶ float: $\text{int}(max_features \times n_features)$
 - ▶ int
- ▶ **Bootstrap (default=True)**: Nitelikler tekrar kullanılacak mı?

Random Forest II



A Random Tree



Let's say that the Training Set had X features.
Then each node is split by choosing a feature out of a random sample of $x = \sqrt{\text{size}(X)}$ features.
The splitting feature is the one that gives the best Information Gain.
The tree is unpruned, and thus over fit.

Lab-2

AdaBoost I

Boosting

- ▶ Nitelik gösterimlerinin (feature representations) otomatik olarak seçimi
- ▶ Optimizasyon tabanlı yaklaşım
 - ▶ choose a representation
 - ▶ choose a loss
 - ▶ minimize the loss

Tanımlar

- ▶ **Zayıf öğrenme (weak learning) algoritması**
- ▶ **Güçlü sınıflandırıcı (Strong classifier)**
- ▶ **Boosting:** Zayıf sınıflandırıcıların ağırlıklı birleştirimi kullanılarak güçlü sınıflandırıcı hipotez oluşturulması

AdaBoost II

- ▶ Girdi veri kümesi : $\mathbf{x} \in \mathbb{R}^n$, etiketler $y \in \{-1, 1\}$
- ▶ *Feature functions*: $\phi_j : \mathbb{R}^n \rightarrow \{-1, 1\}$
- ▶ Ağırlık vektörü: $\theta = [\theta_1 \theta_2 \dots]$
- ▶ Güçlü sınıflandırıcı

$$h_{\theta}(\mathbf{x}) = \text{sign} \left(\sum_{j=1}^{\infty} \theta_j \phi_j(\mathbf{x}) \right) \quad (1)$$

AdaBoost I

Adaptive Boosting

Algorithm 1: AdaBoost

Data: Eğitim veri kümesi: $\mathbf{X} \in \mathbb{R}^{m \times n}$, etiket vektörü: $\mathbf{y} \in \{-1, 1\}$,
ensemble sayısı: T

Result: Güçlü sınıflandırıcı hipotez: h_θ

```

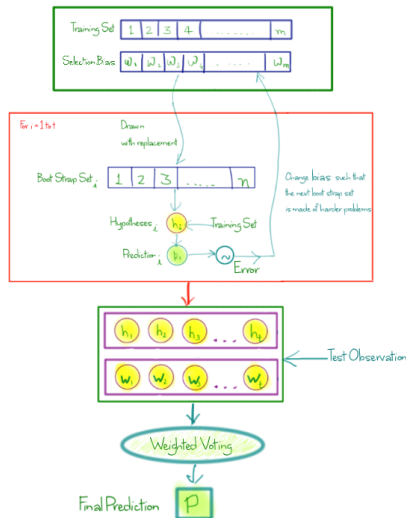
1  $\mathbf{w}^{(0)} \leftarrow \frac{1}{m}$  for  $i = 1, \dots, m$ ;
2 for  $i = 1 \dots T$  do
    /* Veri kümesi  $\mathbf{X}$  ile zayıf hipotez  $h_t$  oluştur.          */
3    $h_t \leftarrow \text{train\_model}(\mathbf{X})$ ;
    /* Hipotezin ağırlıklı hata toplamını hesapla          */
4    $\epsilon_t = \sum_{i=1}^n (w_i \times \mathbf{I}(h_t(\mathbf{x}) \neq y_i))$ ;
5    $\alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$ ;
    /* zayıf hipotezi  $\alpha$  ile Ensemble'a ekle                */
6    $F_t(x) = F_{t-1}(x) + \alpha F_t(x)$ ;
7    $\mathbf{w}^{(t+1)} = \frac{1}{\sum \mathbf{w}_t} \mathbf{w}_t e^{-y_i \alpha_t h_t(x_i)}$  // Ağırlıkları güncelle

```

AdaBoost II

Adaptive Boosting

Boosting



Lab-3

İçindekiler

- Giriş
- Bagging meta-estimator
- Random Forest
- AdaBoost
- AdaBoost

- Temel Bileşen Analizi
- Doğrusal diskriminant analizi

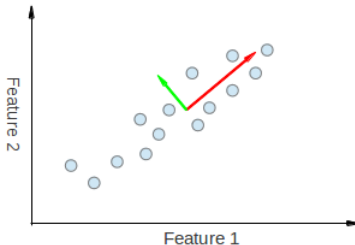
- Gaussian Distribution
- One-class SVM

Temel Bileşen Analizi I

Principal Component Analysis (PCA)

PCA

- ▶ Bir veri kümesinde bulunan temel bileşenlerin (principal components) bulunmasıdır.
- ▶ Temel bileşenler: verilerdeki temel yapıdır.
- ▶ Varyansın en çok olduğu yönleri bulmaktadır.
- ▶ PCA yeni boyutlar bulmaktadır.
 - ▶ Birbirinden bağımsız (linearly independent)
 - ▶ Dikgen (orthogonal)



PCA

- ▶ PCA: 2 adet *eigenvector* bulacaktır.
- ▶ Herbir *eigenvector* uzunluğu *eigenvalue* ile bulunur.

Doğrusal diskriminant analizi I

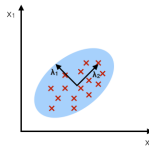
Linear Discriminant Analysis

LDA

- ▶ *PCA'e çok benzer bir algoritmadır.*
- ▶ *PCA sadece varyansı artıran component axes bulmaktadır.*
- ▶ *LDA ek olarak, sınıflar arasında uzaklığı maksimize etmeye çalışmaktadır.*

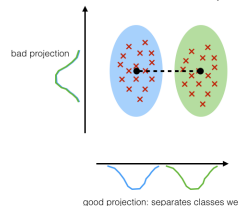
PCA:

component axes that maximize the variance



LDA:

maximizing the component axes for class-separation



Doğrusal diskriminant analizi II

Linear Discriminant Analysis

Adımlar

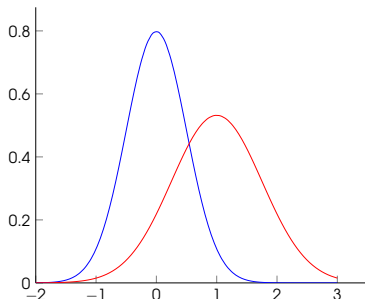
- ▶ Her bir sınıf için d boyutlu ortalama vektörünü hesapla
- ▶ Scatter matrix (in-between-class and within-class scatter matrix)
- ▶ Eigenvectors and eigenvalues of scatter matrix
- ▶ En yüksek k eigenvalue ait eigenvector kullanılarak girdi matrisi \mathcal{X} , k boyuta indirilir.

Lab-4

İçindekiler

- 1 Topluluk Yöntemleri (Ensemble)
 - Giriş
 - Bagging meta-estimator
 - Random Forest
 - AdaBoost
 - AdaBoost
- 2 Boyut Azaltma
 - Temel Bileşen Analizi
 - Doğrusal diskriminant analizi
- 3 Anomali Tespiti
 - Gaussian Distribution
 - One-class SVM

Gaussian Distribution I



```
from scipy.stats import norm
```

```
In [21]: norm.pdf(177,180,2)
```

```
Out[21]: 0.06475879783294587
```

```
In [22]: norm.pdf(178,180,2)
```

```
Out[22]: 0.12098536225957168
```

```
In [23]: norm.pdf(179,180,2)
```

```
Out[23]: 0.17603266338214976
```

```
In [24]: norm.pdf(180,180,2)
```

```
Out[24]: 0.19947114020071635
```

```
In [25]: norm.pdf(181,180,2)
```

```
Out[25]: 0.17603266338214976
```

```
In [26]: norm.pdf(182,180,2)
```

```
Out[26]: 0.12098536225957168
```

```
In [27]: norm.pdf(183,180,2)
```

```
Out[27]: 0.06475879783294587
```

Gaussian Distribution II

Gaussian Distribution

- Bir fonksiyon tarafından tanımlanabilen tanıdık bir çan şeklinde eğridir. $\mathcal{N}(\mu, \sigma^2)$
- Gauss dağılımı ortalama ve varyans ile ifade edilmektedir.
- μ : ortalama (eğrinin ortası)
- σ : standart sapma (eğrinin genişliği)

$$p(x; \mu, \sigma^2) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2} \quad (2)$$

$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)} \quad (3)$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^m \left(x^{(i)} - \mu \right)^2 \quad (4)$$

Gaussian Distribution III

Algoritma

- ▶ $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ where $\mathbf{x} \in \mathbb{R}^n$ and $\mathcal{D} \in \mathbb{R}^{m \times n}$
- ▶ **Independence Assumption:**

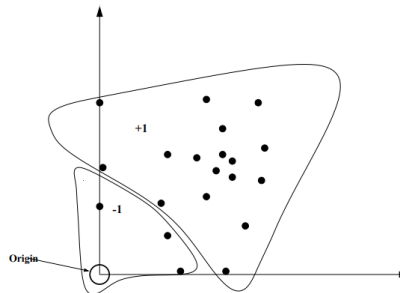
$$\begin{aligned} p(x) &= p(x_1; \mu_1, \sigma_1^2) p(x_2; \mu_2, \sigma_2^2) \cdots p(x_n; \mu_n, \sigma_n^2) \\ &= \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) \end{aligned} \quad (5)$$

Lab-5

One-class SVM

One-class SVM

- ▶ Unsupervised
- ▶ SVM karar sınırını maksimize etmeye çalışmaktadır.
- ▶ Bütün örnekler +1 orjin uzakta bir nokta ise -1 olacak şekilde yapılmaktadır.
- ▶ **sklearn.svm.OneClassSVM**



Lab-6