



T.C
KARABÜK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Buse Yaren TEKİN

1928126503

Dr. Öğr. Üyesi Caner ÖZCAN

BSM754 – Uzaktan Algılama ve Uygulamaları Dersi Raporu

1. Oluşturulan Özellik Vektörünün İncelenmesi

Kod içerisinde oluşturulan özellik vektörünün incelenebilmesi için Excel ortamına veriler şekildeki kod ile aktarılmalıdır.

```
xlswrite("FeatureVector.xls",feature_vektor);
```

Şekil 1. Verilerin Excel Dosyası Olarak Kaydedilmesi

Elde edilen özellik veriler tablo haline getirilerek incelemeye alındığında bu değerler arasında aşırı farklar bulunmuştur.

Sütun3	Sütun4	Sütun5	Sütun6	Sütun7	Sütun8	Sütun9	Sütun10	Sütun11	Sütun12	Sütun13	Sütun14	Sütun15	Sütun16	Sütun17
22,32499	0	0	0,087549	8,700476	1,62E-15	9,86E-17	26,966359	26,966359	26,966359	0	0	0,1057506	11,618892	7,045E-15

Şekil 2. Özellik Vektörü Değerlerinin İncelenmesi

Bu değerler feature_vektor değişkeni içerisinde sırasıyla v1, v2 ve GLCM matrisinin özelliklerinden gelmektedir.

```
%kontrast-korelasyon-enerji-homojenlik  
feature_vektor=[v1 v2 statsg.Contrast statsg.Correlation statsg.Energy statsg.Homogeneity];
```

Şekil 3. Özellik Vektörünün İçerik Değerleri

RGB, HSV, CIE renk kanallarından elde edilen toplamda 18 adet değer Excel dosyasına kaydedildi ve ardından değerler kontrol edildi.

Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6	Sütun7	Sütun8	Sütun9	Sütun10
22,32499	22,32499	22,32499	0	0	0,087549	8,700476	1,62E-15	9,86E-17	26,966359

Şekil 4. V1 Değişken Tablosu

Geriye kalan 8 adet satır ise GLCM matrisi ile elde edilen özellik değerleridir.



Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6
22,32499	26,96636	727,1845	39,89798	NAN	1225,59

Şekil 5. V2 Değişken Tablosu

Bu değerlerin farklılıklarının indirgenebilmesi normalizasyon işlemi gerçekleştirilmelidir. Bu sebeple MATLAB’da normalizasyon işlemi aşağıda verilen şekildeki gibi gerçekleştirilmektedir. Aşağıdaki satırda ortalama 0 ve standart sapma 1 olacak şekilde veriler normalize edilmektedir.

```
%Normalizasyon işlemi , (-1)-1 aralığına normalleştirme  
normv1=normalize(v1);
```

Şekil 6. Normalizasyon MATLAB Kodu


$$X_{\text{new}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$


Normalizasyon işlemi yapılmadan önce ortalama değerlerde olan tutarsızlık normalleştirme işlemi sonucunda değer değişimleri azaltılmıştır.

Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6	Sütun7	Sütun8	Sütun9
22,32499	22,32499	22,32499	0	0	0,087549	8,700476	1,62E-15	9,86E-17

Şekil 7. Normalize Edilmeden Önceki Ortalama Veriler

Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6	Sütun7	Sütun8	Sütun9
1,112121	1,112121	1,112121	-0,80212	-0,80212	-0,79462	-0,05611	-0,80212	-0,80212

Şekil 8. Normalize Edilmiş Ortalama Verileri

Sütun10	Sütun11	Sütun12	Sütun13	Sütun14	Sütun15	Sütun16	Sütun17	Sütun18
26,966359	26,966359	26,966359	0	0	0,1057506	11,618892	7,045E-15	1,141E-15

Şekil 9. Normalize Edilmemiş Standart Sapma Verileri

Sütun10	Sütun11	Sütun12	Sütun13	Sütun14	Sütun15	Sütun16	Sütun17	Sütun18
1,5100929	1,5100929	1,5100929	-0,802124	-0,802124	-0,793057	0,1941319	-0,802124	-0,802124

Şekil 10. Normalize Edilmiş Standart Sapma Verileri

Hesaplanan GLCM matrisindeki özellikler baz alındığında elde edilen V2 değişkeni içerisindeki değerler de tıpkı V1 değişkeni gibi mean 0 ve standart sapma 1 olacak şekilde normalize edilerek değerlerin farklılaştırılması engellenmiştir.

Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6
22,32499	26,96636	727,1845	39,89798	NAN	1225,59

Şekil 11. Normalize Edilmemiş Ortalama - Standart Sapma Değerleri

Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6
-0,70493	-0,69646	0,582089	-0,67284	NAN	1,492139

Şekil 12. Normalize Edilmiş Ortalama - Standart Sapma Değerleri

Sütun 5' de görülen NAN (Not A Number) ibaresi GLCM için hesaplanan M adet entropi değerinden gelmektedir. Özellik vektörü oluşturulurken değer olarak gösterilen v1 ve v2 değerleri yerine normalleştirilmiş normv1 ve normv2 değerleri gönderilmiştir.

Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6	Sütun7	Sütun8	Sütun9	Sütun10	Sütun11	Sütun12	Sütun13
22,32499	22,32499	22,32499	0	0	0,087549	8,700476	1,62E-15	9,86E-17	26,966359	26,966359	26,966359	0

Şekil 13. Normalize Edilmemiş Feature Vektör Değerleri

Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6	Sütun7	Sütun8	Sütun9	Sütun10	Sütun11	Sütun12	Sütun13
1,112121	1,112121	1,112121	-0,80212	-0,80212	-0,79462	-0,05611	-0,80212	-0,80212	1,5100929	1,5100929	1,5100929	-0,802124

Şekil 14. Normalize Edilmiş Feature Vektör Değerleri

Normalizasyon yapıldıktan sonra veriler incelendiğinde içerisinde negatif değerler olduğu da görülmüştür. Bu sebeple MATLAB özel komutlarından olan **normalize(value, 'range')** komutu ile bu değerler 0-1 aralığına normalleştirilmiştir.

```
% 0-1 aralığına normalleştirme
nrmrange=normalize(v1,'range');
```

Şekil 15. V1 Değişkeni 0 - 1 Aralığına Normalleştirme

1x18 double	1	2	3	4	5	6	7	8	9	10	11
1	0.8279	0.8279	0.8279	0	0	0.0032	0.3226	5.9998e-17	3.6576e-18	1	1

Şekil 16. Normalize Edilmiş V1 Değişkeni

```
% 0 - 1 aralığına normalleştirme
nrmrangev2=normalize(v2,'range');
```

Şekil 136. V2 Değişkeni 0 - 1 Aralığına Normalleştirme

Ardından özellik vektörüne normalize edilmiş bu değerler verildiğinde sayısal değerler arasındaki değişimin normalleştiği fark edilmiştir.

A	B	C	D	E	F	G	H	I	J	K
Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6	Sütun7	Sütun8	Sütun9	Sütun10	Sütun11
0,827883	0,827883	0,827883	0	0	0,003247	0,322642	6,00E-17	3,66E-18	1	1

Şekil 17. Normalleştirilmiş Feature Vektör

Şekil 18' de görüldüğü üzere **mentropy** değeri görüntü değerlerinin logaritmik olarak hesaplanıp görüntüde bulunan satır x sütun değerlerine bölünerek elde edilmektedir.

Entropy

$$\frac{\sum_{x=1}^M \sum_{y=1}^N I(x, y) (-\log_2 I(x, y))}{M \times N}$$

Şekil 148. M Adet Verinin Entropi Değerinin Formülleştirilmesi

Şekil 15' de görülen formül aşağıdaki satırda çalıştırıldığında etropi değeri NAN gelmektedir. Logaritmik değer olan bölüm değişken olarak tutulduğunda normal bir değer olduğu görülmektedir.

```
logdeger=(gImage(row,col)*(-(log2(gImage(row,col)))));
ent= ent + (gImage(row,col)*(-(log2(gImage(row,col)))));
```

Şekil 19. Logaritmik Değer ve Entropi Hesaplanması

logdeger	×
1x1 double	
1	
1	-560.5361

Şekil 20. Logaritmik Değer Çıktısı

Formülde yer alan görüntünün satır ve sütun piksel değerleri değişken olarak Şekil 18’ de tutulduğu gibi saklanabilmektedir.

```
gimagedeger=double (gImage (row,col) );
```

Şekil 21. Görüntü Piksel Değerlerinin Değişkende Tutulması

GLCM matrisinde kullanılacak olan entropi fizikte olduğu gibi aslında termal enerjiyi temsil etmektedir. Matematiksel açıdan değişkende tutulabilmesi mümkün kılınmamaktadır.

2. Makine Öğrenimi Yöntemleri

Sınıflandırma, belirli özellikteki nesnelerin önceden belirlenmiş alt gruplara atanması işlemidir.

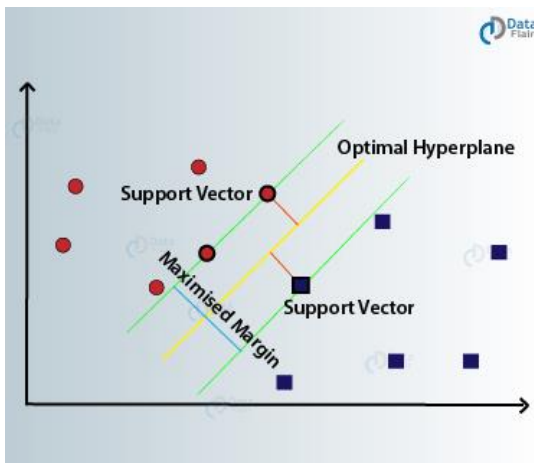
2.1. Destek Vektör Makineleri (SVM)

Değişkenler arasındaki örüntülerin bilinmediği veri setlerindeki sınıflandırma problemleri için önerilmiş bir makine öğrenimi yöntemidir.

- Uygulanması kolay
- İyi genelleştirme performansı
- Sınıflandırma konusunda oldukça etkili

Nasıl Çalışır ?

Sınıflandırma için düzlemde bulunan 2 grup arasında sınır çizilerek 2 gruba ayırım sağlanmaktadır. Eğitim setlerinde girdi ve çıktılar bulunur. Test ve yeni veri girdilerinde ise değişkenleri sınıflandıracak **karar fonksiyonları** elde edilir.



Şekil 23. SVM Karar Fonksiyonu ve Eğrisi Çizimi

gimagedeger		
1x1 double		
	1	2
1		87

Şekil 22. Görüntü Piksel Değer Çıktısı

Name,Region,X,Y,Width,Height	
11_1_dilaraacar.bmp,1,310,59,28,33	
11_1_dilaraacar.bmp,2,306,93,38,47	
11_1_dilaraacar.bmp,3,311,187,38,41	

Şekil 24. Ana Projede Bulunan Etiketler, X, Y, Width, Height

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
```

Şekil 25. Python SVM ve Veri Seti Hazırlığı

2.2. Naive Bayes Sınıflandırıcı

Naive Bayes Sınıflandırması makine öğreniminde denetimli öğrenme alt sınıfındadır. Daha açık bir ifadeyle sınıflandırılması gereken sınıflar (kümeler) ve örnek verilerin hangi sınıflara ait olduğu bellidir.

E-posta kutusuna gelen e-postaların spam olarak ayrıştırılması işlemi buna örnek verilebilir. Bu örnekte spam e-posta ve spam olmayan e-posta ayrıştırılacak iki sınıfı temsil eder. Veriler belli olmadığı için Naive Bayes Sınıflandırma Algoritması bu proje için uygun

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

$P(A|B)$; B olayı gerçekleştiği durumda A olayının meydana gelme olasılığıdır (bakınız **koşullu olasılık**)

$P(B|A)$; A olayı gerçekleştiği durumda B olayının meydana gelme olasılığıdır

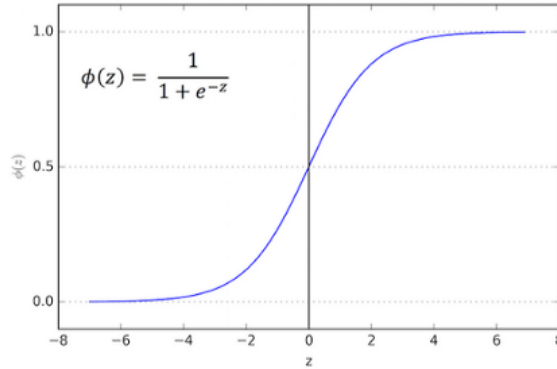
$P(A)$ ve $P(B)$; A ve B olaylarının **önsel olasılıklarıdır**.

bulunmamaktadır.

Şekil 26. Olasılık Temelli Çalışma

2.3. Lojistik Regresyon

Lojistik Regresyon, binary sınıflandırma problemini çözmeye yarayan bir sınıflandırıcıdır. İkili durum söz konusu olan modellerde sonuç genellikle 0 veya 1 olarak tanımlanmaktadır.



Şekil 27. Logaritmik Fonksiyon Eğrisi - Sigmoid

Bu projede True – False değerleri beklenmediği için Lojistik Regresyon Algoritması uygun bulunmamaktadır.

2.4. Desicion Tree (Karar Ağaçları)

Bir karar ağacı, çok sayıda kayıt içeren bir veri kümesini, bir dizi karar kuralları uygulayarak daha küçük kümelerle bölmek için kullanılan bir yapıdır. Yani basit karar verme adımları uygulanarak, büyük miktarlardaki kayıtları, çok küçük kayıt gruplarına bölerek kullanılan bir yapıdır.

- Sayısal ve kategorik değerler işlenmektedir.
- Kayıp değerler desteklenmemektedir.
- Çok çıktılı problemleri ele alabilmektedir.

- Anlama ve yorumlaması kolaydır.
- Overfitting yaşanabilmesi mümkündür. (Budama, vs. çözümler)

En çok kullanılan karar ağacı algoritmaları:

- ID3
- C4.5
- CHAID
- CART

2.5. Random Forest (Rastgele Orman)

Rastgele ormanlar veya rastgele karar ormanları, sınıflandırma, regresyon ve diğer görevler için, eğitim aşamasında çok sayıda karar ağacı oluşturarak problemin tipine göre sınıf (sınıflandırma) veya sayı (regresyon) tahmini yapan bir toplu öğrenme yöntemidir. Rastgele karar ormanları, karar ağaçlarının eğitim setlerine aşırı uyma problemlerini gidermektedir.



Karar ağaçlarının en büyük problemlerinden biri aşırı öğrenme-veriyi ezberlemedir. Rassal orman modeli bu problemi çözmek için hem veri setinden hem de öznitelik setinden rassal olarak 10'larca 100'lerce farklı alt-setler seçiyor ve bunları eğitiyor. Bu yöntemle 100'lerce karar ağacı oluşturuluyor ve her bir karar ağacı bireysel olarak tahminde bulunuyor.

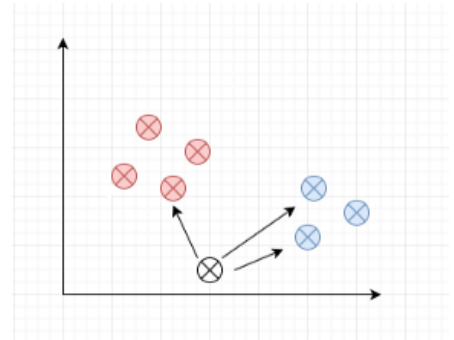
- ✓ Rassal orman modelinde farklı veri setleri üzerinde eğitim gerçekleştiği için varyans, diğer bir deyişle karar ağaçlarının en büyük problemlerinden olan aşırı öğrenme azalır.
- ✓ Kategorik ve sayısal yapılarda çalışabilmektedir.

2.6. K-En Yakın Komşu Algoritması

k-en yakın komşuluk (KNN) algoritması, uygulaması kolay gözetimli öğrenme algoritmalarındandır. Hem sınıflandırma hem de regresyon problemlerinin çözümünde kullanılmaktadır.

Örnek veri setine katılacak olan yeni verinin, mevcut verilere göre uzaklığı hesaplanıp k sayıda yakın komşuluğuna bakılır. Uzaklık hesapları için genelde 3 tip uzaklık fonksiyonu kullanılmaktadır;

- 1) Euclidean (Öklid) Uzaklığı
- 2) Manhattan Uzaklığı
- 3) Minkowski Uzaklığı



ADIMLAR

- ✓ İlk olarak k parametresi belirlenir. Bu parametre verilen bir noktaya en yakın komşuların sayısıdır. Örneğin: k=2 olsun. Bu durumda en yakın 2 komşuya göre sınıflandırma yapılacaktır.
- ✓ Örnek veri setine katılacak olan yeni verinin, mevcut verilere göre uzaklığı tek tek hesaplanır. İlgili uzaklık fonksiyonları yardımıyla.
- ✓ İlgili uzaklıklardan en yakın k komşu ele alınır. Öznitelik değerlerine göre k komşu veya komşuların sınıfına atanır.
- ✓ Seçilen sınıf, tahmin edilmesi beklenen gözlem değerinin sınıfı olarak kabul edilir. Yani yeni veri etiketlenmiş (label) olur.

```
# Import Modules
from sklearn import datasets
from sklearn import metrics
from sklearn.neighbors import KNeighborsClassifier

# load iris the datasets
dataset = datasets.load_iris()

# fit a k-nearest neighbour model to the data
model = KNeighborsClassifier()
model.fit(dataset.data, dataset.target)

KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=1, n_neighbors=5, p=2,
                     weights='uniform')
```

Şekil 28. Örnek Python Kodu

```
{'data': array([[5.1, 3.5, 1.4, 0.2],
                [4.9, 3. , 1.4, 0.2],
                [4.7, 3.2, 1.3, 0.2],
                [4.6, 3.1, 1.5, 0.2],
```

Şekil 29. Kod Çıktısı (Yüklenen Sayısal Değerler)