Hiyerarşik Kümeleme

Eyüp Kaan Ülgen 10 Ocak 2021 Genel bir bakış, Hiyerarşik Kümeleme, Kümeleme, Makine Öğrenmesi, Python, Teknik, Uygulama, Veri Bilimi, Veri Görselleştirme 0 15593 2145 kelime - 14 dakika 66

Merhabalar! Uzun bir aradan sonra yine sizlerleyim 🙂 Bu yazımda **denetimsiz öğrenme (unsupervised learning)** algoritmalarından **hiyerarşik kümelemeyi** ele alacağız.

İlk olarak **denetimsiz öğrenme** nedir?

Denetimsiz Öğrenme

Denetimli öğrenmede (supervised learning) bilindiği üzere etiket bilgisine sahip verilerle çalışıyoruz. Buna karşılık denetimsiz öğrenmede (Gözetimsiz Öğrenme / Kümeleme) elimizde sadece girdi verileri bulunuyor. Herhangi bir sınıf (etiket) bilgisine sahip değiliz. Denetimsiz öğrenmede amaç, girdi verisindeki dağılımları veya düzenlilikleri bulmak. Bunun doğal bir sonucu olarak verilerde belirli bir örüntü var mı sorusuna cevap arıyoruz.

Girdi uzayının bir yapısı mevcut, bazı örneklere daha sık rastalanır; istediğimiz neyin sık, neyin seyrek gerçekleştiğini belirlemek. Bu durumda literatürde “dağılım kestirimi (density estimation)” olarak da bilinmektedir [10].

Örneğin, bir şirketin müşteri profillerini ele alalım. Bu şirket müşterilerinin yaptıkları alışverişlerinin yanında onların adres bilgilerini de saklıyor olsun. Bu şekilde müşteri profilleri oluşturulup, ne tür müşterilerin hangi yoğunlukta olduğu kolayca takip edilebilir. Bu durumda bir denetimsiz öğrenme algoritması, birbirine benzeyen müşterileri veya ayrık olanları tespit edebilir. Bu profillere uygun olarak indirimler yapılabilir.

Özetle, denetimsiz öğrenme kullanılarak, kümeleme, olasılık yoğunluk tahmini, öznitelikler arasındaki ilişkilerin bulunması veya boyut indirgeme işlemleri gerçekleştirilebilir.

Denetimsiz öğrenme yöntemleri temel olarak 3 gruba ayrılıyor diyebiliriz [4]:

- Bölümlemeli Yöntemler (k-means)
- Hiyerarşik Yöntemler (Agglomerative, Divisive)
- Yoğunluk Tabanlı Yöntemler (GMM, DBSCAN, OPTICS)

Hiyerarşik Kümeleme

Bu yazımda hiyerarşik kümeleyi inceleyeceğiz. Bilindiği üzere k-merkezli kümeleme yönteminin bir dezavantajı vardır. Küme sayısını önceden belirlemeniz gerekmektedir. Bu dezavantajı ortadan kaldırmak için hiyerarşik kümeleme

geliştirilmiştir. Hiyerarşik kümeleme algoritmasının temel mantığı, benzer özniteliklerin bir araya gelmesine veya tam tersine bölünmesine dayanmaktadır. Bu çalışma mantığına göre **birleştirici (agglomerative)** ve **bölücü (divisive)** olmak üzere iki temel yaklaşım mevcuttur. **Tüme varım (bottom up)** olarak da bilinen birleşitirici yaklaşımda, başlangıçta tüm nesneler birbirlerinden ayrıdır. Yani eldeki verinin herbiri ayrı bir küme olarak kabul edilirek işe başlanır. Ardından benzer özniteliklere sahip kümeler bir araya gelerek tek bir küme elde edilmeye çalışılır. **Tümden gelim (top bottom)** yaklaşımda ise tüme varım metodunun aksine ayrıştırıcı bir strateji hakimdir. Bu yaklaşımda başlangıçta tek bir küme vardır. Her aşamada uzaklık/benzerlik matrisine göre nesneler ana kümeden ayrılarak, farklı alt kümeler oluşur. Süreç sonucunda her veri bir küme olur.

Hiyerarşik kümeleme analizide, veriler arasındaki benzerlik ve uzaklık hesaplamaları her adımda güncellenmektedir. Hesaplanan uzaklık/ benzerlik değerlerinden oluşan matris, seçilen **bağlantı yöntemi**nin kullanılmasına temel teşkil etmektedir. Aşağıda sıklıkla kullanılan bağlantı yöntemlerinden bahsedeceğiz.

Birleştirici (agglomerative) hiyerarşik küme algoritmasını ele alırsak: Bu yöntemde her birim başlangıçta ayrı bir küme olarak kabul edilir ve benzer birimler bir araya getirilerek n birim aşamalı olarak sırasıyla n, n-1, n-2, n-r kümeye yerleştirilir.

Algoritmanın temel çalışma yapısı [5]:

1. n tane birey n tane küme olmak üzere işlemlere başlanır.
2. En yakın iki küme (dij değeri en küçük olan) birleştirilir.
3. Küme sayısı bir indirgenerek yinelenmiş uzaklıklar matrisi bulunur.
4. 2. ve 3. adımlar (n-1) kez tekrarlanır.

Sıklıkla kullanılan bağlantı (linkage) yöntemleri şöyledir [6]:

- **Bağlantı Temelli Teknikler:**
  - Tek Bağlantı (Single Linkage) / En Yakın Komşu Yöntemi (Nearest Neighbor)
  - Tam Bağlantı (Complete Linkage) / En Uzak Komşu Yöntemi (Furthest Neighbor)
  - Ortlama Bağlantı (Average Linkage)
- **Varyans Temelli Teknikler**

    - Ward Yöntemi (Ward's Linkage)
- **Merkezileştirme Temelli Teknikler**
    - Medyan Bağlantı (Median Linkage)
    - Merkezi Yöntem (Centroid Linkage)

Tek Bağlantı (Single Linkage)

Uzaklık matrisinden yararlanılarak birbirine en yakın iki yapı veya küme birleştirilmektedir. Bu yöntemin dezavantajı, işlemlerin uzun sürmesidir.

Tam Bağlantı (Complete Linkage)

Bu yöntemde ilgili yapılar arasındaki en büyük uzaklık dikkate alınarak birleştirme işlemi gerçekleşmektedir.

Bu yöntemin dezavantajı, veri setindeki uç noktalara karşı duyarlı olmasıdır.

Ortalama Bağlantı (Average Linkage)

İki yapı içerisindeki verilerin birbirleri arasındaki uzaklıkların ortalama değerini dikkate alarak gerçekleşen birleşme işlemidir. Bu yöntem, Tek ve Tam Bağlantı arasında uygun bir tercihtir.

Ward Bağlantı (Ward's Linkage)

Ward (1963), her gruplamayla ilişkili bilgi kaybını en aza indirecek ve bu kaybı kolayca yorumlanabilir biçimde ölçecek bir yöntem önermiştir. Bilgi kaybı, Ward tarafından hata kareler toplamı olarak tanımlanır. Bu yöntem, en küçük mesafeli grupları bir araya getirmez, ancak belirli bir heterojenite ölçüsünü çok fazla artırmayan yapıları birleştirir (minimum varyans değerine sahip olanları). Ward yönteminin amacı, yapıları, bu yapılar içindeki çeşitliliğin çok fazla artmaması için birleştirmektir. Bu, mümkün olduğunca homojen kümeler halinde yapılar oluşturur.

Merkezi Bağlantı (Centroid Linkage)

Bu yöntemde iki yapı (küme) arasındaki uzaklık ölçüsü olarak Kareli Euclid (Squared Euclidean) uzaklığı kullanılmaktadır. Her küme, o andaki kümenin ağırlık noktası ile temsil edilir. İki küme birleştiğinde, ağırlık noktalarının birbirlerinden minimal uzaklıkta olması yeterlidir. Bu yöntemin en önemli avantajı farklı nitelikteki gözlemlerden çok fazla etkilenmemesidir.

Medyan Bağlantı (Median Linkage)

Birleştirilecek iki yapının boyutları çok farklıysa, yeni yapının ağırlık merkezi daha büyük olan yapının ağırlık merkezine çok yakın olacaktır. Centroid yönteminin bu dezavantajından dolayı 1967 yılında Gower, medyan yöntemini önermiştir. Bu yöntem hem benzerlik hem de uzaklık yaklaşımları için uygun hale getirilebilir.

Hiyerarşik kümeleme analizinde kullanılan en etkin görselleştirme aracı **dendrogram**'lardır. Dendrogram, hiyerarşik kümeleme yöntemiyle elde edilen sonuçların kolaylıkla anlaşılmasını sağlamaktadır.

Uygulama

Bu yazı kapsamında birleştirici (agglomerative) hiyerarşik küme algoritmasını kullanarak basit bir uygulama yapacağız. Uygulama kapsamında kullanılan veri setine şuradan ulaşabilirsiniz [7]. Veri seti 200 satır, 5 kolondan oluşuyor.

İlk olarak uygulama için kullanacağımız python kütüphanelerini çağırdık. Numpy, pandas ve matplotlib veri bilimi uygulamalarında temel oluşturan 3 kütüphane. Bunun yanında birleştirici (agglomerative) hiyerarşik kümele analizi için sklearn kütüphanesinden, Dendrogram için ise SciPy kütüphanesinden yararlanışmıştır.

```
1.  import numpy as np
2.  import pandas as pd
3.  import matplotlib.pyplot as plt
4.  from scipy.cluster.hierarchy import dendrogram, linkage
5.  from sklearn.cluster import AgglomerativeClustering
```

Kütüphaneleri çağırdıktan sonra veri setini okutma süreci geliyor. csv (comma-separated values) formatında olan veri setimiz pandas kütüphanesi kullanarak okutulmuştur.

```
1.  musteri = pd.read_csv('./shopping_data.csv')
2.  musteri.head()
```

Şekil-1: Veri seti içerisindeki ilk 5 satırın gösterimi

Veri seti içerisinde boş veri olup olmadığını, satır ve sütun sayılarıyla ilgili bilgileri veya ilgili sütunların veri tipini öğrenmek için pandas kütüphanesinde mevcut olan info() özelliği kullanıldı.

```
musteri.info()
```

Sırada kümele analizi için kullanılacak verileri seçmeye geldi. Bu uygulama kapsamında "senelik gelir" (annual income) ve "harcama puanı" (spending score) kolonları dikkate alındı.

```
veri = musteri[["AnnualIncome","SpendingScore"]]
```

Yukarıda da belirtildiği üzere kümele analizi için iki kolon seçildi. Bu kolonların nasıl dağıldıklarını görmek için matplotlib kütüphanesi kullanılarak saçılım (scatter) grafiği üretildi.

```
plt.figure(figsize=(10, 7))
plt.scatter(veri.AnnualIncome, veri.SpendingScore)
plt.xlabel("Senelik Gelir", fontsize=20)
plt.ylabel("Harcama Puanı", fontsize=20)
plt.show()
```

Şekil-2: Senelik Gelir – Harcama Puan grafiği

İlgili grafiği incelediğimizde kabaca 5 farklı bir öbeğin varlığı görülmekte. İlgili veri setinde kabaca kaç küme olduğunu dendrogram kullanmadan direk saçılım grafiği üzerinden belirleyebildik. Gerçek problemlerle uğraşırken her zaman bu kadar şanslı olmuyoruz.

Hiyerarşik kümeleme analizlerinde küme sayısını belirlemek veya elde edilen sonuçların kolaylıkla yorumlanması için dendrogramlar kullanılmaktadır. İlgili veri seti için dendrogram üretirken yukarıda belirtilen bağlantı (linkage) yöntemlerinden **ward** kullanılmıştır. Bu işlem için SciPy kütüphanesinden yararlanıldı.

```
labels = range(1, 201) # Birleştirici hiyerarşik kümele analizinde başlaşg
# kabul ediliyor. Bu sebeple 200 verinin takibini yapmak için etiketleme (

linked = linkage(veri, 'ward') #Kümele işlemi için kullanılan bağlantı met
# minimizasyonuna dayanan bir yöntem.

plt.figure(figsize=(20, 7))
dendrogram(linked,
            orientation='top',
            labels=labels,
```

```
                show_leaf_counts=True)
    plt.axhline(200, ls="--", c="k")
    plt.show()
```

Şekil-3: Dendrogram gösterimi

Üretilen dendrogramın y-ekseninde kümeler arasındaki uzaklık görülmektedir. x-ekseninde ise kümeleri oluşuran veri noktalarının ID (label) numaraları bulunmaktadır. y-ekseni incelendiğinde 200 değerinde kesikli çizgi mevcuttur. Nihai kümelerimiz arasındaki mesafenin en az 200 olmasını istediğimiz için alt sınır (threshold) uygulandı. Görüldüğü üzere bu alt sınır dikey 5 sütunu kesmektedir. Bu aslında veri kümemizi temsil eden 5 küme olduğunu söylemektedir. Hem saçılım grafiğindeki dağılım hem de dendrogram dikkate alındığında modelimiz 5 küme şartı altında üretildi.

Modelimizi üretmek için sklearn kütüphanesi kullanıldı. Ayrıca bağlantı yöntemi olarak ward tercih edildi. Fark ettiğiniz üzere dendrogram içinde benzer yöntem kullanılmıştı.

```
kume = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage
kume.fit(veri)
```

Şekil-2'de herhangi bir renk kodu kullanmadan sadece fiziksel uzayda 5 farklı öbeğin dağılımını görmüştük. Şimdi ise saçılım grafiği üzerinde modelimizin sonucunu renk kodu olarak ele alacağız. Bunun için kume.labels_ ifadesi kullanılmıştır.

```
plt.figure(figsize=(10, 7))
plt.scatter(veri.AnnualIncome, veri.SpendingScore, c=cluster.labels_, cmap
plt.xlabel("Senelik Gelir", fontsize=20)
plt.ylabel("Harcama Puanı", fontsize=20)
plt.show()
```

Şekil-4: Kümele işlemi sonucu elde edilen saçılım (scatter) grafiği.

Her küme farklı bir renkle temsil edilmektedir.Modelimizin sonuçlarını her bir küme için farklı renklendirme yaparak inceledik. Şekil-4'de de görüldüğü üzere beş küme için gayet güzel bir dağılım mevcut. Grafiğin sağ alt tarafı incelendiğinde geliri yüksek ama harcamaları az olan müşterileri görüyoruz. Bu müşteriler gelirlerini dikkatli bir şekilde harcıyorlar. Grafiğin sağ üst kısımında ise hem gelirleri hem de harcamaları yüksek müşteriler bulunuyor. Bunlar, şirketlerin hedeflediği türden müşteriler. Buna karşılık en fazla müşteri sayısının grafiğin orta kısımda yer aldığı görülmekte. Bunlar orta gelir grubuna sahip müşteriler. Şirketler, sayıca fazla oldukları gerçeğini göz önüne alarak bu müşterileri de hedefleyebilir.

Bu yazımda elimden geldiğince sizlere hiyerarşik kümele yönteminden bahsettim.

Bilimin mihraplarında bilginin peşinde koşmanız dileğiyle. Sağlıcakla Kalın 🙂

Kaynaklar:

[1] – https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html

[2] – Haldun Akpınar, *DATA*, Papatya Bilim

[3] – İlker Arslan, *Python ile Veri Bilimi,* Pusula

[4] – Metin Bilgin, *Makine Öğrenmesi*, Papatya Bilim

[5] – Barış Doğan, *Bankaların Gözetiminde Bir Araç Olarak Kümeleme Analizi: Türk Bankacılık Sektörü İçin Bir Uygulama (*Doktora Tezi)

[6] – Nazmiye Yalçın, *Kümeleme Analizi ve Uygulaması,* (Yüksek Lisans Tezi)

[7] – https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/

[8] – https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html#sklearn.cluster.AgglomerativeClustering

[9]- https://unsplash.com/photos/Q1p7bh3SHj8

[10]- Ethem Alpaydın, Yapay Öğrenme, Boğaziçi Üniversitesi Yayınları

Ek Bilgi:

- Kullanılan Python versiyonu: 3.7.3
- numpy 1.16.2

- scipy 1.5.4
- matplotlib 3.1.0
- pandas 1.0.4
- sklearn 0.23.2
- İşletim sistemi: macOS

Eyüp Kaan Ülgen

Hiyerarşik kümeleme Kümeleme Python Unsupervised Veri Bilimi Veri Görselleştirme

Yazar Hakkında Toplam 6 yazı

Eyüp Kaan Ülgen

İstanbul Üniversitesi, Astronomi ve Uzay Bilimleri Bölümünde Doktora Öğrencisi. Doktora çalışma konusu: Galaksi kümelerindeki parlak galaksilerin çevreyle olan ilişkisi. İlgi alanları; Galaksi Evrimi, Veri Bilimi, Makine Öğrenimi, Derin Öğrenme ...

Tüm yazılarını gör

Şunlar da ilginizi çekebilir

İlgili içerik

dbt (DataBuildTool) ile Veri Analitiği Yolculuğunda Yeni Bir Dönem

İlgili içerik

Veri Sürüm Kontrolü: Yazılımdan Veri Dünyasına

Önceki yazı

SSIS'de Merge ve Merge Join Farkı

Sonraki yazı ▸

Power BI'da Veri Modelleme ve Tablo İlişkileri

Yorumlar (Yorum yapılmamış)

Bir yanıt yazın

E-posta adresiniz yayınlanmayacak. Gerekli alanlar * ile işaretlenmişlerdir

Yorumunuzu yazın *

Adınız *

E-posta adresiniz *

Website

Daha sonraki yorumlarımda kullanılması için adım, e-posta adresim ve site adresim bu tarayıcıya kaydedilsin.

Yorum gönder

Son Yorumlar

Burçin Minitab'de X-R Kontrol Grafiği Oluşturma

Nihat Tensorflow Lite Modeli ile Colab Üzerinden Görüntü Sınıflandırma: Derin Öğrenme Uygulaması

melih Microsoft Excel' de VBA Yardımıyla Otomatik Olarak E-posta Gönderimi

Hakan NAKIŞ Microsoft Excel' de VBA Yardımıyla Çalışma Sayfalarını Ayrı Ayrı PDF Olarak Kaydetme

kenan R ile Makine Öğrenmesi Uygulamaları: Doğrusal Regresyon

Eyüp Kaan Ülgen

İstanbul Üniversitesi, Astronomi ve Uzay Bilimleri Bölümünde Doktora Öğrencisi. Doktora çalışma konusu: Galaksi kümelerindeki parlak galaksilerin çevreyle olan ilişkisi. İlgi alanları; Galaksi Evrimi, Veri Bilimi, Makine Öğrenimi, Derin Öğrenme ...

Son Yorumlar

Burçin Minitab'de X-R Kontrol Grafiği Oluşturma

Nihat Tensorflow Lite Modeli ile Colab Üzerinden Görüntü Sınıflandırma: Derin Öğrenme Uygulaması

melih Microsoft Excel' de VBA Yardımıyla Otomatik Olarak E-posta Gönderimi

Hakan NAKİŞ Microsoft Excel' de VBA Yardımıyla Çalışma Sayfalarını Ayrı Ayrı PDF Olarak Kaydetme

kenan R ile Makine Öğrenmesi Uygulamaları: Doğrusal Regresyon

Bunları Da Oku

1 **3 Adımda CRM (Müşteri İlişkileri Yönetimi)**

2 **AB Testi Nedir ? İstatistiksel A/B Testleri Nasıl Yapılır?**

3 **Aşırı Öğrenme (Overfitting)**

VBO Ekibine Özel

VBO Dosya Merkezi

VBO Ekip Eğitimi