BSM622 · HAFTA 9

Denetimsiz Öğrenme

Temel Bileşen Analizi · K-Means · Hiyerarşik Kümeleme · Model-Tabanlı Kümeleme · Gower's Distance

Kaynak: Bruce, Bruce & Gedeck (2026), AI-Assisted Statistics for Data Scientists, 3rd ed., Bölüm 7

Doç. Dr. Onur Polat

Hacettepe Üniversitesi · Bilişim Enstitüsü · Veri Analitiği ve İstatistik

2026-2027 Güz · Tezsiz Yüksek Lisans

Hafta 9 — İçerik

Denetimsiz öğrenmenin dört temel aracı

01

Temel Bileşen Analizi (PCA)

Boyut indirgeme + yorumlama

02

K-Means Kümeleme

Hızlı ve ölçeklenebilir ayrıştırma

03

Hiyerarşik Kümeleme

Dendrogram + agglomerative algoritma

04

Model-Tabanlı Kümeleme

Çok değişkenli normal karışımı (GMM)

05

Ölçekleme + Kategorik veriler

Standardizasyon · Gower's Distance

BÖLÜM 1

Temel Bileşen Analizi (PCA)

Karl Pearson — değişkenlerin birlikte değişimini keşfetmek

PCA — Temel Fikir

Değişkenler birlikte değişir — fazlalığı keyfedelim

Birçok problemde değişkenler birlikte değişir (covary) — örn. restoran hesabı ile bahşiş; CVX ile XOM petrol hisselerinin günlük getirileri. PCA bu kovaryans yapısını keşfeder ve çok sayıda sayısal değişkeni, orijinal setin varyansının çoğunu açıklayan daha az sayıda yeni değişkenle (temel bileşenlerle) değiştirir.

Anahtar Kavramlar

Temel bileşen: tahmin değişkenlerinin doğrusal kombinasyonu

Yükler (Loadings): tahmin değişkenlerini bileşenlere dönüştüren ağırlıklar (eş anlamlı: ağırlıklar)

Scree grafik: bileşenlerin varyanslarının grafiği — göreli önemi gösterir

İki değişkenlik örnek: $Z_i = w_{i,1} \cdot X_1 + w_{i,2} \cdot X_2$

- Z_1 — toplam varyansı en çok açıklayan doğrusal kombinasyon
- Z_2 — Z_1'e dik, kalan varyansı mümkün olduğunca açıklar
- Daha fazla bileşen eklenirse her biri öncekilere diktir

CVX & XOM Örneği (sayfa 302)

İki enerji hissesi getirisi üzerinde PCA → Loadings: CVX: −0.747 / 0.665 XOM: −0.665 / −0.747. Birinci bileşen, iki hisse arasındaki yüksek korelasyonu yansıtan ortalama; ikinci bileşen, iki hissenin ne zaman ayrıştığını (divergence) ölçer. Her bileşen toplam varyansın %50'sini açıklıyor.

PCA — Hesaplama Algoritması

Varyans-maksimizasyonu, sıralı bileşenler

1. Toplam varyansı açıklayan yüzdeyi maksimize eden tahmin-değişken doğrusal kombinasyonu bul → bu, Z_1'dir
2. Aynı değişkenlerle farklı ağırlıklar kullanarak Z_2'yi oluştur. Ağırlıklar Z_1 ve Z_2'nin **korelasyonsuz** olacağı şekilde seçilir
3. Yeni değişken (bileşen) sayısı orijinal değişken sayısına ulaşana kadar devam eder
4. Varyansın çoğunu açıklayan ilk birkaç bileşeni tutmayı tercih et
5. Orijinal veriyi bu ağırlıklarla çarp → indirgenmiş boyutlu veri seti

Önemli notlar

- Sadece sayısal değişkenlerle çalışır — kategorik için Correspondence Analysis tercih edilir
- Korelasyon veya kovaryans matrisine dayanır — iterasyon yok, hızlıdır
- PCA, LDA'nın denetimsiz versiyonu olarak görülebilir

Temel Bileşenlerin Yorumlanması

Scree plot + yükler grafiği

Scree Plot (Serpinti Grafiği)

- Her bileşenin özdeğer/varyansını çubuk grafikte göster
- "Dirsek" (elbow) noktasına kadar olan bileşenleri tut
- Veya: kümülatif varyans %80-90'a ulaştığı noktayı seç
- Çok dik düşüş → 1-2 bileşen dominant

Yükler (Loadings) Grafiği

- Her bileşen için değişkenlerin yüklerini çubuk grafikte göster
- PC1'de tüm değişkenler aynı işaret → ortak faktör (örn. piyasa trendi)
- PC2 bazı değişkenleri +, bazılarını − ile ayrıştırır → sektör rotasyonu
- Yükler, bileşenin ekonomik anlamını ortaya koyar

S&P 500 örneği — 16 hisse üzerinde PCA'nın ilk 5 bileşeninin yorumu (sayfa 307)

PC1 Genel piyasa trendi — tüm hisseler aynı işaretli, en güçlü enerji (SLB, CVX, XOM)
PC2 Tüketici/teknoloji vs enerji rotasyonu — AAPL + perakende (COST, HD, WMT) +, enerji −
PC3 Apple vs defansif perakende — AAPL güçlü +, perakende −
PC4 Enerji yapısı — SLB (oil services) − vs entegre üreticiler (CVX, XOM, COP) +
PC5 Finansal risk vs defansif perakende — AXP −, COST +

Correspondence Analysis

PCA'nın kategorik versiyonu

Problem: PCA kategorik veriyle çalışmaz. Correspondence Analysis, kategoriler arasındaki ilişkileri keşfeder.

- **Girdi:** bir tablo — satırlar bir değişkenin kategorileri, sütunlar diğer değişkenin kategorileri, hücreler kayıt sayısı
- **Çıktı:** biplot — ölçeklenmiş eksenlerde dağılım grafiği (her eksenin açıkladığı varyans yüzdesi gösterilir)
- **Yakın noktalar birbirine ilişkili kategorilerdir** — görsel analizle çıkarım yapılır
- **Big data için değil** esas olarak düşük boyutlu kategorik veri görselleştirmesi için
- **Python kütüphanesi:** `prince (scikit-learn API ile uyumlu)`

Ev işleri örneği (sayfa 308, Figure 7-4)

13 ev işi × kim yapar (Wife/Husband/Jointly/Alternating) tablosunda CA uygulanır. Sonuçta çamaşır, kahvaltı, akşam yemeği "Wife" yakınında kümelenir; onarım ve araç kullanma "Husband" yakınında. Tatil ve finans "Jointly"ye yakındır. İki eksenin açıkladığı varyans: %48.7 + %39.9 = %88.6. Bu, ev işi dağılımının güçlü biçimde cinsiyetlenmiş olduğunu görselleştiriyor.

BÖLÜM 2

K-Means Kümeleme

Veriyi K gruba ayır — en basit ve en yaygın kümeleme yöntemi

K-Means Kümeleme

Küme-içi kareler toplamını minimize eden K küme

Temel fikir: Veriyi K kümeye ayır — her kayıt ile atandığı kümenin ortalaması arasındaki karedevik uzaklıkların toplamını minimize et.

Anahtar Kavramlar

Küme (Cluster): birbirine benzer kayıtlardan oluşan grup
Küme ortalaması: küme içindeki kayıtların değişken-ortalamaları vektörü
K: oluşturulacak küme sayısı — kullanıcı tarafından belirlenir

Küme-içi Kareler Toplamı

$$SS_k = \Sigma\ (x_i - \bar{x}_k)^2 + (y_i - \bar{y}_k)^2 \quad \text{for } i \in \text{Cluster } k$$

K-Means, $\Sigma(SS_1 + SS_2 + \ldots + SS_K)$'yi minimize eden atama yapar. Sonuç: iyi ayrılmış kümeler — ancak kümelerin boyutu eşit olmayabilir.

⚠ Normalizasyon Gerekli

Sürekli değişkenleri standartlaştır (z-skor): ortalamayı çıkar, standart sapmaya böl. Aksi halde büyük ölçekli değişkenler (örn. yıllık gelir) kümeleme sürecini domine eder ve küçük ölçekli değişkenler (örn. çalışılan yıl) rolünü kaybeder.

Not:

Günlük hisse getirileri doğaları gereği standartlaştırılmış olduğundan onlarda normalizasyona gerek yoktur.

K-Means Algoritması

İteratif atama-güncelleme döngüsü

Girdi: p değişkenli veri X_1, …, X_p ve kullanıcı tarafından belirlenen K.

Başlangıç: K rastgele küme ortalaması belirle (ya da kayıtları K kümeye rastgele ata, ortalamaları hesapla)

İteratif adım 1: Her kaydı en yakın küme ortalamasına ata (karedevik uzaklığa göre)
İteratif adım 2: Yeni atamalara göre küme ortalamalarını güncelle

Yakınsama: Kayıt atamaları değişmediğinde algoritma durur

Birden Çok Başlangıç (n_init)

Algoritma lokal minimumda takılabilir. 10 farklı rastgele başlangıç ile tekrarla (scikit-learn'de varsayılan n_init=10), en düşük küme-içi kareler toplamını veren çözümü seç. iter.max / max_iter ile her başlangıçtaki maksimum iterasyon sayısını sınırla (sklearn varsayılan 300).

Kümeleri Yorumlama + K Seçimi

Küme boyutları, ortalamalar ve dirsek yöntemi

Küme Çıktıları

- Küme boyutları (size): dengesizlik varsa outlier'a bak
- Küme ortalamaları (cluster centers): her küme ne temsil eder
- 16 S&P 500 hissesi için 5 küme → piyasa rejimleri:
  C1: tüketici/tech-önderliğinde büyüme
  C2: enerji-özgü düşüş
  C3: enerji ↑, tüketici ↓
  C4: tüm sektörler bull-market
  C5: sistemik stres/düzeltme

Elbow (Dirsek) Yöntemi

- K = 2-14 için her değerde modeli kur
- Her K için küme-içi SS (inertia) veya % açıklanan varyansı kaydet
- Grafiği çiz → "dirsek" (artışın yavaşladığı nokta) = optimum K
- Pratik: Gürültülü veride net dirsek olmayabilir — sezgisel karar

K seçiminde pratik yaklaşımlar (sayfa 319)

1. **Uygulama kuralları:** örn. satış ekibi müşteri segmentasyonu için 4-6 persona (2 çok az, 8 çok fazla)
2. **Elbow yöntemi:** kümülatif açıklanan varyansın düzleştiği nokta
3. **Gap statistic (Tibshirani):** istatistiksel yaklaşım, nadiren gerekli
4. **Cross-validation:** kümelerin yeni veri üzerinde tekrar üretilebilirliği

→ *Tek bir kural yoktur; en önemli test:* ***kümeler yorumlanabilir mi?***

BÖLÜM 3

Hiyerarşik Kümeleme

Dendrogram + agglomerative algoritma

Hiyerarşik Kümeleme — Temel Fikir

Her kayıt kendi kümesidir, kümeler birleştirilir

K-Means'in alternatifi: farklı K değerlerinin sonucunu aynı anda görmek, outlier/aykırı grupları keşfetmek, sezgisel dendrogram görseli elde etmek.

Anahtar Kavramlar

Dendrogram: kayıtların ve kümelerin hiyerarşisini gösteren ağaç
Distance: iki kaydın birbirine ne kadar yakın olduğunun ölçüsü
Dissimilarity: iki kümenin birbirine ne kadar yakın olduğunun ölçüsü

Avantajları ve Dezavantajları

✓ K'yı önceden belirlemeye gerek yok
✓ Dendrogram ile farklı K'lar aynı anda görülür
✓ Outlier'ları keşfetmede duyarlı
✗ Milyonlarca kayıt için uygun değil (ölçeklenmez)
✗ Hesaplama maliyeti: $O(n^2)$ hafıza, $O(n^3)$ zaman

S&P 500 hisseleri örneği (sayfa 324, Figure 7-8)

18 hisse üzerinde hiyerarşik kümeleme → dendrogramda:
• Google ve Amazon birbirinden ve diğerlerinden çok farklı — her biri kendi kümesinde
• Petrol hisseleri (SLB, CVX, XOM, COP) birlikte kümeleniyor
• Apple kendi başına
• Geri kalanlar birbirine benzer
Dendrogramı yatay bir çizgi ile kesmek → istenen küme sayısı elde edilir (cutree / fcluster).

Agglomerative (Birleştirici) Algoritma

Aşağıdan yukarıya hiyerarşi inşası

1. Her kayıt tek başına bir küme olarak başlar (n kayıt → n küme)

2. Tüm kümeler arasındaki dissimilarity $D(C_k, C_l)$ hesapla

3. En düşük dissimilarity'ye sahip iki kümeyi birleştir

4. Birden fazla küme kalırsa 2'ye dön; aksi halde algoritma biter (tek bir ana küme)

Kayıtlar arası uzaklık (Öklid):

$$d(x,y) = \sqrt{[(x_1-y_1)^2 + (x_2-y_2)^2 + \ldots + (x_p-y_p)^2]}$$

Dendrogram çizimi: Python'da `scipy.cluster.hierarchy.linkage(X, method=...)` + `dendrogram(Z)`

Dissimilarity Ölçüleri

Dört temel küme-arası uzaklık tanımı

Complete Linkage

$$D(A,B) = \max d(a_i, b_j)$$

İki kümedeki tüm çift kayıtlar arasındaki EN BÜYÜK uzaklık. Benzer üyeli, kompakt kümeler üretir. Varsayılan seçim.

Single Linkage

$$D(A,B) = \min d(a_i, b_j)$$

İki kümedeki tüm çift kayıtlar arasındaki EN KÜÇÜK uzaklık. "Greedy" — birçok noktayı tek kümeye atar, zincir etkisi yaratır.

Average Linkage

$$D(A,B) = \text{mean } d(a_i, b_j)$$

Tüm çift uzaklıkların ortalaması. Complete ve single arasında bir uzlaşım.

Minimum Variance (Ward)

$$\text{Ward's method}$$

Küme-içi kareler toplamını minimize eder — K-Means'e en yakın yöntem. Dengeli, kompakt kümeler üretir.

BÖLÜM 4

Model-Tabanlı Kümeleme

Çok değişkenli normal karışımlarının Bayesian seçimi

Model-Tabanlı Kümeleme

Her küme bir çok değişkenli normal dağılımdan gelir

Fark: K-Means ve hiyerarşik kümeleme heuristiktir (sadece uzaklığa bakar). Model-tabanlı kümeleme istatistiksel teorik temele dayanır.

Çok değişkenli normal dağılım (Multivariate Normal)

$X_1, X_2, \ldots, X_p \sim N_p(\mu, \Sigma)$ — μ: ortalama vektörü (p boyutlu), Σ: kovaryans matrisi (p×p, simetrik)
Kovaryans matrisi değişkenler arası korelasyonları yakalar. Kovaryans $\sigma_{ij} > 0 \rightarrow$ değişkenler pozitif korele.

Normal Karışımları (Mixture of Normals)

Her kayıt K tane çok değişkenli normal dağılımdan birinden örneklenmiş kabul edilir — her birinin kendi μ_k ve Σ_k'si. K, küme sayısıdır. R: mclust paketi; Python: sklearn.mixture.GaussianMixture.

⚠ *Finansal getiriler gibi uzun kuyruk dağılımlarında, GMM ikinci bir "yüksek varyanslı" bileşen ekler. K-Means kompakt kümeler üretirken, GMM outlier'ları yakalar.*

BIC ile Küme Sayısı Seçimi

Bayesian Information Criterion ile otomatik K seçimi

BIC formülü: BIC = −2·log(L) + k·log(n) — L: likelihood, k: parametre sayısı, n: gözlem sayısı. Düşük BIC daha iyidir.

Amaç: daha fazla küme ekledikçe likelihood artar ama parametre sayısı da artar
Denge: BIC, fit iyileşmesini parametre cezasıyla dengeler — overfitting'i önler
K seçimi: K=1'den K=8-14'e kadar denenir, en düşük BIC'li çözüm seçilir
Kovaryans tipi: full / tied / diag / spherical — her biri farklı sayıda parametre, GMM her birini dener

Model-Tabanlı Kümelemenin Sınırları

- **Dağılım varsayımı** gerektirir — veri normal değilse sonuç yanıltıcı olabilir
- Hesaplama maliyeti hiyerarşik kümelemeden bile yüksek — büyük veriye zor ölçeklenir
- Algoritma daha karmaşık ve diğer yöntemlere göre daha az erişilebilir
- **Pratik tavsiye:** *önce K-Means dene; kompakt olmayan yapılar varsa GMM'e geç*

BÖLÜM 5

Ölçekleme ve Kategorik Değişkenler

Standardizasyon · Dominant değişkenler · Gower's Distance

Değişkenleri Ölçekleme

Farklı birimlerin baskın olmasını önle

⚠ Problem (sayfa 335)

Kredi veri setinde değişkenler çok farklı ölçeklerde: loan_amnt (binlerce $), annual_inc (on binlerce $), open_acc (0-50 arası sayı), dti (0-30 arası oran). Ölçeklenmeden K-Means uygulanırsa kümeler sadece annual_inc ve revol_bal'a göre belirlenir — küçük ölçekli değişkenler rol oynamaz. Birinci küme 52 kişi (yüksek gelirli), diğerleri çok kalabalık → yanlış kümeleme.

Standardizasyon (z-skor)

$$z = (x - \bar{x}) / s$$

Her değişken: ortalamayı çıkar, standart sapmaya böl. Python: StandardScaler. Ölçek bağımsız analizler için gerekli — PCA ve tüm kümeleme yöntemleri için.

Normalize edilmiş sonuç

Standardize edilmiş veriyle K-Means → küme boyutları çok daha dengeli: 7355 / 5309 / 3713 / 6294. annual_inc ve revol_bal artık domine etmiyor, dti ve diğerleri de rol alıyor. Yapı daha anlamlı hale geliyor.

💡 **Dominant Değişkenler: Aynı ölçekteki veriler bile bazen rescale edilmelidir. Örn. S&P 500'e GOOGL ve AMZN eklenirse, bu iki hissenin varyansı diğerlerini çok geride bırakır → ilk iki bileşen neredeyse tamamen bu iki hisse tarafından belirlenir.**

Gower's Distance

Karışık (sayısal + kategorik) veriler için özel uzaklık

Temel Fikir

- **Sayısal değişkenler ve ordinal faktörler:** iki kaydın mutlak fark (Manhattan uzaklığı)
- **Kategorik değişkenler:** iki kaydın kategorisi farklıysa 1, aynıysa 0
- Her değişken uzaklığı [0, 1] aralığına ölçeklenir
- Pairwise uzaklıklar **basit ya da ağırlıklı ortalama** ile birleştirilir → nihai uzaklık matrisi

Hesaplama Adımları

1. Her değişken çifti (i, j) için d_{ij} hesapla
2. Her çift d_{ij}'yi 0-1 aralığına ölçekle
3. Ölçeklenmiş uzaklıkları ortalayarak nihai matrisi oluştur

Kullanım

- R: cluster::daisy(data, metric="gower")
- Python: gower.gower_matrix(X, cat_features=[...])
- Çıktı: scipy.cluster.hierarchy.linkage ile hiyerarşik kümeleme için uzaklık matrisi
- K-Means doğrudan kullanamaz!

⚠ K-Means + one-hot encoded kategorik veri: binary değişkenler z-skor ile domine eder, kümeler faktör seviyelerinin kopyası olur. Çözüm: Gower's Distance + hiyerarşik kümeleme.

Dört Yöntemin Karşılaştırması

Hangi yöntem ne zaman?

Yöntem	Hız	K önceden?	Ölçeklenme	Veri türü	Güçlü yönü
PCA	Hızlı	Bileşen sayısı	Çok iyi	Sadece sayısal	Boyut indirgeme + yorumlanabilirlik
K-Means	Çok hızlı	Evet (K)	Çok iyi	Sayısal (normalize)	Büyük veriye ölçeklenir, basit
Hiyerarşik	Yavaş $O(n^2)$	Hayır	Zayıf	Karışık (Gower's)	Dendrogram, outlier keşfi
GMM (Model)	Yavaş	Hayır (BIC)	Zayıf	Sayısal	İstatistiksel temel, dağılım modelleme

👂 ***Pratik öneri: Büyük veri → K-Means. Küçük veri + keşifsel analiz → Hiyerarşik. Karışık veri (sayısal + kategorik) → Gower + Hiyerarşik. Dağılım-dayalı analiz → GMM. Boyut indirgeme → PCA (önce!).***

Hafta 9 — Özet

Denetimsiz öğrenmenin dört yapıtaşı

PCA

- Sayısal değişkenlerin doğrusal kombinasyonu
- Scree + loadings yorumu
- Boyut indirgeme + büyük data için ön işlem

K-Means

- Küme-içi SS minimize
- 2 adımlı iteratif algoritma
- Elbow yöntemi ile K seçimi

Hiyerarşik

- Agglomerative: n küme → 1 küme
- Dendrogram görseli
- 4 dissimilarity ölçüsü (Ward, Complete...)

Model (GMM)

- Çok değişkenli normal karışımı
- BIC ile otomatik K seçimi
- Outlier'ları ikinci bileşende yakalar

Ortak İlke — Ölçekleme

Tüm denetimsiz öğrenme yöntemlerinde değişkenlerin uygun şekilde ölçeklenmesi şarttır. Normalizasyon (z-skor) → sayısal veriler için en yaygın; Gower's Distance → sayısal + kategorik karışım için. Bu olmadan büyük ölçekli değişkenler analizi domine eder.