Topaklandırma

Dr. Öğr. Üyesi Işık İlber Sırmatel

T.C. Trakya Üniversitesi
Mühendislik Fakültesi
Elektrik - Elektronik Mühendisliği Bölümü
Kontrol Anabilim Dalı

Kaynak (source)

Lecture Slides for Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares.

Stephen Boyd, Lieven Vandenberghe

Konu listesi

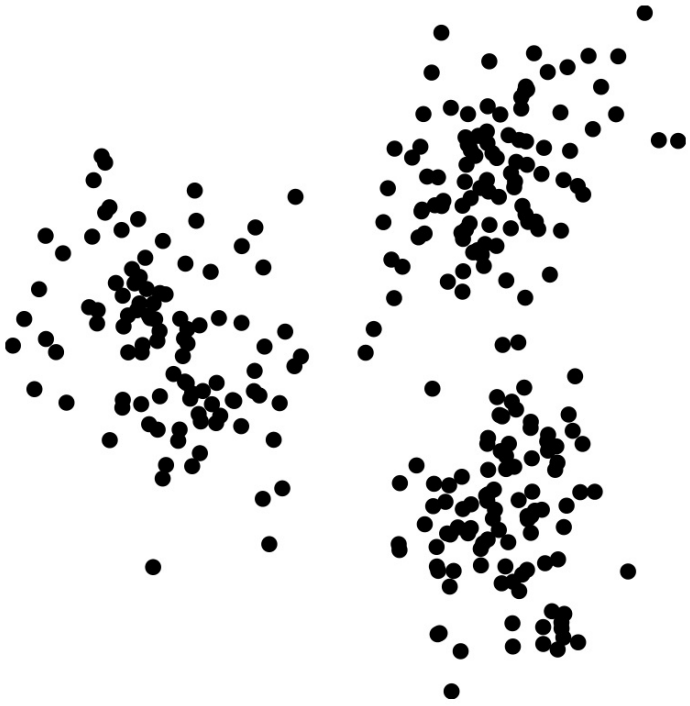
1. Tanım ve örnekler
2. Algoritma (topaklandırma)
3. Örnek (çalışma şekli)

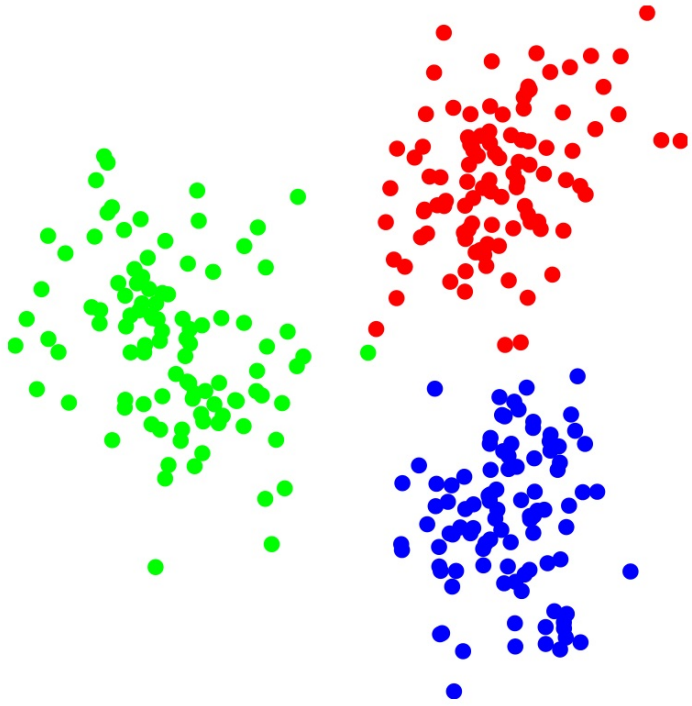
Bölüm 1

Tanım ve örnekler

Topaklandırma (*clustering*)

- N adet n-vektör $x_1, x_2 \ldots, x_N$ verilsin
- amaç: vektörleri k adet grup olarak ayır
- not: ayırma, topaklandırma ve bölüntüleme (*partitioning*) kavramlarını burada eş anlamlı kullanıyoruz
- aynı gruptaki vektörlerin birbirine yakın olmasını isteriz





Uygulama alanları

- konu keşfi (*topic discovery*) ve belge sınıflandırma (*classification*)
  - x_i belge i'nin sözcük sayısı histogramı
- hasta topaklandırma
  - x_i hasta i'nin özellikleri, tahlil sonuçları, semptomlar
- müsteri piyasa bölümlendirmesi (*market segmentation*)
  - x_i müşteri i'nın alışveriş geçmişi ve diğer özellikleri
- görüntü renk sıkıştırma (*compression*)
  - x_i RGB piksel değerleri
- ekonomik sektörler
  - x_i şirket i'nin finansal özellikleri

Topaklandırma amaç fonksiyonu

- grup j, $G_j \subset \{1, \ldots, N\}$ ile gösterilir ($j = 1, \ldots, k$)
- x_i'in dahil olduğu grup c_i ile gösterilir: $i \in G_{c_i}$
- grup temsilcileri: n-vektörler $z_1, \ldots, z_k$
- topaklandırma amaç fonksiyonu (*objective function*):

$$J^{\text{topak}} = \frac{1}{N} \sum_{i=1}^{N} \|x_i - z_{c_i}\|^2$$

  (ilgili temsilcilerle vektörler arasındaki mesafenin ortalama-karesel değeri)
- J^{topak}'nin küçük olması iyi topaklandırma anlamına gelir
- amaç: J^{topak}'yi minimize edecek şekilde topaklandırmayı (c_i) ve temsilcileri (z_j) seç

Bölüm 2

Algoritma (topaklandırma)

Algoritma (topaklandırma)

verilen temsilciler için vektörleri bölüntüleme (*partitioning*)

- temsilcilerin $(z_1, \ldots, z_k)$ verildiğini farz edelim
- vektörler gruplara nasıl atanır? (yani, $c_1, \ldots, c_N$ nasıl seçilir?)

- c_i J^{topak}'de sadece $\|x_i - z_{c_i}\|^2$ teriminde mevcuttur
- c_i üzerinden minimize etmek için, $\|x_i - z_{c_i}\|^2 = \min_j \|x_i - z_j\|^2$'yi sağlayacak şekilde c_i seçilir (yani, her x_i vektörü kendisine en yakın temsilci z_j'ye atanır)

Algoritma (topaklandırma)

verilen bölüntü için temsilcileri seçme

- verilen bölüntü $G_1, G_2, \ldots, G_k$ için, J^{topak}'yi minimize edecek şekilde temsilciler $z_1, z_2, \ldots, z_k$ nasıl seçilir?
- J^{topak} k adet toplamın (her z_j için bir tane) toplamı olarak ayrılabilir:

$$J^{\text{topak}} = J_1 + J_2 + \cdots + J_k, \qquad J_j = \frac{1}{N} \sum_{i \in G_j} \|x_i - z_j\|^2$$

- dolayısıyla, z_j'yi kendi grubundaki noktalara olan ortalama-karesel uzaklığı minimize edecek şekilde seçeriz
- bu z_j noktası, j grubundaki noktaların ortalamasıdır (yani, geometrik merkezidir (*centroid*))

$$z_j = \frac{1}{|G_j|} \sum_{i \in G_j} x_i$$

(not: $|G_j|$, G_j bölüntüsündeki nokta sayısıdır)

k-ortalamalar (k-*means*) algoritması

- bölüntü ve temsilciler sırayla güncellenir
- amaç fonksiyonu J^{topak} her adımda azalır

verilenler: $x_1, x_2, \ldots, x_n \in \mathbb{R}^n$; $z_1, z_2, \ldots, z_n \in \mathbb{R}^n$

tekrarla:

1) bölüntüyü güncelle: i'yi G_j'ye ata, $j = \underset{j'}{\operatorname{argmin}} \|x_i - z_{j'}\|^2$

2) geometrik merkezleri güncelle: $z_j = \dfrac{1}{|G_j|} \displaystyle\sum_{i \in G_j} x_i$

$z_1, z_2, \ldots, z_n$ değişmeyi bıraktığında **dur**

k-ortalamalar algoritmasının yakınsaması

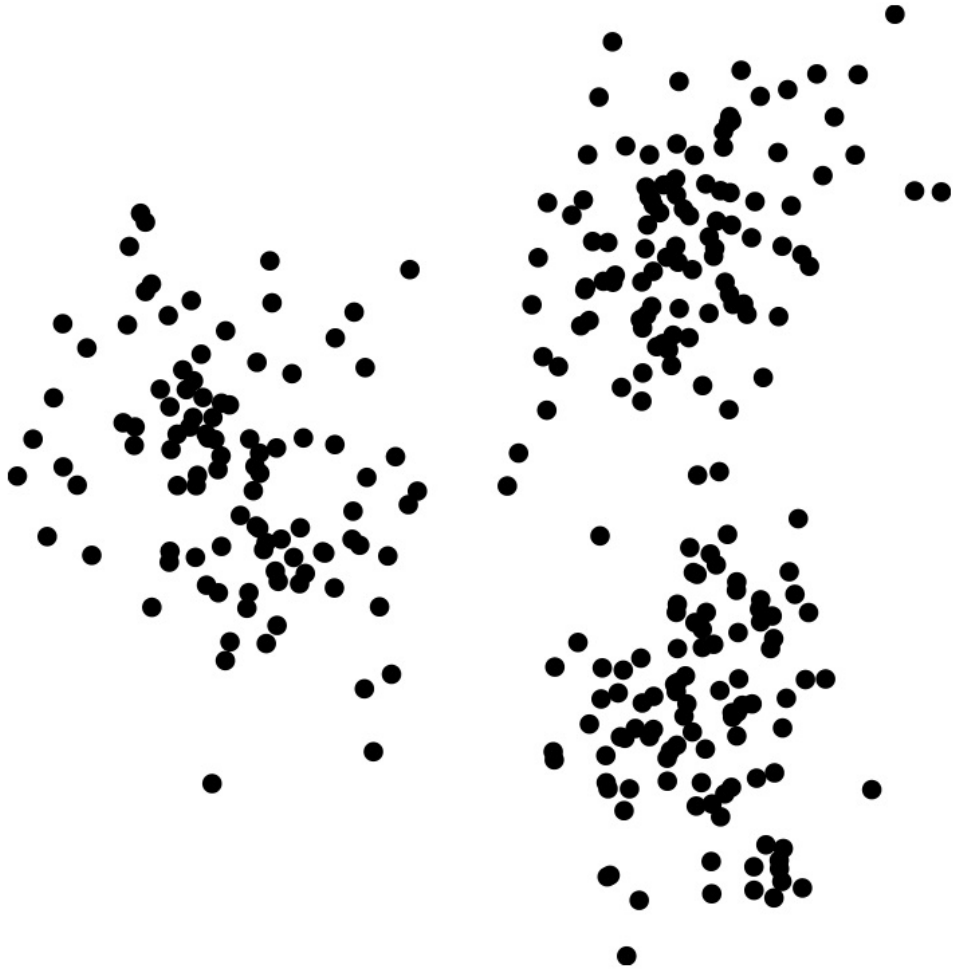
- J^{topak} her adımda ($z_1, z_2, \ldots, z_n$ değişmeyi bırakana kadar) azalır
- ancak (genel olarak) k-ortalamalar algoritması J^{topak}'yi minimize eden bölüntüyü bulmaz
- k-ortalamalar algoritması buluşsal (*heuristic*) bir yöntemdir: J^{topak}'nin mümkün olan en küçük değerini bulma garantisi yoktur
- sonuçta elde edilen bölüntü (ve karşılık gelen J^{topak} değeri) başlangıç temsilcilerine bağlı olarak değişebilir
- yaygın yaklaşım:
  - k-ortalamalar algoritmasını farklı (genellikle rastgele seçilmiş) başlangıç temsilcileriyle 10 defa çalıştır
  - en küçük J^{topak} değerini veren bölüntüyü seç

Bölüm 3

Örnek (çalışma şekli)

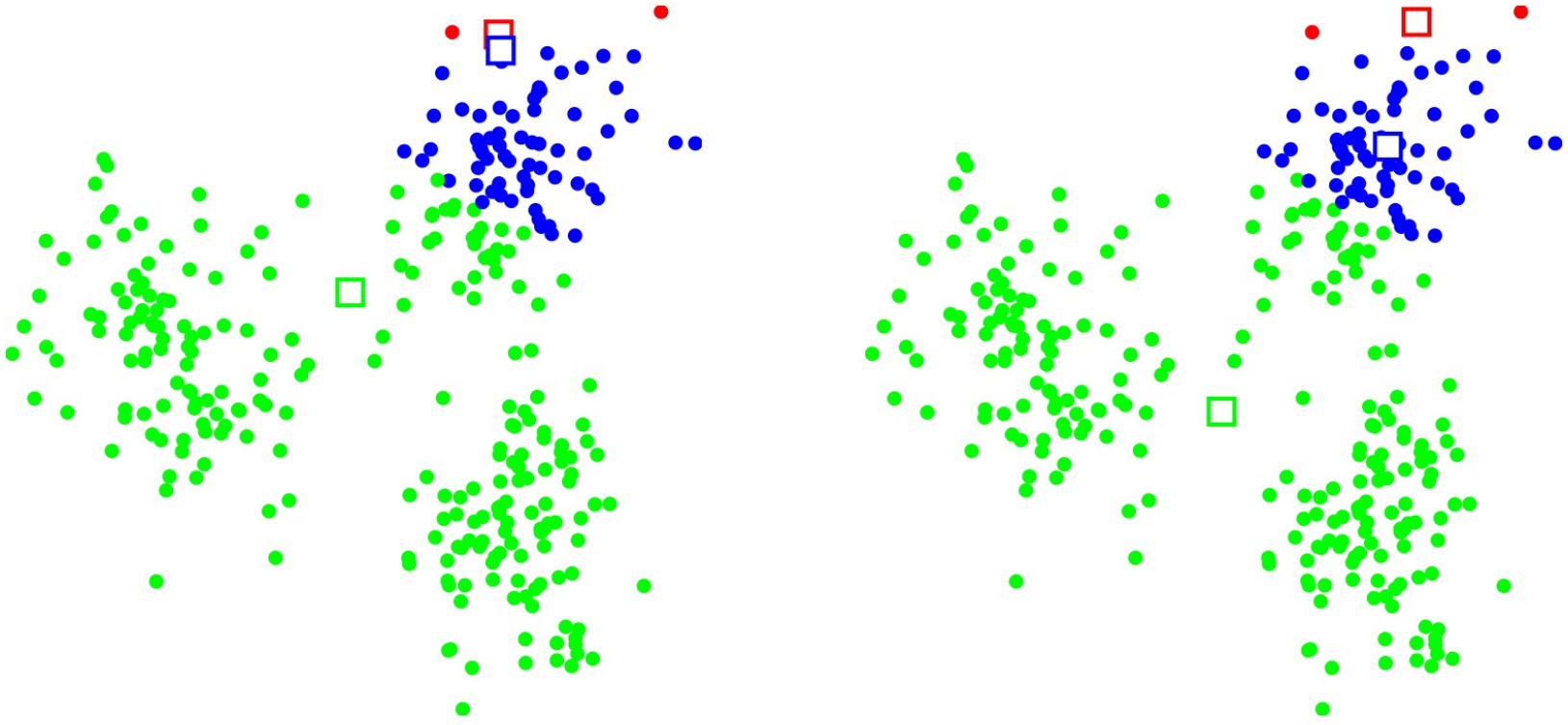
k-ortalamalar algoritması - Örnek

veri (N adet n-vektör $x_1, x_2 \ldots, x_N$)



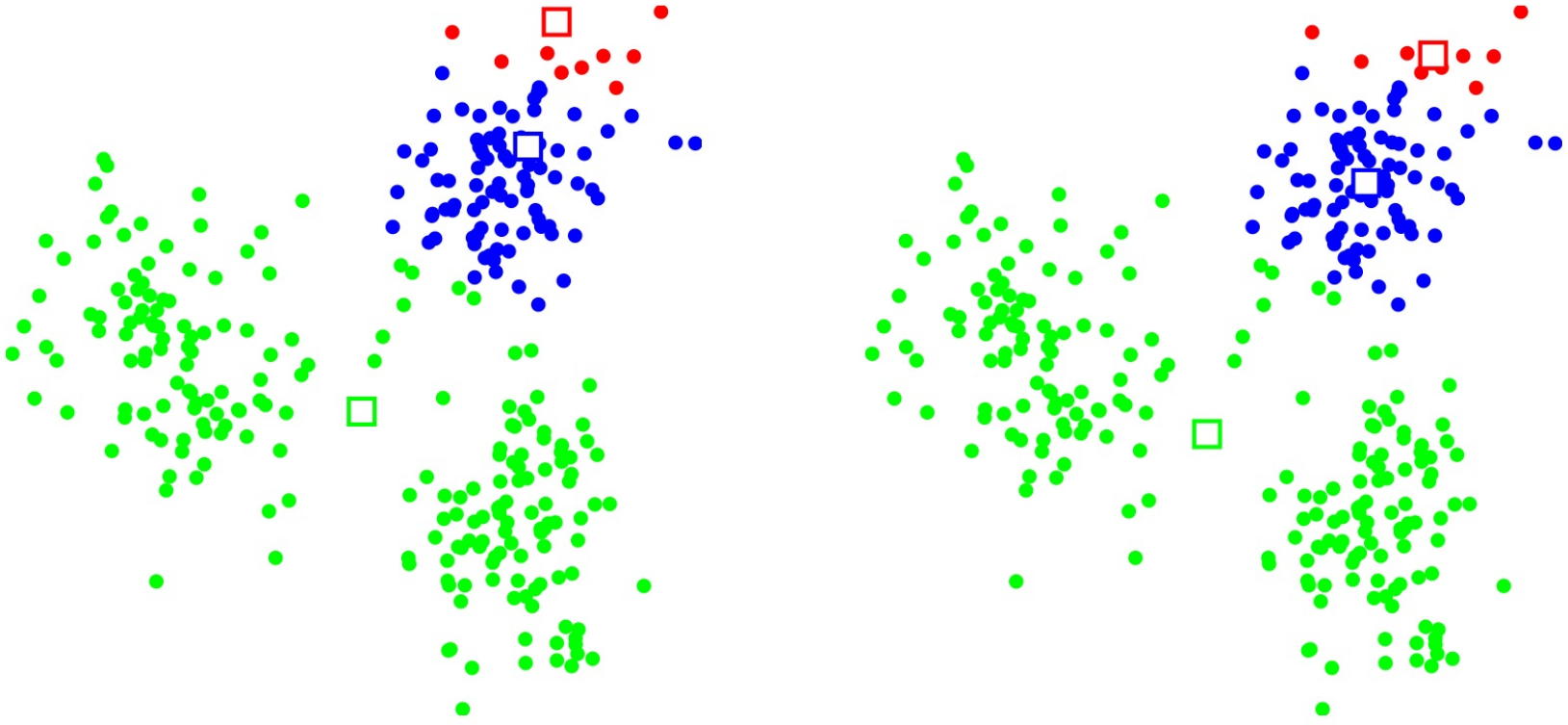
k-ortalamalar algoritması - Örnek

yineleme 1



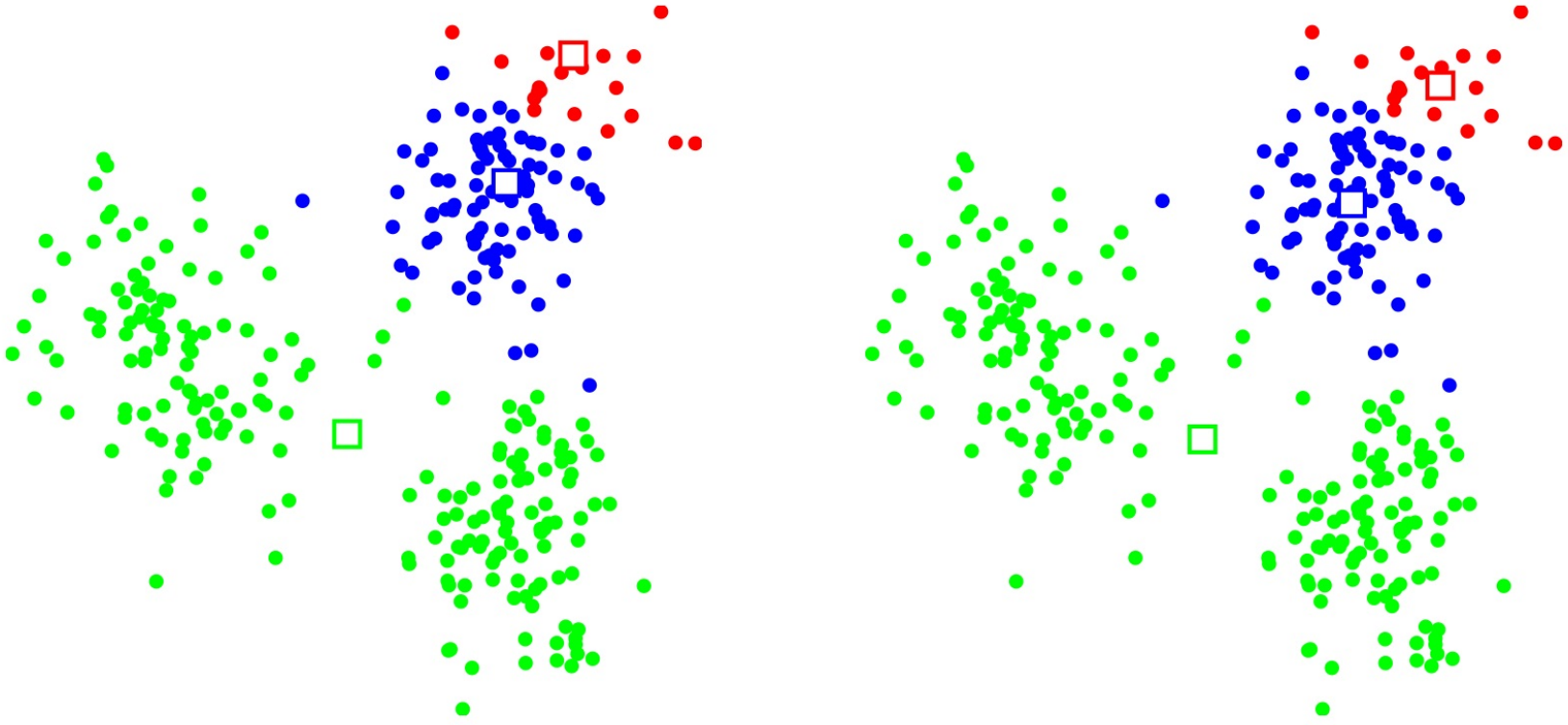
k-ortalamalar algoritması - Örnek

yineleme 2



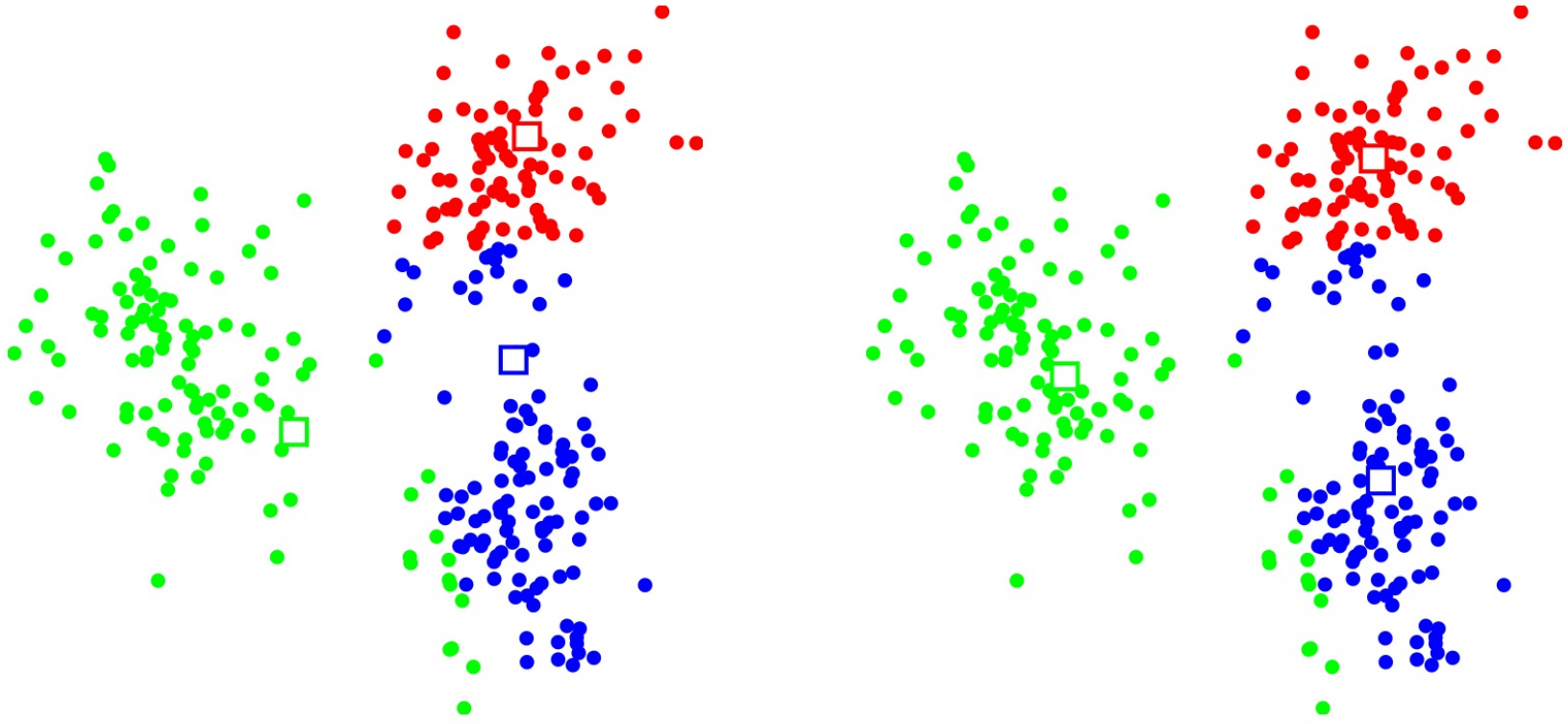
k-ortalamalar algoritması - Örnek

yineleme 3



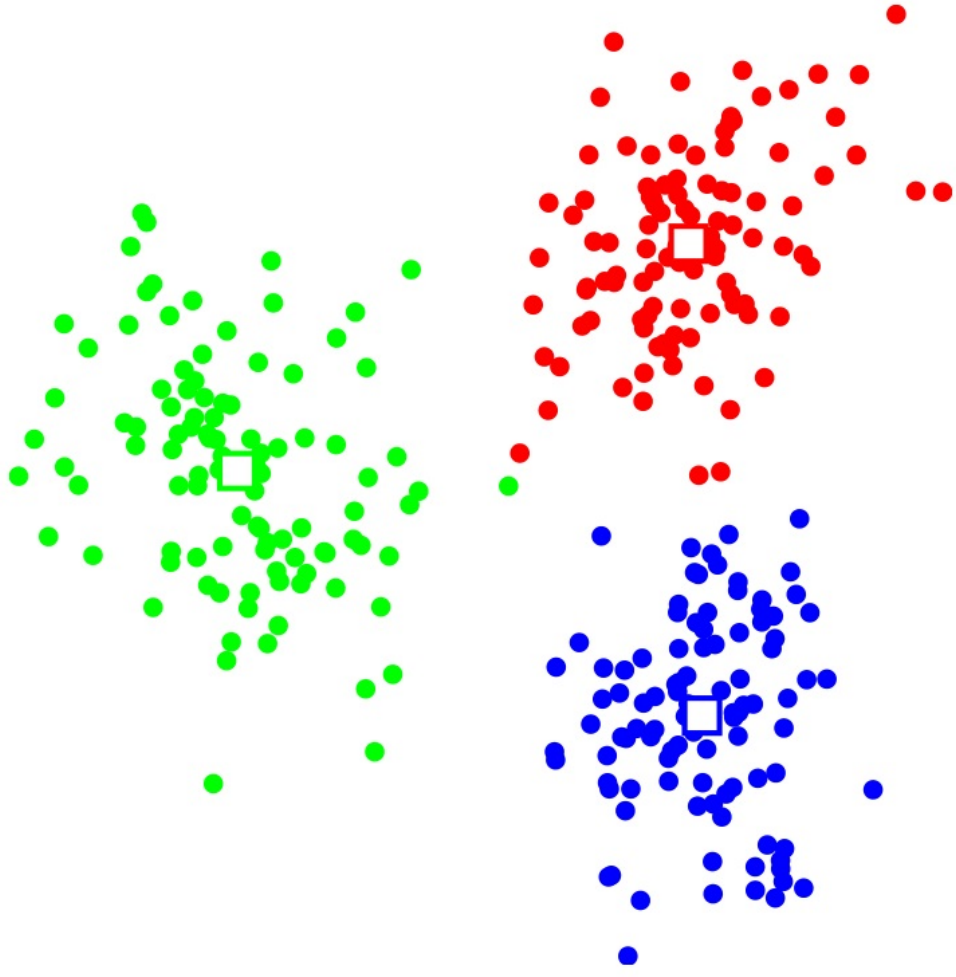
k-ortalamalar algoritması - Örnek

yineleme 10



k-ortalamalar algoritması - Örnek

sonuç (yineleme 15)



k-ortalamalar algoritması - Örnek