Phân cụm K-Means

Le Nhat Tung

Contents

1 Giới thiệu về K-Means

1.1 Thuật toán phân cụm K-Means là gì?

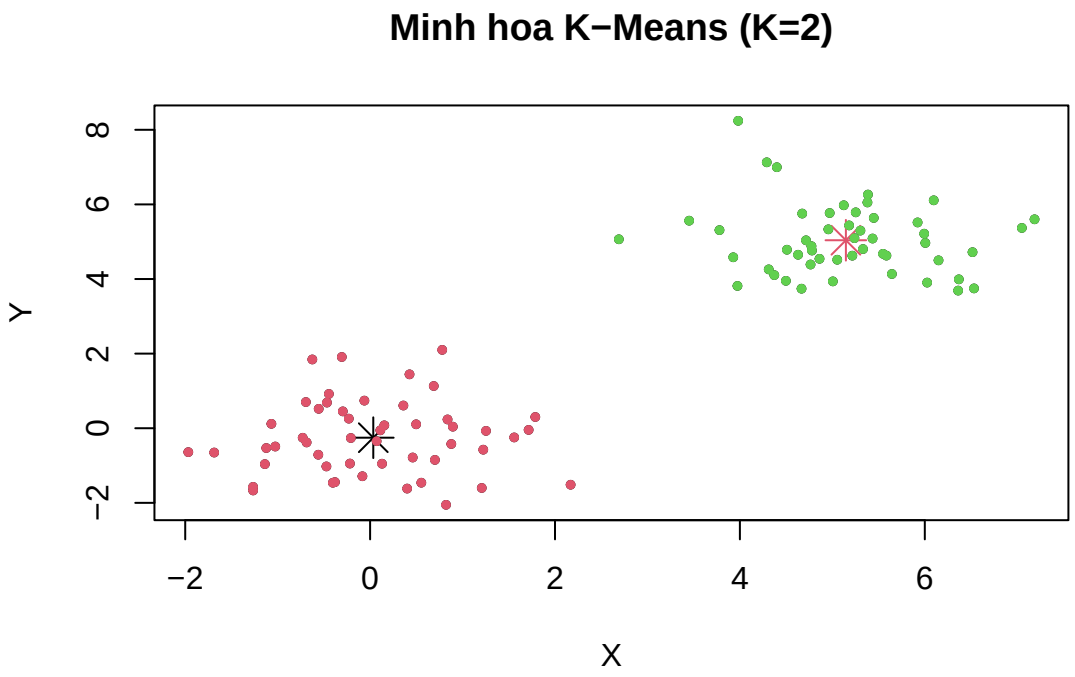
K-Means là một trong những thuật toán phân cụm (clustering) phổ biến nhất trong học máy không giám sát (unsupervised learning). Thuật toán này nhằm mục đích phân chia một tập dữ liệu thành K cụm (clusters) khác nhau, trong đó mỗi điểm dữ liệu thuộc về cụm có trung tâm (centroid) gần nhất Hannun et al. (2014).

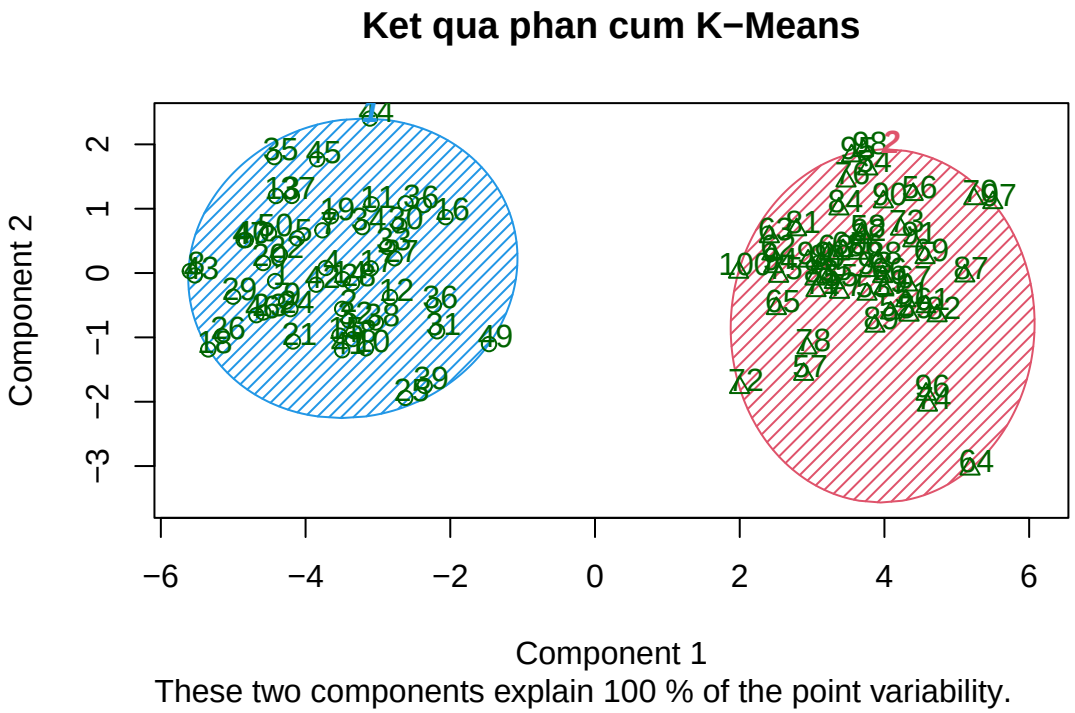
Mục tiêu chính của K-Means là giảm thiểu tổng bình phương khoảng cách giữa các điểm dữ liệu và trung tâm cụm của chúng, được gọi là **Within-Cluster Sum of Squares (WCSS)**:

$$WCSS = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

Trong đó:

- K là số lượng cụm
- C_i là tập hợp các điểm thuộc cụm thứ i
- μ_i là trung tâm (centroid) của cụm thứ i
- $||x - \mu_i||^2$ là bình phương khoảng cách Euclidean giữa điểm x và trung tâm cụm μ_i





1.2 Các bước của thuật toán K-Means

Thuật toán K-Means hoạt động theo các bước sau:

1. **Khởi tạo**: Chọn K điểm làm trung tâm cụm ban đầu (có thể ngẫu nhiên hoặc theo chiến lược khác)

2. **Gán nhãn**: Gán mỗi điểm dữ liệu cho cụm có trung tâm gần nhất
3. **Cập nhật trung tâm**: Tính lại trung tâm của từng cụm bằng cách lấy trung bình cộng của tất cả các điểm trong cụm đó
4. **Lặp lại**: Lặp lại bước 2 và 3 cho đến khi các trung tâm cụm không thay đổi nhiều (hội tụ) hoặc đạt đến số lần lặp tối đa

2 Vấn đề lựa chọn K trong K-Means

Một trong những thách thức lớn nhất khi sử dụng K-Means là xác định số lượng cụm K phù hợp. Không có một giá trị K tối ưu duy nhất cho mọi bộ dữ liệu, và việc chọn K phụ thuộc vào đặc điểm của dữ liệu và mục tiêu phân tích.

2.1 Các phương pháp xác định K tối ưu

2.1.1 1. Phương pháp Elbow (Khuỷu tay)

Phương pháp Elbow là một trong những cách phổ biến nhất để xác định số cụm K phù hợp. Phương pháp này dựa trên việc đánh giá tổng bình phương khoảng cách trong cụm (Within-Cluster Sum of Squares - WCSS) khi thay đổi giá trị K.

Nguyên lý hoạt động: - Khi tăng số lượng cụm K, WCSS sẽ luôn giảm (tới giới hạn WCSS = 0 khi K = n, với n là số điểm dữ liệu) - Tuy nhiên, tốc độ giảm WCSS không đều - ban đầu giảm nhanh, sau đó chậm dần - "Điểm khuỷu tay" (elbow point) là điểm mà sau đó việc tăng K không làm giảm WCSS đáng kể

Công thức tính WCSS:

$$WCSS = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

Đánh giá phương pháp: - **Ưu điểm**: Trực quan, dễ hiểu và dễ thực hiện - **Nhược điểm**: Đôi khi "điểm khuỷu tay" không rõ ràng hoặc không tồn tại - **Phạm vi giá trị**: WCSS luôn không âm, giá trị càng thấp càng tốt (cần cân đối với số lượng cụm) - **Khi nào tốt**: K tối ưu là điểm mà WCSS "gãy khúc" rõ ràng - thường là điểm mà đường cong bắt đầu phẳng dần

Các bước thực hiện: 1. Tính WCSS cho các giá trị K khác nhau (thường từ 1 đến 10 hoặc 15) 2. Vẽ đồ thị WCSS theo K 3. Tìm "điểm khuỷu tay" - điểm mà tại đó WCSS bắt đầu giảm chậm lại đáng kể

```
# Cài đặt và nạp các gói cần thiết
library(cluster)  # Cho phân tích cụm
library(factoextra)  # Cho trực quan hóa phân cụm
library(NbClust)  # Cho việc xác định số cụm tối ưu
library(fpc)  # Cho metrics đánh giá cụm
```

```
# Tạo dữ liệu mẫu
set.seed(123)
# Tạo 3 cụm dữ liệu
cluster1 <- data.frame(
  x = rnorm(50, 0, 1),
  y = rnorm(50, 0, 1)
)
cluster2 <- data.frame(
  x = rnorm(50, 5, 1),
  y = rnorm(50, 5, 1)
)
cluster3 <- data.frame(
  x = rnorm(50, 10, 1),
  y = rnorm(50, 0, 1)
```

```r
)

# Gộp dữ liệu
synthetic_data <- rbind(cluster1, cluster2, cluster3)
```

```r
wcss <- numeric(10)
for(i in 1:10) {
  kmeans_model <- kmeans(synthetic_data, centers=i, nstart=25)
  wcss[i]  <- kmeans_model$tot.withinss
}


# Vẽ biểu đồ Elbow
plot(1:10, wcss, type = "b", pch = 19,
     xlab = "So luong cum (K)",
     ylab = "Tong binh phuong khoang cach trong cum (WCSS)",
     main = "Phuong phap Elbow")
```

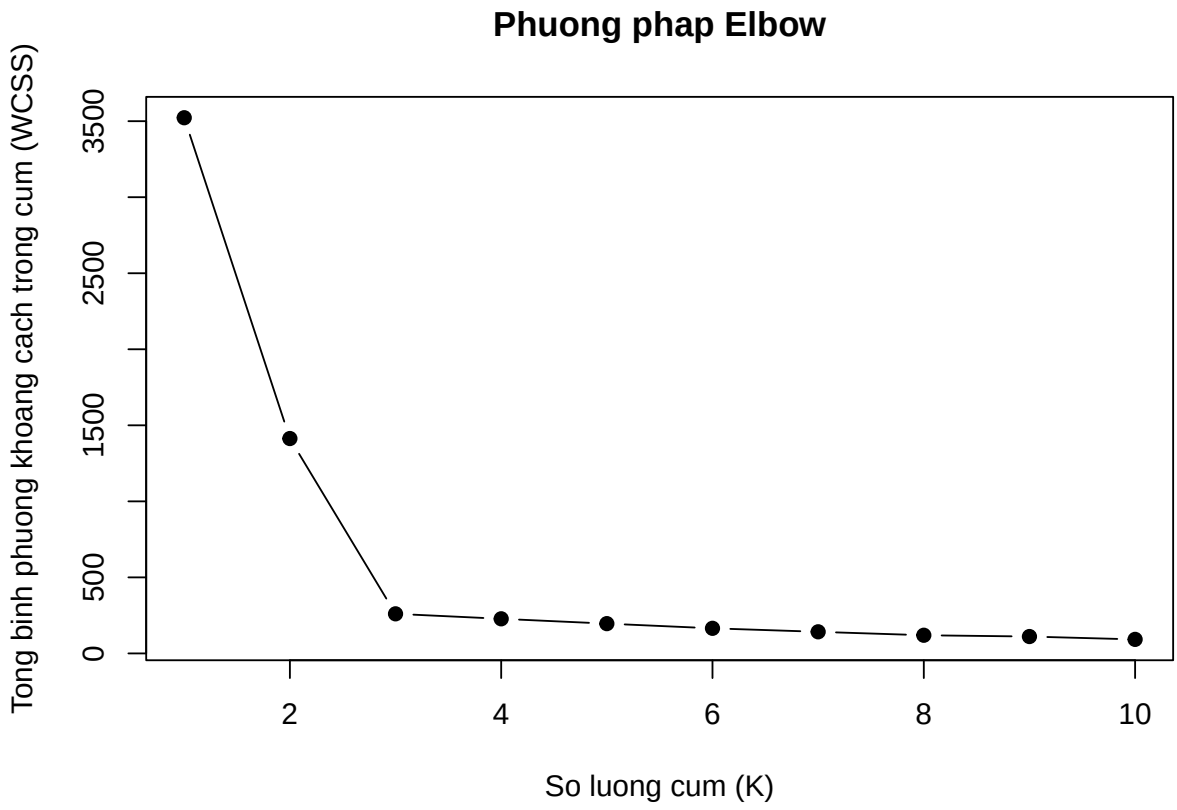


Figure 1: Bieu do Elbow cho viec xac dinh K toi uu

2.1.2 2. Phương pháp Silhouette

Phương pháp Silhouette đánh giá chất lượng phân cụm dựa trên độ gắn kết và tách biệt của các cụm. Chỉ số Silhouette đo lường mức độ tương đồng của một đối tượng với cụm của nó (gắn kết) so với các cụm khác (tách biệt).

Nguyên lý hoạt động: - Với mỗi điểm dữ liệu i, tính: - a(i): Khoảng cách trung bình từ i đến tất cả các điểm khác trong cùng cụm (gắn kết) - b(i): Khoảng cách trung bình nhỏ nhất từ i đến tất cả các điểm trong cụm khác gần nhất (tách biệt) - Hệ số Silhouette s(i) = (b(i) - a(i)) / max(a(i), b(i))

Công thức tính điểm Silhouette:

$s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$

Phạm vi giá trị và cách giải thích: - **Phạm vi**: [-1, 1] - **Gần 1**: Điểm dữ liệu được gán chính xác vào cụm của nó (rất tốt) - **Gần 0**: Điểm dữ liệu nằm ở biên giữa hai cụm (không rõ ràng) - **Gần -1**: Điểm dữ liệu có thể bị gán vào cụm sai (rất tệ)

Đánh giá chất lượng theo điểm Silhouette trung bình: - 0.71 - 1.00: Cấu trúc phân cụm mạnh - 0.51 - 0.70: Cấu trúc phân cụm hợp lý - 0.26 - 0.50: Cấu trúc phân cụm yếu, cần xem xét lại - ≤ 0.25: Không tìm thấy cấu trúc phân cụm đáng kể

Ưu điểm và nhược điểm: - **Ưu điểm**: Không phụ thuộc vào số cụm hoặc thuật toán phân cụm cụ thể - **Nhược điểm**: Tính toán phức tạp, không hiệu quả cho bộ dữ liệu lớn

Cách xác định K tối ưu: - Tính điểm Silhouette trung bình cho nhiều giá trị K khác nhau - Chọn K có điểm Silhouette trung bình cao nhất

```
# Tính điểm Silhouette trung bình cho các giá trị K từ 2 đến 10
avg_sil <- numeric(9)
for(k in 2:10) {
  km <- kmeans(synthetic_data, centers = k, nstart = 25)
  ss <- silhouette(km$cluster, dist(synthetic_data))
  avg_sil[k-1] <- mean(ss[, 3])
}

# Vẽ biểu đồ Silhouette
plot(2:10, avg_sil, type = "b", pch = 19,
     xlab = "So luong cum (K)",
     ylab = "Diem Silhouette trung binh",
     main = "Phuong phap Silhouette")
```

2.1.3 3. Phương pháp Gap Statistic

Gap Statistic là phương pháp thống kê cao cấp hơn, được giới thiệu bởi Tibshirani và đồng nghiệp năm 2001. Phương pháp này so sánh tổng biến thiên trong cụm của dữ liệu thực với dữ liệu được tạo từ phân phối đồng nhất (null reference distribution).

Nguyên lý hoạt động: - Tạo một phân phối tham chiếu bằng cách lấy mẫu ngẫu nhiên từ không gian của dữ liệu gốc - So sánh độ phân tán của dữ liệu thực với dữ liệu tham chiếu - Tìm sự khác biệt lớn nhất giữa hai phân phối này

Công thức tính Gap Statistic:

$Gap(k) = E_n^*[\log(W_k)] - \log(W_k)$

Trong đó: - W_k là tổng phương sai trong cụm của dữ liệu thực với k cụm - $E_n^*[\log(W_k)]$ là giá trị kỳ vọng của $\log(W_k)$ từ dữ liệu lấy mẫu ngẫu nhiên - n là số lần lấy mẫu

Cách xác định K tối ưu: - Tính Gap Statistic cho các giá trị K khác nhau - Chọn K nhỏ nhất sao cho:

$Gap(k) \geq Gap(k+1) - s_{k+1}$

Trong đó s_{k+1} là sai số chuẩn của $Gap(k+1)$

Phạm vi giá trị và đánh giá: - **Phạm vi**: Không có giới hạn cố định, nhưng thường dương - **Giá trị cao**: Cho thấy cấu trúc phân cụm tốt - **Chất lượng**: K tối ưu là điểm mà Gap Statistic bắt đầu đạt ngưỡng ổn định hoặc giảm

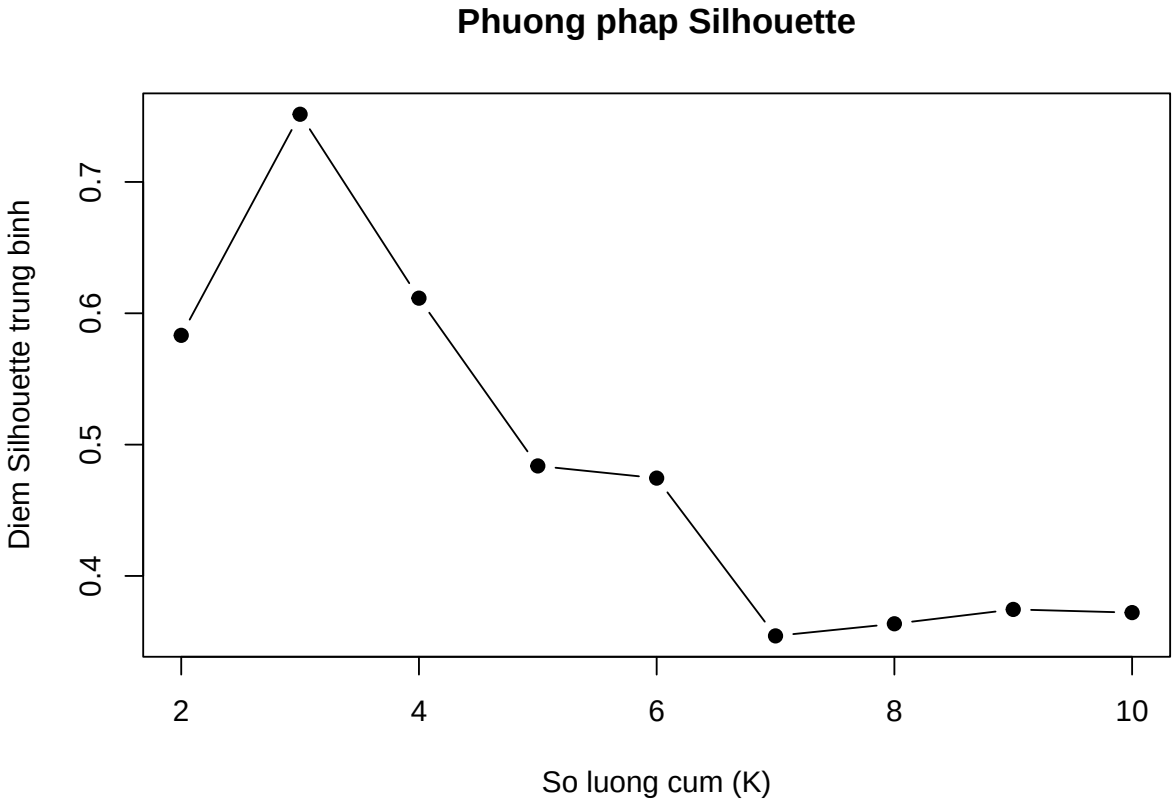
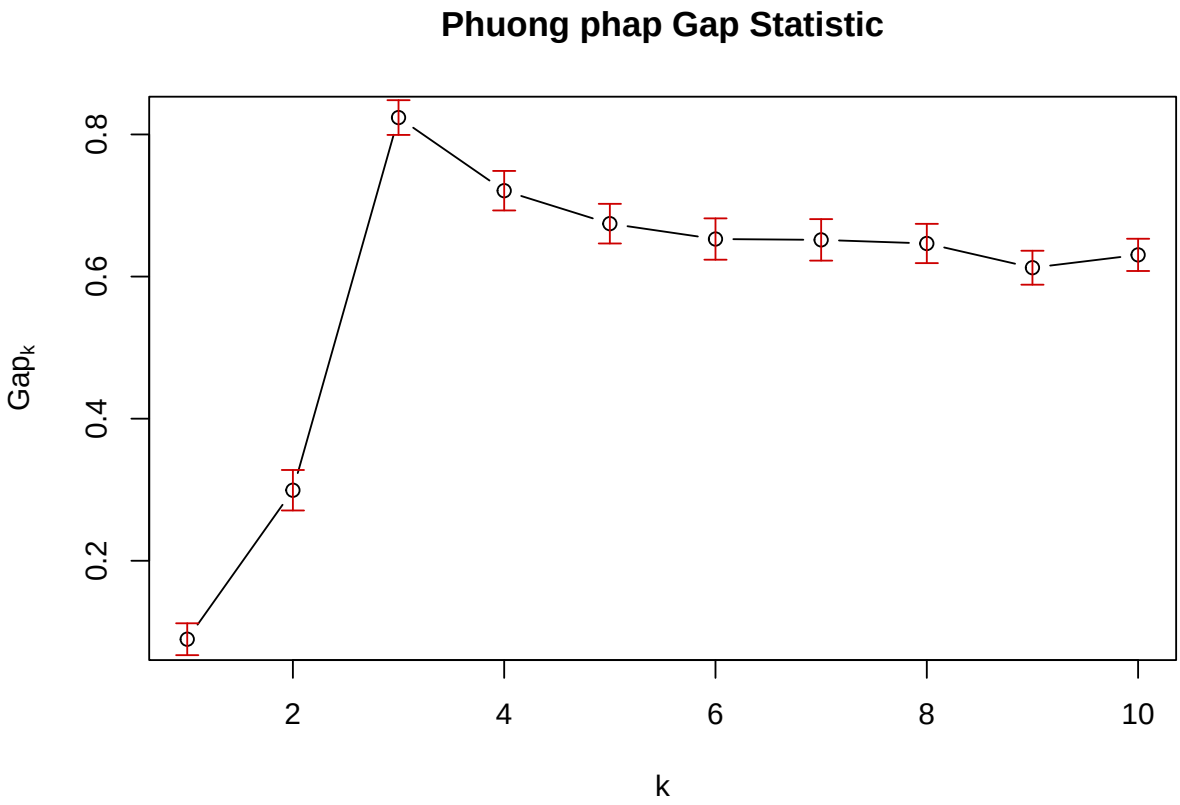


Figure 2: Bieu do Silhouette cho viec xac dinh K toi uu

Ưu điểm và nhược điểm: - **Ưu điểm**: - Có cơ sở thống kê vững chắc - Hoạt động tốt với nhiều dạng cụm khác nhau - Ít chủ quan hơn Elbow method - **Nhược điểm**: - Tính toán phức tạp và tốn thời gian - Cần nhiều lần chạy để tạo phân phối tham chiếu (thường ≥ 20 lần)

```
# Tính Gap Statistic (có thể mất thời gian)
set.seed(123)
gap_stat <- clusGap(synthetic_data, FUN = kmeans, nstart = 25,
                    K.max = 10, B = 50)
```

```
# Vẽ biểu đồ Gap Statistic
plot(gap_stat, main = "Phuong phap Gap Statistic")
```



2.1.4 4. Các phương pháp khác

Ngoài các phương pháp trên, còn có nhiều phương pháp khác để xác định số cụm K phù hợp:

- **Chỉ số Davies-Bouldin (DBI)**: Chỉ số càng thấp càng tốt
- **Chỉ số Calinski-Harabasz (CH)**: Chỉ số càng cao càng tốt
- **Phương pháp Bayesian Information Criterion (BIC)**
- **Phương pháp Cross-validation**

```
# Sử dụng NbClust để đánh giá nhiều phương pháp
# Chú ý: Kết quả này có thể mất thời gian để tính toán
nb <- NbClust(synthetic_data, distance = "euclidean", min.nc = 2,
              max.nc = 10, method = "kmeans")
```

```
## *** : The Hubert index is a graphical method of determining the number of clusters.
```

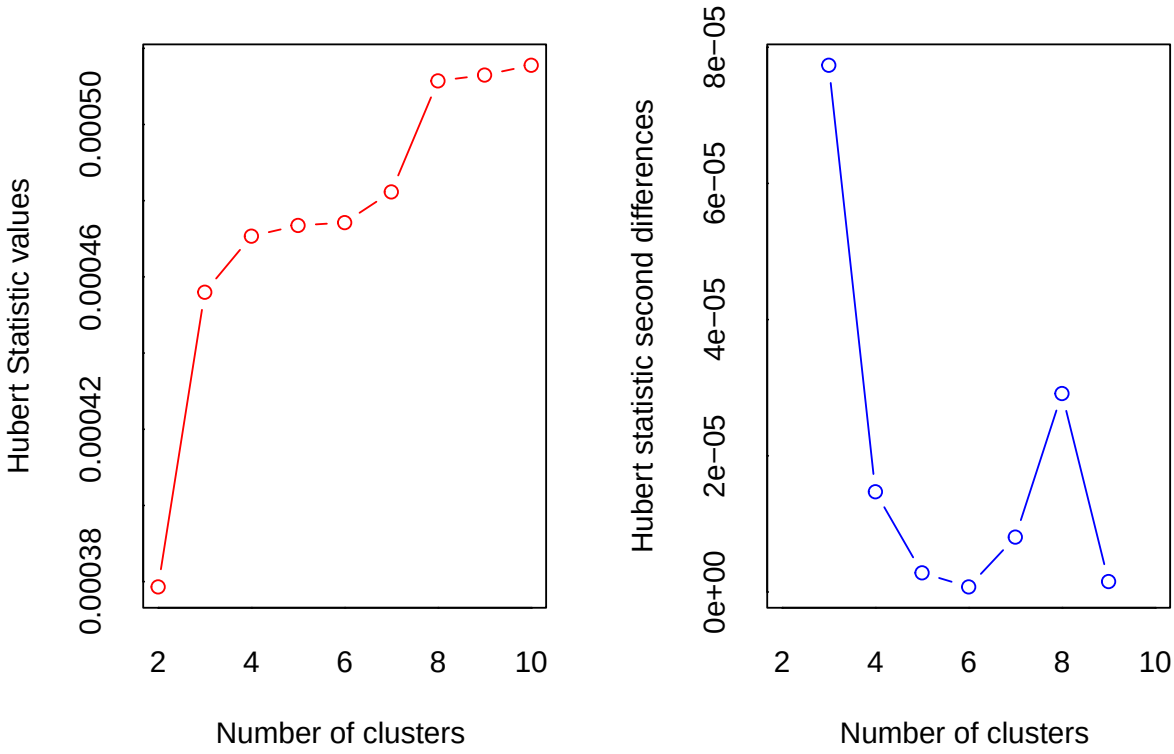


Figure 3: So sanh nhieu phuong phap xac dinh K toi uu

```
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```

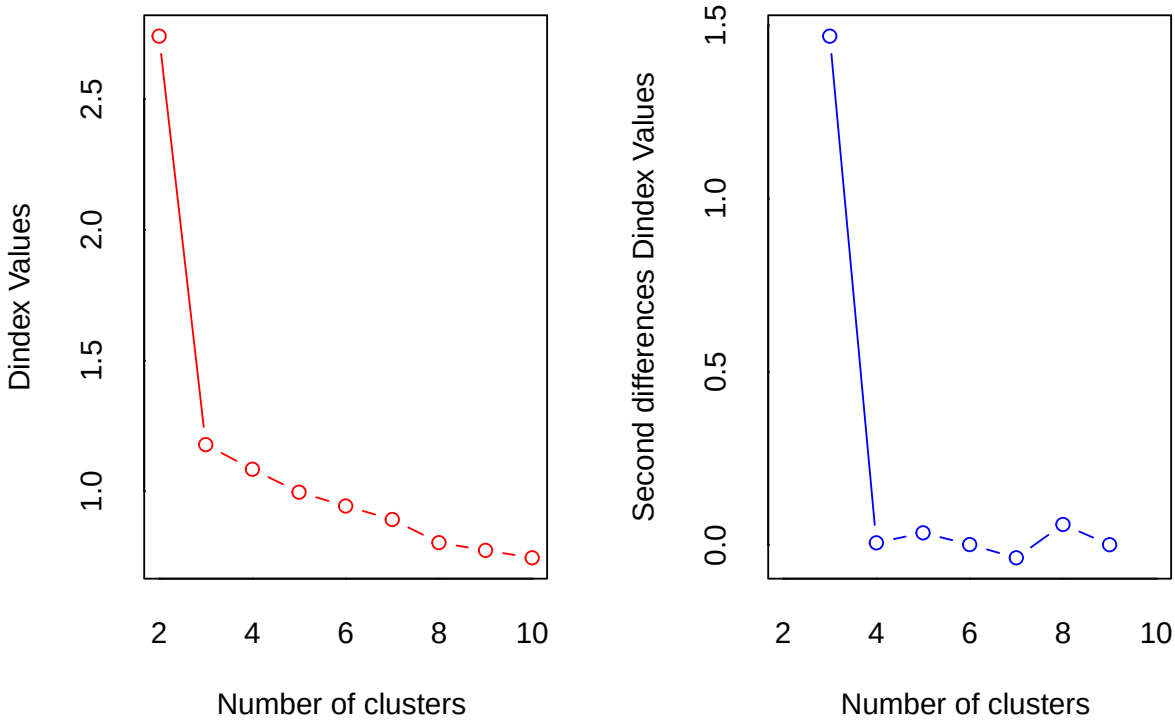
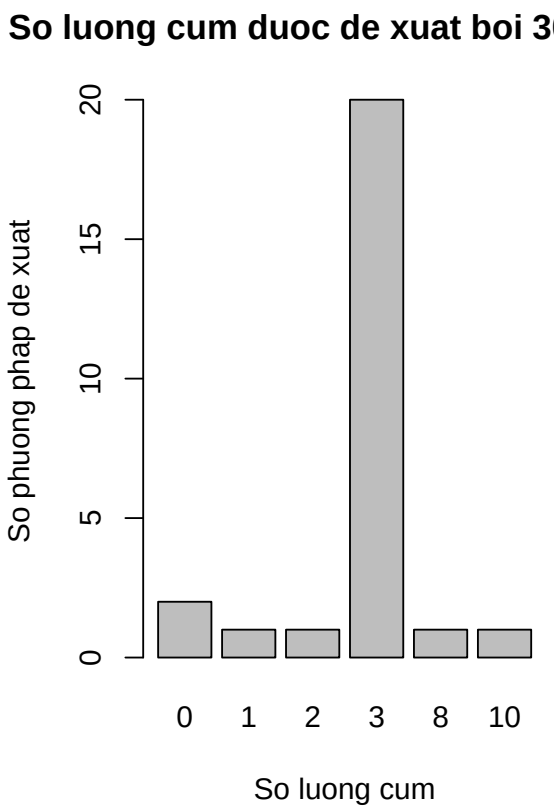


Figure 4: So sanh nhieu phuong phap xac dinh K toi uu

```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 1 proposed 2 as the best number of clusters
## * 20 proposed 3 as the best number of clusters
## * 1 proposed 8 as the best number of clusters
## * 1 proposed 10 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************
```

```r
# Hiển thị biểu đồ tần suất của các K được đề xuất
barplot(table(nb$Best.n[1,]),
        xlab = "So luong cum",
        ylab = "So phuong phap de xuat",
        main = "So luong cum duoc de xuat boi 30 chi so")
```



Đánh giá cụm thể

Để đưa ra quyết định cuối cùng về số lượng cụm K, cần kết hợp:

1. **Kết quả từ các phương pháp định lượng** (Elbow, Silhouette, Gap Statistic)
2. **Kiến thức về lĩnh vực** (domain knowledge)
3. **Mục tiêu phân tích** (exploratory vs confirmatory)
4. **Tính giải thích được** (interpretability) của kết quả

3 Đánh giá chất lượng phân cụm

Đánh giá chất lượng phân cụm là một thách thức vì không có nhãn đúng (ground truth) để so sánh. Chúng ta có thể sử dụng các phương pháp đánh giá nội tại (internal) và ngoại tại (external).

3.1 Các chỉ số đánh giá nội tại (Internal Evaluation)

Đánh giá nội tại đo lường chất lượng phân cụm dựa trên chính dữ liệu đã được phân cụm, không cần thông tin nhãn bên ngoài. Các chỉ số này thường tập trung vào hai yếu tố: - **Gắn kết (cohesion)**: Các điểm trong cùng một cụm nên gần nhau - **Tách biệt (separation)**: Các cụm khác nhau nên xa nhau

3.1.1 1. Silhouette Coefficient

Silhouette Coefficient đo lường mức độ gắn kết của các điểm trong cùng một cụm và tách biệt của các điểm giữa các cụm khác nhau.

Công thức: (Đã trình bày chi tiết ở phần trên)

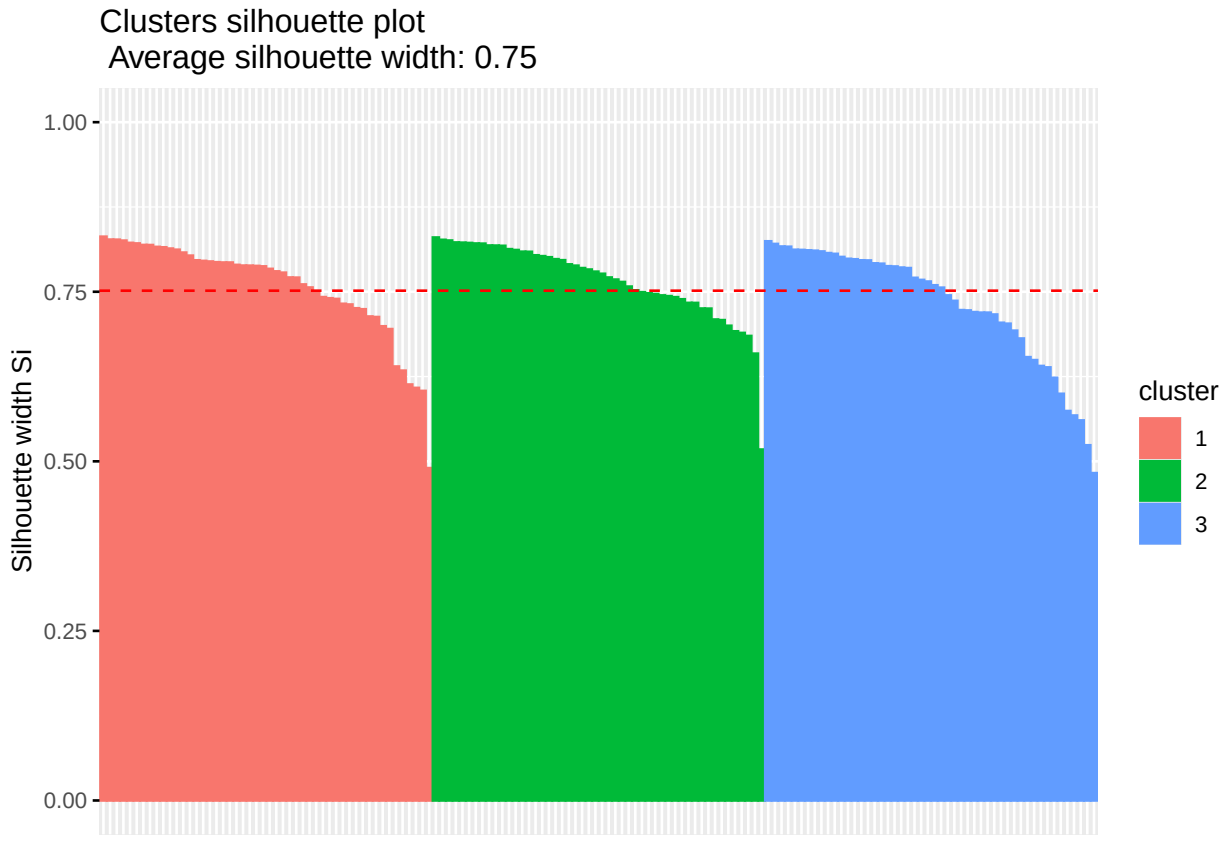
Phạm vi giá trị: [-1, 1] - **> 0.7**: Cấu trúc phân cụm rất mạnh - **0.5 - 0.7**: Cấu trúc phân cụm hợp lý - **0.25 - 0.5**: Cấu trúc phân cụm yếu, có thể giả tạo - **< 0.25**: Không tìm thấy cấu trúc phân cụm đáng kể

Ý nghĩa: Giá trị càng gần 1 càng tốt

```
# Áp dụng K-Means với K = 3
km_result <- kmeans(synthetic_data, centers = 3, nstart = 25)

# Tính và vẽ biểu đồ Silhouette
sil <- silhouette(km_result$cluster, dist(synthetic_data))
fviz_silhouette(sil, print.summary = TRUE)
```

```
##   cluster size ave.sil.width
## 1       1   50          0.76
## 2       2   50          0.77
## 3       3   50          0.73
```



2. Dunn Index

Dunn Index là tỷ lệ giữa khoảng cách nhỏ nhất giữa các cụm và đường kính lớn nhất của các cụm.

Công thức:

$DI = \frac{\min_{1 \le i < j \le k} \delta(C_i, C_j)}{\max_{1 \le m \le k} \Delta(C_m)}$

Trong đó: - $\delta(C_i, C_j)$ là khoảng cách giữa cụm C_i và C_j (khoảng cách giữa hai điểm gần nhất) - $\Delta(C_m)$ là đường kính của cụm C_m (khoảng cách giữa hai điểm xa nhất trong cụm)

Phạm vi giá trị: $(0, \infty)$ - Không có giới hạn trên cụ thể, nhưng giá trị thực tế thường < 1 - Giá trị càng lớn càng tốt

Ý nghĩa: - Chỉ số cao: Cụm gọn và tách biệt tốt - Chỉ số thấp: Cụm phân tán hoặc chồng lấn nhau

Hạn chế: - Rất nhạy cảm với dữ liệu ngoại lai (outliers) - Tính toán phức tạp đối với bộ dữ liệu lớn

```
# Tính Dunn Index
library(clValid)
dunn_index <- dunn(distance = dist(synthetic_data), km_result$cluster)
cat("Dunn Index:", dunn_index, "\n")
```

```
## Dunn Index: 0.5567089
```

3.2 Trực quan hóa kết quả phân cụm

Trực quan hóa là một phương pháp quan trọng để đánh giá chất lượng phân cụm.

```
# Trực quan hóa kết quả phân cụm
fviz_cluster(km_result, data = synthetic_data,
             palette = c("#1B9E77", "#D95F02", "#7570B3"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_minimal(),
             main = "Ket qua phan cum K-Means (K=3)")
```

References

Hannun, Awni, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, et al. 2014. “Deep Speech: Scaling up End-to-End Speech Recognition.” arXiv. https://doi.org/10.48550/arXiv.1412.5567.

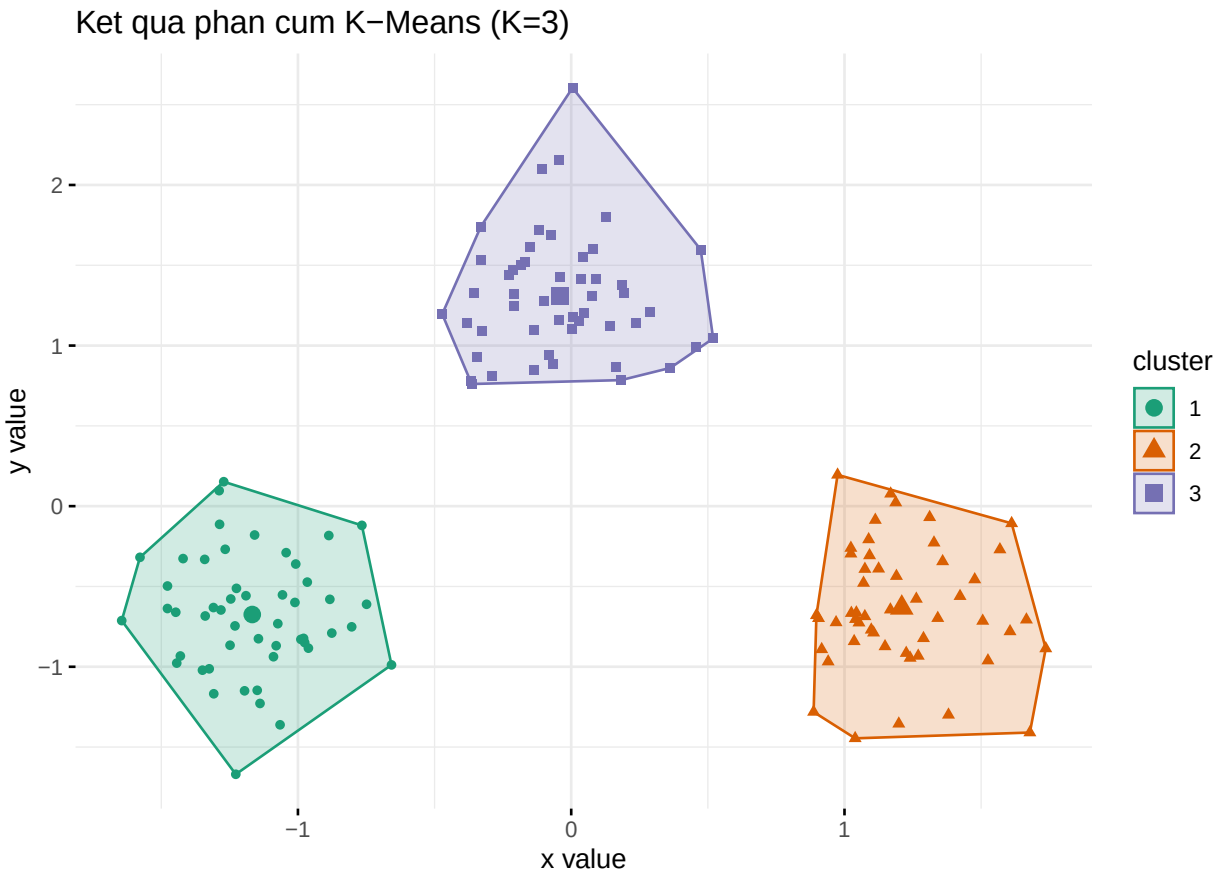


Figure 5: Truc quan hoa ket qua phan cum K-Means