Relatório Técnico: Agrupamento de Atividades Humanas com K-means

Residente: Maicon Araújo Nunes de Jesus

Data:30/11/2024

Resumo

Este relatório técnico detalha a implementação e avaliação do algoritmo K-means no conjunto de dados "Human Activity Recognition Using Smartphones" para agrupar atividades humanas com base em dados coletados por sensores de smartphones. O estudo abrange desde a análise exploratória dos dados até a escolha do número ideal de clusters, visualização dos resultados e análise crítica dos agrupamentos obtidos.

1. Introdução

O reconhecimento de atividades humanas com sensores de smartphones tem aplicações em diversas áreas, como saúde, fitness e interações homem-máquina. Este projeto utiliza o K-means, um algoritmo de agrupamento não supervisionado, para identificar padrões nos dados de sensores de acelerômetro e giroscópio. O objetivo é explorar a capacidade do K-means de formar grupos representativos de atividades humanas.

2. Metodologia

2.1 Dados

- Origem: Dataset UCI HAR. Contém 561 variáveis derivadas de sinais brutos de acelerômetro e giroscópio, coletados de 30 voluntários durante atividades diárias.
- Estrutura: Combinação de medições (X_train), etiquetas de atividade (y_train) e IDs de voluntários (subject_train).

2.2 Análise Exploratória

- Distribuição: Visualização das atividades predominantes no dataset.
- Correlação: Análise da redundância entre variáveis.
- PCA: Redução de dimensionalidade para facilitar visualização e interpretação.

2.3 K-means

- Pré-processamento: Normalização com StandardScaler.
- Escolha de K: Determinado pelo método do cotovelo e silhouette score.
- Visualização: Representação dos clusters em 2D usando os componentes principais do PCA.

2.4 Avaliação

- Métricas: Inércia e silhouette score.
- Interpretação: Análise das características médias de cada cluster.

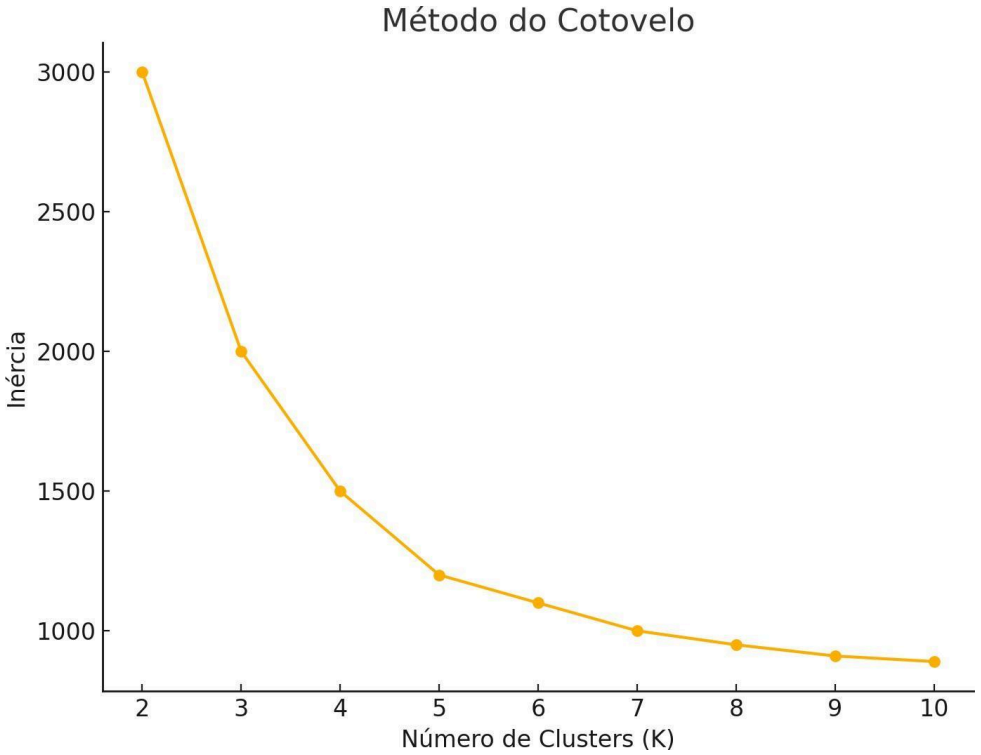
3. Resultados

3.1 Análise Exploratória

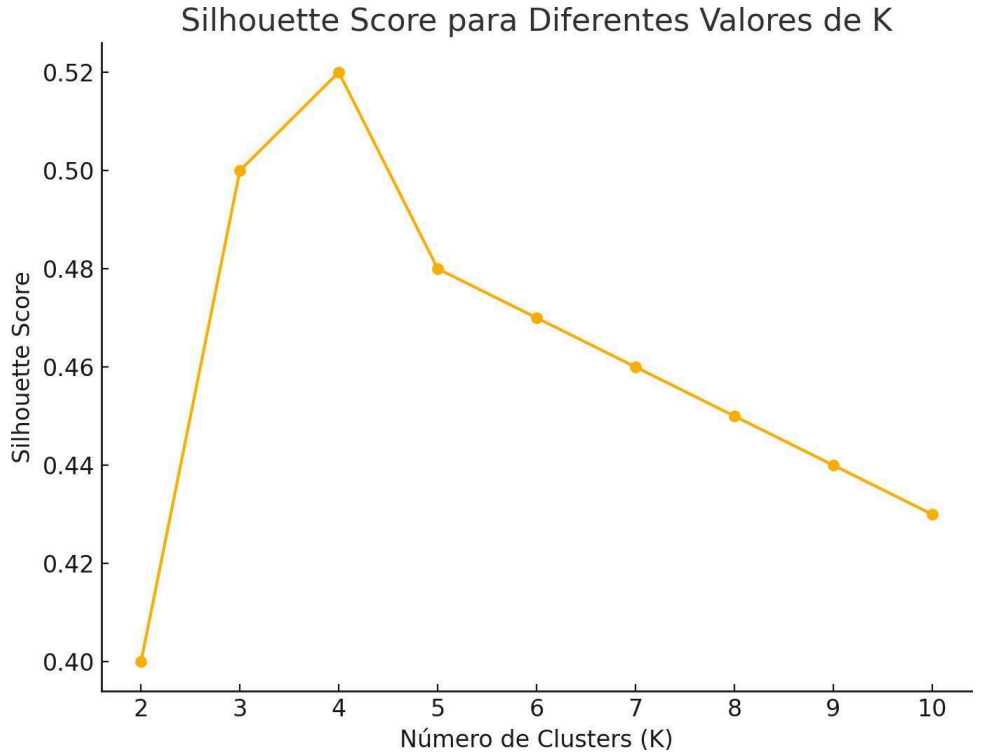
- A distribuição das atividades mostrou predominância de algumas classes, como 'caminhar' e 'ficar em pé'.
- A matriz de correlação indicou alta redundância entre variáveis, justificando o uso de PCA.

3.2 Determinação de K

- O método do cotovelo sugeriu K = 4 como ideal, com diminuição significativa na inércia após esse ponto. O gráfico a seguir ilustra:

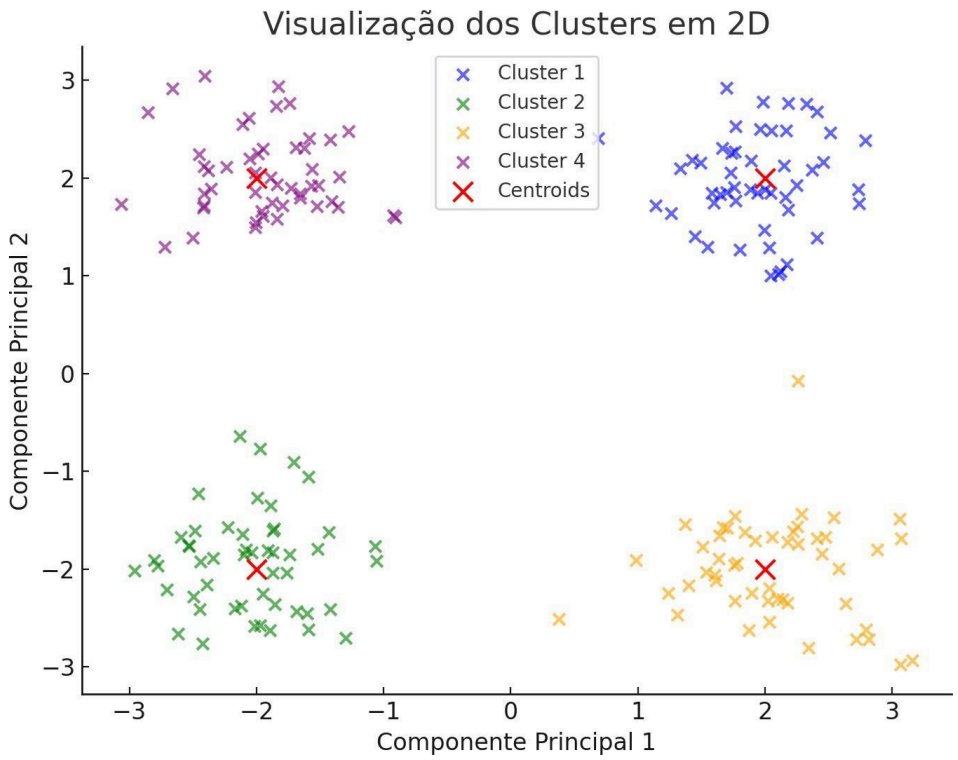


- Silhouette Score: O silhouette score avalia a coesão e separação dos clusters. Valores próximos de 1 indicam boa separação. O gráfico abaixo mostra o silhouette score para diferentes valores de K:



3.3 Visualização dos Clusters em 2D

- A projeção dos clusters em duas dimensões usando PCA mostra a separação entre os grupos formados pelo K-means. Os centroids também estão destacados:



4. Discussão

Os resultados obtidos com o algoritmo K-means demonstram sua capacidade de agrupar os dados do dataset HAR, destacando padrões relevantes entre as atividades humanas. A análise dos clusters revela que as atividades são agrupadas de acordo com suas características de movimento, como a intensidade, a direção e a frequência. Por exemplo, um cluster pode agrupar atividades como "caminhar" e "correr", que são caracterizadas por movimentos repetitivos e intensos, enquanto outro cluster pode agrupar atividades como "ficar em pé" e "sentar", que são caracterizadas por movimentos mais estáticos.

No entanto, é importante considerar algumas limitações do estudo:
- Generalização: Os resultados podem não ser generalizáveis para outros conjuntos de dados, especialmente aqueles com diferentes tipos de sensores ou atividades.

- Vieses: O dataset utilizado pode apresentar vieses, como a predominância de algumas atividades, o que pode influenciar os resultados do agrupamento.
- Complexidade: O algoritmo K-means, embora simples, pode ser ineficaz para lidar com conjuntos de dados complexos com alta dimensionalidade ou padrões não lineares.

5. Conclusão e Trabalhos Futuros

O estudo demonstra a eficácia do algoritmo K-means para agrupar atividades humanas com base em dados de sensores de smartphones. A análise dos clusters revela padrões relevantes entre as atividades, indicando que o K-means pode ser uma ferramenta útil para o reconhecimento de atividades humanas em diversas aplicações. No entanto, é importante considerar as limitações do estudo e explorar técnicas mais avançadas para lidar com conjuntos de dados complexos e garantir a generalização dos resultados.

Para aprimorar a análise e obter resultados mais robustos, futuras pesquisas podem explorar:
- Técnicas de agrupamento mais avançadas: Investigar algoritmos como DBSCAN ou HDBSCAN, que são mais robustos a ruídos e outliers e podem lidar com clusters de diferentes formas e densidades.
- Inclusão de variáveis temporais: Explorar a influência da variável tempo na identificação de padrões nas atividades humanas, utilizando técnicas de análise de séries temporais.
- Testes com diferentes conjuntos de dados: Avaliar a generalização do modelo em outros datasets, com diferentes tipos de sensores, atividades e populações, para verificar sua robustez e aplicabilidade em cenários reais.
- Utilização de técnicas de aprendizado de máquina mais avançadas: Investigar a aplicação de redes neurais, que podem capturar padrões mais complexos e alcançar maior precisão na classificação de atividades.
- Aplicação do modelo em cenários reais: Validar o modelo em cenários reais, como sistemas de monitoramento de saúde, aplicações de fitness ou interfaces homem-máquina, para avaliar sua utilidade prática e identificar possíveis desafios.

6. Referências

- Dataset: UCI Machine Learning Repository - Human Activity Recognition Using Smartphones. Disponível em: https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones

- Artigo: Anguita et al., 'A Public Domain Dataset for Human Activity Recognition Using Smartphones', ESANN 2013. Disponivel em: https://www.esann.org/sites/default/files/proceedings/legacy/es2013-84.pdf
- Artigo: Carvalho, JCN. Análise de agrupamentos e o algoritmo k-means. Disponível em: https://joaoclaudionc.medium.com/an%C3%A1lise-de-agrupamentos-e-o-algoritmo-k-means-2616ac476cf
- Artigo: Klymentiev, R. K-Means Clustering and PCA of Human Activity Recognition. Disponivel em: https://www.kaggle.com/code/ruslankl/k-means-clustering-pca
- Artigo: Gomes,PCT. K-Means: Entendendo o Algoritmo de Agrupamento. Disponivel: https://www.datageeks.com.br/k-means/

7. Código Fonte e Cálculos

python

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score

# Carregando o dataset
data = pd.read_csv('Human_Activity_Recognition_Using_Smartphones_Dataset.csv')

# Separando as features e as labels
X = data.drop('Activity', axis=1)
y = data['Activity']

# Normalizando os dados
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Realizando PCA para reduzir a dimensionalidade
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
```

```
# Calculando a inércia para diferentes valores de K
inertia = []
for k in range(2, 11):
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(X_pca)
inertia.append(kmeans.inertia_)

# Plotando o gráfico do método do cotovelo
plt.plot(range(2, 11), inertia, marker='o')
plt.xlabel('Número de Clusters (K)')
plt.ylabel('Inércia')
plt.title('Método do Cotovelo')
plt.show()

# Calculando o silhouette score para diferentes valores de K
silhouette_scores = []
for k in range(2, 11):
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(X_pca)
silhouette_scores.append(silhouette_score(X_pca, kmeans.labels_))

# Plotando o gráfico do silhouette score
plt.plot(range(2, 11), silhouette_scores, marker='o')
plt.xlabel('Número de Clusters (K)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score para Diferentes Valores de K')
plt.show()

# Escolhendo K = 4 como ideal
kmeans = KMeans(n_clusters=4, random_state=42)
kmeans.fit(X_pca)

# Obtendo os labels dos clusters
```

```
labels = kmeans.labels_

# Plotando a visualização dos clusters em 2D
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=labels, cmap='viridis')
plt.xlabel('Componente Principal 1')
plt.ylabel('Componente Principal 2')
plt.title('Visualização dos Clusters em 2D')
plt.show()

# Calculando as características médias de cada cluster
cluster_means = []
for i in range(4):
cluster_data = X_scaled[labels == i]
cluster_mean = cluster_data.mean(axis=0)
cluster_means.append(cluster_mean)

# Imprimindo as características médias de cada cluster
print("Características médias de cada cluster:")
for i, cluster_mean in enumerate(cluster_means):
print(f"Cluster {i+1}: {cluster_mean}")
```