Algoritmos Voraces (Greedy)

Utilizar un enfoque Greedy para comparar cada curva con las demás en el conjunto, seleccionando características clave que puedan determinar la similitud y comparando de forma secuencial.

Ventajas

1. Puede ser muy rápido si se seleccionan las características adecuadas para comparar las curvas
2. Es relativamente fácil de implementar, especialmente si los criterios de similitud son claros

Desventajas

1. Puede quedarse atrapado en soluciones locales, especialmente si los criterios de similitud no están bien definidos
2. La solución no garantiza una comparación completa si se emplean heurísticas simples

Dividir y Conquistar

Dividir el conjunto de curvas en subconjuntos y comparar las curvas dentro de cada subconjunto, para luego combinar los resultados para clasificar todas las curvas

Ventajas

1. Puede ser eficiente en memoria ya que no necesita almacenar todas las comparaciones simultáneas
2. La comparación en subconjuntos puede ser realizada en paralelo, lo que podría aumentar la velocidad

Desventajas

1. La implementación podría ser más compleja en la fase de combinación de resultados
2. Si la división no está bien diseñada, podría no mejorar significativamente la eficiencia

Algoritmos de Aproximación

Desarrollar una métrica de similitud aproximada para comparar las curvas y aplicar esta métrica para clasificar las curvas en grupos similares o diferentes

Ventajas

3. Puede ser muy rápido si la métrica de aproximación está bien diseñada
4. Puede manejar un gran número de curvas ya que no necesita una comparación exhaustiva

Desventajas

3. La aproximación puede llevar a errores en la clasificación, especialmente si la métrica no refleja adecuadamente la similitud en forma de las curvas

4. La elección de una métrica adecuada podría ser desafiante y requerir conocimientos profundos sobre las curvas de Bézier

Ranking

1. Algoritmos de Aproximación
2. Algoritmos Greedy
3. Dividir y Conquistar