Clustering en anillos con ruido

Inteligencia Artificial (IS) 2019/20 – Propuesta de trabajo

Agustín Riscos Núñez

1. Introducción y objetivos

El tema del trabajo está inspirado en un subproblema real del experimento LHCb en el CERN, aunque obviamente lo adaptaremos simplificándolo bastante. Los detectores RICH (Ring Imaging CHerenkov) tratan de identificar patrones con forma de circunferencia o elipse que son el resultado de registrar el impacto de la radiación de Cherenkov: conos de fotones emitidos por partículas provenientes de las colisiones de hadrones cuando estas partículas atraviesan medios gaseosos especialmente acondicionados para ello a velocidades mayores que la de la luz en dichos medios.

Sin embargo, en este trabajo no se hará uso de ningún detalle físico, simplificando muchas restricciones adicionales. Informalmente, podemos plantear nuestro problema como sigue: Tenemos un conjunto de puntos en un cierto área (por ejemplo, un cuadrado 100x100) y queremos identificar de la forma más fiable posible un conjunto de "anillos" o circunferencias tales que todos (o casi todos) los puntos pertenezcan a alguna de las circunferencias, asumiendo que habrá algo de ruido.

El **objetivo principal** de esta propuesta es diseñar e implementar un algoritmo de clustering con incertidumbre adaptado para el caso en el que el objetivo es buscar grupos de puntos que formen circunferencias. Es decir, para cada grupo habrá que identificar el centro y el radio de su "circunferencia representativa", pero asumiremos que los puntos pueden encontrarse algo dispersos en un entorno relativamente próximo a dicha circunferencia. Para incorporar esa incertidumbre al modelo, en lugar de asignar los puntos a un solo cluster, cada punto estará vinculado a todas las circunferencias, aunque con distinto "grado de pertenencia".

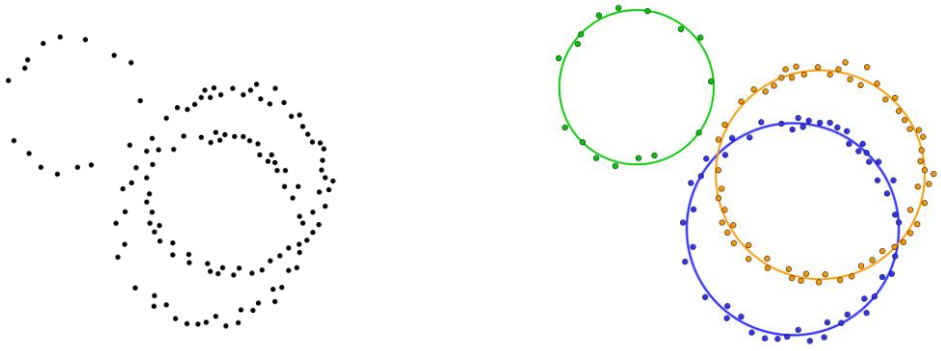

Fig. 1. Ejemplo de datos de entrada (izquierda) y su clasificación (derecha)

El algoritmo diseñado deberá seguir el siguiente **esquema general**:

1. Inicialización de circunferencias iniciales (centro y radio de cada una)
2. Repetir (hasta condición de parada)
   2.1. actualizar grados de pertenencia de los puntos a las circunferencias
   2.2. actualizar centros y radios
3. Asignar cada punto únicamente a su cluster de mayor grado de pertenencia, y devolver la información completa para cada cluster (centro, radio y lista de puntos asignados a él).

Pseudocódigo 1. Esquema general del algoritmo de clustering que hay que diseñar e implementar

Observaciones sobre el esquema general:

1. En cuanto a la inicialización, se podrá elegir entre fijar a priori unos valores (versión básica) o bien implementar algún método heurístico/aleatorio (versión avanzada).
2. En cuanto al criterio de parada, se podrá elegir entre un número fijo de iteraciones (versión básica) o bien comprobar que los clusters se han estabilizado, quedándose fijos o con variaciones muy pequeñas (versión avanzada).
3. Respecto a la presentación de datos final, se podrá elegir entre devolver el resultado del algoritmo directamente (versión básica) o bien hacer un procesado que (a) descarte puntos con bajo grado de pertenencia a todos los clusters, considerándolos "ruido" y/o (b) unifique clusters si se detecta que representan circunferencias casi idénticas (versión avanzada).

Objetivos específicos:

1. Diseñar e implementar una estructura de datos en Python para almacenar la información actualizada en cada vuelta del bucle descrito en el algoritmo principal (centros y radios de los clusters y grado de pertenencia de cada punto a cada cluster).
   Deberán proporcionarse los detalles del diseño en la memoria que acompaña al código.
2. Diseñar e implementar un método que **dado un conjunto de puntos** que suponemos colocados alrededor de una sola trayectoria circular, **devuelva** una estimación lo más verosímil posible de **centro y radio** de dicha trayectoria. Deberá explicarse en la memoria el método elegido, tanto si es original como si está sacado de alguna referencia bibliográfica o web. Podrá elegirse entre métodos globales que hagan operaciones con todos los puntos o bien métodos locales que por ejemplo se basen en aproximaciones obtenidas extrayendo tres puntos (preferiblemente alejados entre sí) y calculando la circunferencia que pasa por esos tres puntos.

   Está permitido trabajar bajo la hipótesis de que los puntos están repartidos de forma más o menos uniforme a lo largo de toda la trayectoria de la circunferencia que se está estimando (versión básica), pero se espera que se tenga en cuenta también el caso en que los puntos puedan estar concentrados solo en un arco de circunferencia (versión avanzada).

   Para justificar experimentalmente la idoneidad del método elegido, se proporcionan varios ejemplos con respuesta conocida al final de este documento.

   Deberán explicarse los detalles del método elegido en la memoria que acompaña al código, mencionando explícitamente los resultados obtenidos en las pruebas realizadas sobre los ejemplos proporcionados.
3. Diseñar e implementar una función que **dado un punto P** y un conjunto de clusters **calcule los grados de pertenencia** del punto a cada cluster, debiendo cumplir dos condiciones: (a) que para cada cluster, el valor del grado de pertenencia sea inversamente proporcional a la distancia* a dicho cluster, y (b) que para cada punto se normalicen los valores de forma que la suma de sus grados de pertenencia a todos los clusters sea 1.

   * Nota: la distancia de un punto P a una circunferencia c de centro O y radio r es

   $$d(P,c) = |d(P,O) - r|$$
4. Diseñar e implementar una modificación del método diseñado en el objetivo 2, para que funcione sobre el conjunto total de puntos, de forma que **para calcular** el nuevo valor del **centro y el radio** de un cluster **se tenga en cuenta el grado de pertenencia** de cada punto asociado a dicho cluster.
5. Diseñar e implementar una herramienta completa para resolver problemas de búsqueda de anillos mediante técnica de clustering con incertidumbre, de acuerdo con el pseudocódigo general presentado anteriormente.

6. Documentar el trabajo en un fichero con formato de artículo científico, explicando con precisión las decisiones de diseño en la representación del problema, y los resultados obtenidos en las pruebas de evaluación.
7. Realizar una presentación de los resultados obtenidos en la defensa del trabajo.

Para que el trabajo pueda ser evaluado, se deben satisfacer TODOS los objetivos específicos al completo: el trabajo debe ser original, estar correctamente implementado y funcionar perfectamente, los experimentos se deben haber llevado a cabo y analizados razonadamente, el documento debe ser completo y contener al menos 6 páginas (sin contar posibles anexos), y se debe realizar la defensa con una presentación de los resultados obtenidos.

2. Descripción del trabajo

A continuación se introduce la metodología a seguir para el correcto desarrollo del trabajo.

2.1. Implementación de los algoritmos

Tanto la estructura de datos elegida para la representación y almacenamiento de los datos como las funciones y algoritmos que se pide diseñar e implementar serán preferiblemente originales, pero podrá estar basada en estructuras similares utilizadas en librerías de Python disponibles en repositorios reconocidos (que deberán en su caso citarse convenientemente).

El código no debe diseñarse de forma específica para unos datos concretos, sino que debe hacerse de forma paramétrica y flexible, que permita realizar nuevos experimentos fácilmente con otros datos y ajustes diferentes (distinto rango de valores para las coordenadas de los puntos, distinto número de clusters, etc).

El conjunto de puntos que se recibe de entrada se proporcionará a través de un fichero aparte, que puede ser csv, o tener las restricciones de formato que se deseen, siempre que se aporten las indicaciones necesarias en un fichero README.txt (ver apartado 3).

2.2. Experimentación

La experimentación se debería plantear como una batería de pruebas con distintas configuraciones ejecutadas sobre ejemplos con solución conocida: anillos separados, anillos concéntricos, anillos cercanos con intersecciones múltiples, etc, analizando y comparando los rendimientos obtenidos lanzando múltiples ejecuciones del algoritmo:

- repitiendo varias veces la ejecución para cada configuración (ya que si se eligen los clusters iniciales aleatoriamente eso puede afectar al resultado)
- probando para cada ejemplo con distinto número de clusters iniciales,
- añadiendo para cada ejemplo distintos niveles de ruido (puntos adicionales de fondo que no pertenecen a ningún cluster).
- …

Se espera que se incluya en la memoria un análisis comparativo de los resultados obtenidos para cada experimento, respecto a algunas métricas numéricas (por ejemplo, tiempo de ejecución de cada experimento, número de iteraciones realizadas hasta lograr la convergencia, suma de las distancias de los puntos a la circunferencia para cada cluster, número de puntos clasificados como ruido, etc). Es preferible la presentación de los resultados de forma agrupada en tablas, no limitándose a copiar en el documento todas las salidas obtenidas. También se valorará la inclusión de una presentación gráfica de los resultados como aportación de mejora adicional, pero no es un requisito.

2.3. Documentación

En la página web de la asignatura se pueden encontrar plantillas donde se sugiere una estructura y formato general para la memoria que hay que presentar, en distintos formatos (.tex, .odt y .doc). Estas plantillas han sido elaboradas a partir del formato oficial de los *IEEE conference proceedings* [1]. El documento entregado deberá estar en formato PDF.

El artículo deberá tener una extensión mínima de 6 páginas, y la estructura general del documento debe ser como sigue: en primer lugar realizar una introducción al trabajo explicando el objetivo fundamental, incluyendo un breve repaso de antecedentes en relación con la temática del trabajo y con los métodos empleados (mencionar referencias bibliográficas), a continuación describir la estructura del trabajo, las decisiones de diseño que se hayan tomado a lo largo de la elaboración del mismo (mencionar también elementos que se consideraron inicialmente pero que posteriormente fueron descartados, si los hubiera), y la metodología seguida al implementarlo (nunca poner código, pero sí pseudocódigo), y seguidamente detallar los experimentos llevados a cabo, analizando los resultados obtenidos. Por último, el documento debe incluir una sección de conclusiones, y una bibliografía donde aparezcan no sólo las referencias citadas en la sección de introducción, sino cualquier documento consultado durante la realización del trabajo (incluidas las referencias web a páginas o repositorios).

2.4. Mejoras

Aunque no sea obligatorio, sí se tendrá en cuenta en la calificación la incorporación de un interfaz gráfico o menú amigable que facilite la experimentación o la presentación de resultados. También podrán recibir puntuación adicional otras mejoras o añadidos que se incorporen al trabajo más allá de los requisitos mínimos que se mencionan en la lista de objetivos específicos.

Por ejemplo: implementar varios métodos distintos para calcular el nuevo valor del centro y el radio de un cluster y comparar su eficacia, diseñar e implementar un generador automático de ejemplos, que dado un conjunto de circunferencias genere un conjunto de puntos distribuidos aproximadamente sobre esas trayectorias, etc

2.5. Presentación y defensa

El día de la defensa se deberá realizar una pequeña presentación (PDF, PowerPoint o similar) de 10 minutos en la que participarán activamente todos los miembros del grupo que ha desarrollado el trabajo. Esta presentación seguirá a grandes rasgos la misma estructura que el documento, pero se deberá hacer especial mención a los resultados obtenidos y al análisis crítico de los mismos. Se podrá usar un portátil (personal del alumno), diapositivas y/o pizarra. En los siguientes 10 minutos de la defensa, el profesor procederá a realizar preguntas sobre el trabajo, que podrán ser tanto del documento como del código fuente.

3. Criterios de evaluación

Para que el trabajo pueda ser evaluado, se deberá satisfacer los objetivos concretos descritos en el apartado 1 (todos y cada uno de ellos, al menos en su versión básica). Se entregará un fichero comprimido .zip, que contenga:

- **Una carpeta con el código fuente**. Dentro de dicha carpeta tiene que haber un fichero README.txt, que resuma la estructura del código fuente, e indique cómo usar la interfaz (si se ha implementado), o al menos cómo hacer pruebas con las funciones implementadas, incluyendo ejemplos de uso. Asimismo se deberá indicar cómo reproducir los experimentos realizados. Es importante la coherencia de este fichero con la defensa.

- **El documento – artículo en formato PDF**. Deberá tener una extensión mínima de 6 páginas. Deberá incluir toda la bibliografía consultada (libros, artículos, technical reports, páginas web, códigos fuente, diapositivas, etc.) en el apartado de referencias, y mencionarlas a lo largo del documento.

Para la evaluación se tendrá en cuenta el siguiente criterio de valoración, considerando que la puntuación máxima corresponde a la *versión avanzada* (si se opta por la *versión básica* descrita en el apartado 1, las puntuaciones de los tres primeros apartados se verán reducidas en 0.25 cada una):

- **El código fuente (0.75 puntos)**: se valorará la claridad y buen estilo de programación, corrección y eficiencia de la implementación, y calidad de los comentarios. La claridad del fichero README.txt también se valorará. En ningún caso se evaluará un trabajo con código copiado directamente de otros compañeros o de internet.

  En este trabajo está permitido utilizar en parte librerías de Python existentes, aunque en este apartado se valorará exclusivamente el código original desarrollado por los alumnos.

- **Usabilidad y experimentación (0.75 puntos)**: se valorará la cantidad de experimentos realizados, y la calidad e interés de los datos utilizados: tamaño y complejidad de los ejemplos, variedad de comprobaciones ejecutadas (modificando parámetros de inicialización y/o criterio de parada, etc), y la claridad con la que se presenten y analicen los resultados obtenidos en la memoria (estadísticas, tablas y/o gráficas). También se valorará la usabilidad, a la hora de reproducir los mismos experimentos u otros nuevos, tanto en lo relativo al input (por ejemplo, facilidad de introducir instancia y parámetros), como al output (opción de ofrecer no sólo resultado final, sino detalles de los pasos intermedios del bucle).

- **El documento – artículo científico (0.75 puntos)**:
  - Se valorará el uso adecuado del lenguaje y el estilo general del documento (por ejemplo, el uso de la plantilla sugerida).
  - Se valorará el trabajo previo de investigación realizado, siempre que quede claro que se ha entendido lo estudiado.
  - Se valorará en general la claridad de las explicaciones, el razonamiento de las decisiones, y especialmente el análisis y presentación de resultados en las secciones de experimentación y conclusiones.
  - Igualmente, no se evaluará el trabajo si se detecta cualquier copia del contenido del documento. La sección de referencias deberá incluir menciones a todas las pertinentes fuentes consultadas.

- **La presentación y defensa (0.75 puntos)**: se valorará la claridad de la presentación y la buena explicación de los contenidos del trabajo, así como, especialmente, las respuestas a las preguntas realizadas por el profesor.

- **Mejoras**: Se valorarán hasta con 0.5 puntos extra sin superar el máximo de 3 puntos totales del trabajo.

IMPORTANTE: Cualquier **plagio**, **compartición de código** o uso de material que no sea original y del que no se cite convenientemente la fuente, significará automáticamente la **calificación de cero** en la asignatura para **todos los alumnos involucrados**. Por tanto, a estos alumnos **no se les conserva**, ni para la actual ni para futuras convocatorias, **ninguna nota** que hubiesen obtenido hasta el momento. Todo ello sin perjuicio de las correspondientes **medidas disciplinarias** que se pudieran tomar.

4. Referencias

[1] Plantilla IEEE. https://www.ieee.org/conferences_events/conferences/publishing/templates.html