

Trabajo de Fin de Grado en Física

Aplicación de algoritmos de agrupamiento cuántico para la detección de patrones en conjuntos de datos no supervisados

Junio, 2024

Alumno: Jorge Campos Martí

Tutor: José D. Martín-Guerrero

Resumen / Abstract

En este trabajo, estudiaremos la relación entre aprendizaje automático y física centrándonos en un algoritmo de agrupamiento inspirado en física cuántica, conocido como agrupamiento cuántico probabilístico, un método eficaz para detectar grupos en conjuntos de datos con densidades mixtas. Analizaremos sus características y lo compararemos con otros dos algoritmos populares: K-medias y DBSCAN. A su vez, aplicaremos estos algoritmos a dos problemas físicos: una clasificación de estrellas y una reconstrucción de trayectorias de partículas en un detector, y evaluaremos los resultados usando el índice de Rand ajustado.

Nuestros resultados muestran el buen rendimiento del agrupamiento cuántico probabilístico en ambos problemas, superando a los algoritmos K-medias y DBSCAN y aportando herramientas de gran utilidad. Estos resultados motivan la creación de algoritmos basados en física y el uso de estos en problema físicos, reforzando la importancia de la relación entre física y aprendizaje automático.

In this paper, we will study the relationship between machine learning and physics by focusing on a clustering algorithm inspired by quantum physics, known as probabilistic quantum clustering, an efficient method to detect clusters in datasets with mixed densities. We will analyze its characteristics and compare it with two other popular algorithms: K-means and DBSCAN. At the same time, we will apply these algorithms to two physics problems: a star classification and a reconstruction of particle trajectories in a detector, and evaluate the results using the adjusted Rand index.

Our results show good performance of quantum clustering in both problems, outperforming K-means and DBSCAN algorithms and providing useful tools. These results motivate the creation of physics-inspired algorithms and the use of these on physics problems, reinforcing the importance of the relationship between physics and machine learning.

Agradecimientos

A mi tutor, José D. Martín, y a Raúl Casaña; por su guía, tiempo y todo lo que he aprendido de ellos.

A mis padres, por apoyarme en todas mis decisiones desde que era pequeño.

A Mar, mi PIC.

Índice

1. Introducción	1
2. Agrupamiento cuántico probabilístico	1
2.1. Aprendizaje automático y tipos	1
2.1.1. Aprendizaje supervisado	1
2.1.2. Aprendizaje no supervisado	2
2.2. Aprendizaje automático inspirado en física	2
2.2.1. Ejemplo: Redes de Hopfield	2
2.3. Agrupamiento	4
2.3.1. K-medias	4
2.3.2. DBSCAN	6
2.4. Agrupamiento cuántico probabilístico	7
2.4.1. Ecuación de Schrödinger	8
2.4.2. Asignación de grupos	11
2.4.3. Elección de parámetros: <i>ANLL</i>	12
2.4.4. Agrupamiento jerárquico	14
2.5. Objetivos	15
3. Metodología	15
3.1. Problemas y conjuntos de datos	15
3.1.1. Diagrama HR	15
3.1.2. Seguimiento de partículas	17
3.2. Evaluación de los resultados	21
4. Resultados	22
4.1. Diagrama HR	22
4.2. Seguimiento de partículas	24
5. Conclusiones / <i>Conclusions</i>	25
Referencias	26

1. Introducción

En las últimas décadas, la importancia del aprendizaje automático ha aumentado considerablemente debido al gran número de datos al que investigadores y empresas se enfrentan, además de la constante mejora de la capacidad de computación de los ordenadores.

Aquellos campos de la física que tratan con un gran número de datos se benefician de estas herramientas de aprendizaje automático para obtener conclusiones a las que no podrían llegar con otros métodos, como la clasificación de astros o el procesamiento de imágenes en astrofísica, o el tratamiento de grandes volúmenes de datos en física de partículas.

Además, la física ha demostrado ser una base robusta y útil para la creación de modelos de aprendizaje automático; como la inspiración en mecánica estadística para modelos probabilísticos, las redes neuronales informadas por física (PINNs) o los algoritmos de optimización.

2. Agrupamiento cuántico probabilístico

En esta sección introduciremos el algoritmo de agrupamiento cuántico probabilístico, en el cual nos centraremos durante el trabajo. Para ello, antes explicaremos qué es el aprendizaje automático, sus tipos y cómo puede estar inspirado en física, además de profundizar en los algoritmos de agrupamiento.

2.1. Aprendizaje automático y tipos

El aprendizaje automático (AA), también conocido por su nombre en inglés, *machine learning (ML)*, es la disciplina dentro de las ciencias de la computación que estudia los algoritmos que permiten a un ordenador aprender a partir de datos. Es, por tanto, una rama de la inteligencia artificial (IA), cuyo propósito es construir máquinas que imiten comportamientos humanos, como realizar ciertas tareas o pensar.

Dependiendo de los datos que reciba un algoritmo de aprendizaje automático, diferenciamos dos tipos principales: aprendizaje supervisado, donde se dan datos de entrada (observaciones) y de salida (respuestas) y se busca obtener una función que los relacione; y aprendizaje no supervisado, donde solo tenemos datos de entrada y el objetivo es encontrar patrones. A este último pertenecen los algoritmos de agrupamiento, los cuales estudiaremos.

2.1.1. Aprendizaje supervisado

En el aprendizaje automático supervisado disponemos de variables de entrada con variables de salida asociadas. Podemos expresar estos datos de entrada como vectores \mathbf{x}_i , cada uno con p características $\mathbf{x} = (x_1, x_2, \dots, x_p)$, y sus respuestas asociadas como y_i (generalmente son escalares). Asumiendo que existe una relación entre los diferentes \mathbf{x}_i e y_i , podemos escribirla de forma general como

$$y_i = f(\mathbf{x}_i) + \epsilon \quad (1)$$

donde f es una función desconocida de \mathbf{x} y ϵ es el error asociado, que es independiente de \mathbf{x} y tiene media nula.

El objetivo del aprendizaje supervisado es obtener una expresión para f , es decir, obtener una relación (aproximada) entre \mathbf{x} e y . Las razones principales para obtener esta

relación son la predicción o la inferencia: por un lado, habiendo calculado f , si tenemos un nuevo conjunto de datos de entrada $\hat{\mathbf{x}}_i$ sin sus respuestas asociadas podemos usar f para hacer una predicción de las respuestas \hat{y}_i usando $\hat{y}_i = f(\hat{\mathbf{x}}_i)$; y, por otro lado, si lo que queremos es estudiar la forma de la relación entre \mathbf{x} e y , nuestro objetivo sería la inferencia.

Un ejemplo de aprendizaje supervisado muy usado en física es la regresión lineal: en base a unas medidas obtenidas \mathbf{x}_i e y_i se puede obtener una expresión sencilla de su relación, infiriendo así la relación entre esas magnitudes medidas y obteniendo información física. Por ejemplo, si tenemos un muelle y medimos su elongación Δx para diferentes fuerzas F obtendremos varios pares de datos $(\Delta x_i, F_i)$ que serán nuestras variables (\mathbf{x}_i, y_i) (en este caso, el vector \mathbf{x} solo tiene una característica, es decir, una dimensión). Un ajuste por mínimos cuadrados nos permitiría obtener la relación entre las variables $f = -k$, donde k es la constante elástica.

Desde la simple regresión lineal hasta redes neuronales complejas de aprendizaje profundo, el aprendizaje automático supervisado tiene herramientas para cubrir un gran rango de problemas.

2.1.2. Aprendizaje no supervisado

Mientras que en el aprendizaje supervisado teníamos una serie de observaciones \mathbf{x}_i a las que les correspondían unas respuestas y_i , en el aprendizaje automático no supervisado solo tenemos un conjunto de observaciones \mathbf{x}_i sin respuesta asociada. Por tanto, esta rama no tiene como objetivo la predicción, sino usar herramientas estadísticas para obtener información acerca de nuestro conjunto de observaciones.

Al igual que el aprendizaje supervisado, el aprendizaje no supervisado nos puede dar diferentes tipos de información dependiendo del problema al que nos enfrentemos. Una rama de este es el análisis de componentes principales, que estudia la reducción de las dimensiones de un conjuntos de datos con un número elevado de variables correlacionadas, lo cual puede ser útil para visualizar en dos o tres dimensiones un conjunto de datos complejo. La rama que nos interesa en este trabajo es el agrupamiento, que nos permite encontrar estructuras dentro de un conjunto de datos.

2.2. Aprendizaje automático inspirado en física

A lo largo de la historia, el conocimiento en física ha sido de gran utilidad para crear modelos y algoritmos de aprendizaje automático. Para ilustrar esto, vamos a ver un ejemplo relacionado con las redes neuronales (modelos de aprendizaje supervisado): las redes de Hopfield.

2.2.1. Ejemplo: Redes de Hopfield

Una red neuronal (en inglés: *Neural Network*, *NN*) es un modelo de aprendizaje automático que consiste en unidades (neuronas) conectadas entre ellas. Cada neurona recibe señales de las neuronas conectadas a ella, las procesa y emite un señal.

Las señales son números reales y el procesamiento de estas por una neurona consiste en una función que depende de las señales recibidas y de los pesos de cada conexión. Por ejemplo, si tenemos dos neuronas (a y b) conectadas a otra (c), la señal que producirá esta última vendrá dada por $v_c = f(w_{a,c}, v_a, w_{b,c}, v_b)$, donde w son los pesos entre neuronas y v las señales.

Una red de Hopfield [1] es un tipo de red neuronal recurrente (en inglés: *Recurrent Neural Network, RNN*), es decir, una red neuronal donde las neuronas están conectadas en ambas direcciones, basada en el modelo de Ising para materiales magnéticos [2]. La principal característica de este tipo de redes neuronales es su capacidad de memoria asociativa, donde el concepto de memoria se refiere a la tendencia de la red neuronal hacia un estado concreto.

Las neuronas de una red de Hopfield son binarias (solo pueden estar activas o inactivas), por lo que $v \in \{1, -1\}$. Es una red dinámica, así que para una red de N neuronas el estado de la red en un tiempo t viene dado por el vector $V(t) = (v_1(t), \dots, v_N(t))$.

La evolución de los estados se hará de forma asíncrona, es decir, no se activan todas las neuronas al mismo tiempo, sino que t irá incrementando en intervalos discretos en los que solo cambiará el valor de una neurona.

Para calcular el valor que tomará la neurona V_i en $t + 1$, tomamos la suma ponderada del estado de las otras neuronas y, en función de si esa suma supera un valor umbral U_i o no, asignaremos el valor 1 o -1:

$$\sum_j w_{ij} V_j(t) > U_i \Rightarrow V_i(t+1) = 1 \quad \sum_j w_{ij} V_j(t) < U_i \Rightarrow V_i(t+1) = -1 \quad (2)$$

La capacidad de memoria de la red viene dada por la selección de los pesos w_{ij} , dados por la teoría Hebbiana:

$$w_{ij} = v_i v_j, \quad \forall i \neq j \quad w_{ii} = 0, \quad \forall i \quad (3)$$

Esta regla asigna un peso mayor a las neuronas que se activan (o no) simultáneamente y menor a las que se activan de forma alterna.

En la Figura 1 podemos ver un ejemplo de una red de Hopfield de 4096 neuronas, donde cada neurona está representada por un píxel (los dos valores posibles se asignan a dos colores). En la Figura 1a tenemos la imagen que queremos memorizar, por lo que este estado de la red neuronal será nuestro objetivo y con el que calcularemos los pesos usando (3). En la Figura 1b tenemos el estado en $t = 0$ y, si lo dejamos evolucionar, convergerá al estado estable que vemos en la Figura 1c.

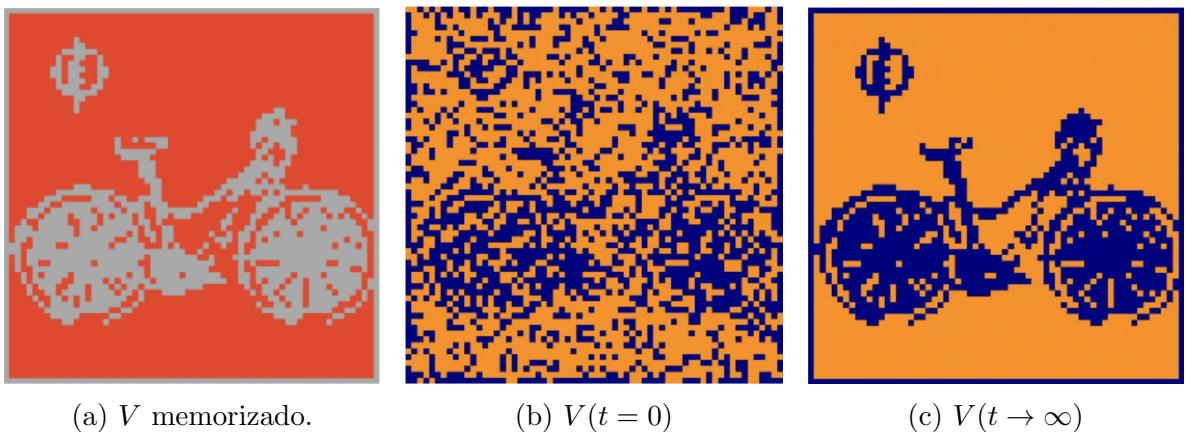


Figura 1: Ejemplo de una red de Hopfield de 4096 neuronas, visualizada como matriz de píxeles, memorizando una imagen [3].

Es decir, un modelo de red neuronal basado en un modelo físico, como es el modelo de Ising, presenta propiedades muy interesantes como la aparición de una memoria asociativa.

Las redes de Hopfield y otros tipos, como las máquinas de Boltzmann (llamadas así por su uso de la distribución de Boltzmann), siguen estudiándose hoy en día, mejorando sus capacidades y su memoria y mostrando cómo la física puede ayudar en el avance del aprendizaje automático.

2.3. Agrupamiento

El agrupamiento (en inglés, *clustering*), una técnica de aprendizaje no supervisado, estudia los diferentes métodos para clasificar los puntos de un conjunto de datos en grupos. Como el concepto de «grupo» no tiene una definición formal, existen un gran número de algoritmos posibles para usar, así que la elección del algoritmo y sus parámetros viene dada por el problema a resolver.

Aporta soluciones a problemas no supervisados en campos tan diversos como, por ejemplo, la genética, buscando similaridades en datos genéticos; la imagen médica, segmentando diferentes tejidos en imágenes tridimensionales, o el estudio de mercados, agrupando consumidores en función de sus preferencias. Además, también es de gran utilidad en problemas de aprendizaje supervisado, donde se puede usar como parte del análisis exploratorio previo para aportar información sobre la estructura de nuestro conjunto de datos (por ejemplo, podemos aplicar diferentes algoritmos a cada grupo obtenido).

A continuación, presentaremos dos populares algoritmos de agrupamiento que usaremos después para comparar con el agrupamiento cuántico probabilístico.

2.3.1. K-medias

El algoritmo K-medias [4] es una técnica sencilla para agrupar las observaciones de un conjunto de datos en K grupos distintos y disjuntos (sin elementos en común).

Dado un conjunto de observaciones $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, donde cada observación es un vector real p -dimensional, y siendo $\mathbf{C} = \{C_1, \dots, C_K\}$ los conjuntos que contienen los índices de las observaciones en cada uno de los K grupos, se debe cumplir:

1. $C_1 \cup C_2 \cup \dots \cup C_K = \{1, \dots, n\}$
2. $C_k \cap C_{k'} = \emptyset \quad \forall k \neq k'$

Es decir, si la observación i -ésima pertenece al cluster k -ésimo, entonces $i \in C_k$. El objetivo del algoritmo es que esta distribución minimice la varianza dentro de cada grupo, es decir:

$$\arg \min_{\mathbf{C}} \sum_{k=1}^K \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 \quad (4)$$

donde $\|\dots\|$ es la norma euclídea y $\boldsymbol{\mu}_k$ es el centroide (la media) del grupo C_k :

$$\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x} \quad (5)$$

siendo $|C_k|$ el tamaño de C_k . Este problema es equivalente a minimizar la distancia entre pares de puntos en un mismo grupo.

A pesar de que buscar el mínimo absoluto es computacionalmente muy costoso, ya que hay aproximadamente K^n formas de agrupar n observaciones en K grupos, el siguiente algoritmo nos permite encontrar un mínimo local:

1. Asignar un número aleatorio, desde 1 hasta K , a cada una de las observaciones. Estas serán las asignaciones a grupos iniciales.
2. Iterar *a)* y *b)* hasta que las asignaciones dejen de cambiar:
 - a)* Para cada uno de los K grupos, calcular su centroide μ .
 - b)* Asignar cada observación al grupo cuyo centroide esté más cerca (usando la distancia euclídea).

En la Figura 2 observamos un ejemplo de los pasos del algoritmo para un conjunto de datos bidimensional, con $K = 3$.



Figura 2: Pasos del algoritmo de K-medias [5].

Este algoritmo simple y eficiente nos da una solución satisfactoria siempre y cuando tengamos en cuenta que la solución es un mínimo local, por lo que es importante repetir el algoritmo tomando diferentes configuraciones iniciales aleatorias y tomando aquel con mínima varianza.

Sin embargo, la varianza que minimizamos en (4) conlleva algunos problemas, ya que asume grupos convexos e isotrópicos, por lo que no funciona bien con formas alargadas o irregulares. Además, el algoritmo de K-medias tiene un problema fundamental: necesita como parámetro el número de grupos que tiene el conjunto de datos, información de la que podríamos no disponer *a priori*.

Para observar cómo se comporta el algoritmo, vamos a crear dos conjuntos de datos bidimensionales de prueba donde tenemos puntos separados en tres grupos con formas diferentes (Figura 3).

En la Figura 4 podemos ver la asignación de grupos correspondiente al algoritmo K-medias para estos dos conjuntos de datos, tomando como parámetro $K = 3$. Observamos que, en la Figura 4a, hay un buen agrupamiento para los puntos céntricos de los grupos, pero no tiene en cuenta sus diferentes densidades; mientras que en la Figura 4b no separa

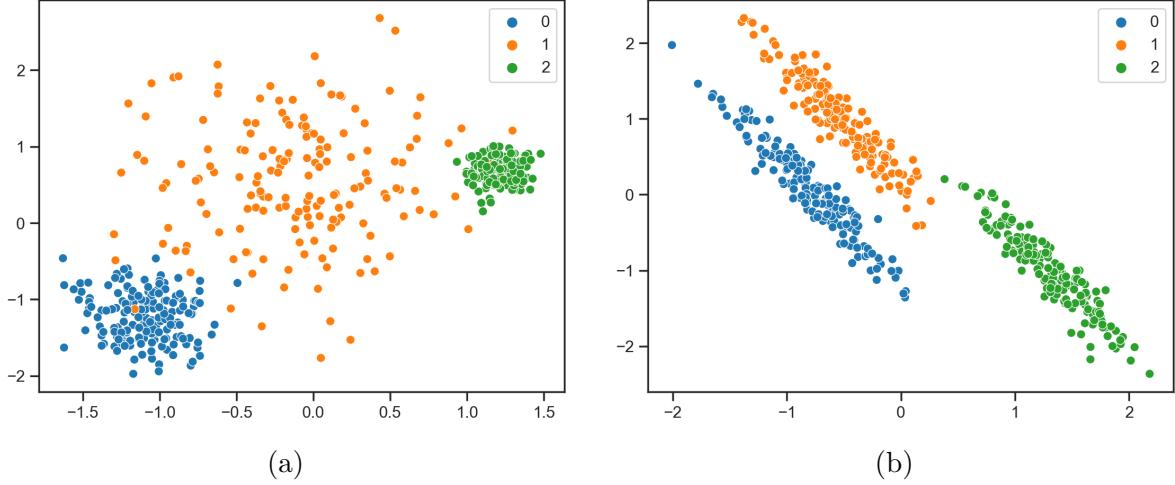


Figura 3: Representación de dos conjuntos de datos bidimensionales. Ambos tienen tres grupos, pero están distribuidos con formas diferentes.

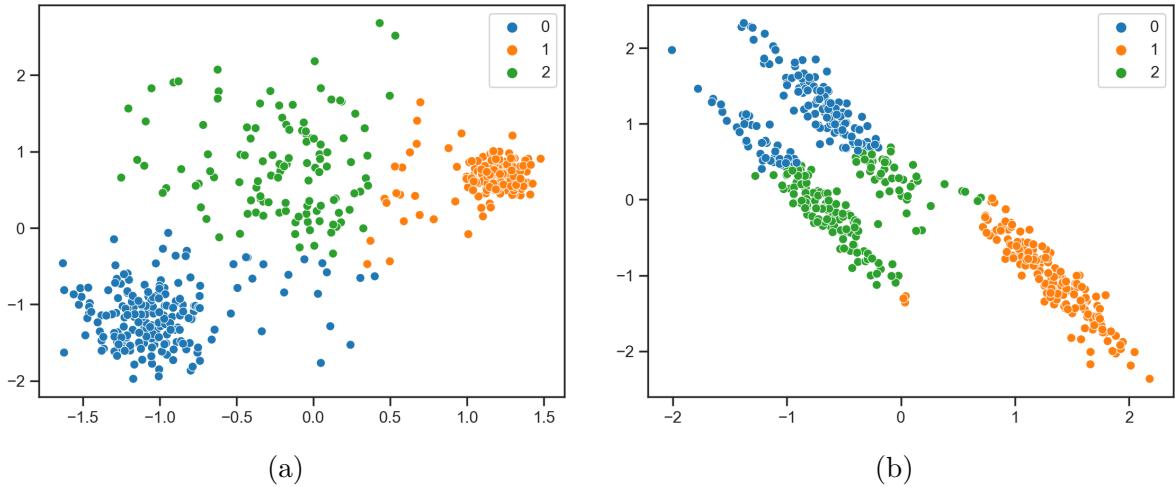


Figura 4: Ejemplos de asignaciones de grupos hechas mediante K-medias, con $K = 3$.

bien dos grupos ya que, como hemos comentado, la varianza del algoritmo no representa bien grupos con distribuciones anisotrópicas.

2.3.2. DBSCAN

El algoritmo DBSCAN (por sus siglas en inglés: *Density-based spatial clustering of applications with noise*, en español: agrupamiento espacial basado en densidad de aplicaciones con ruido) [6] trata a los grupos como zonas de alta densidad rodeadas de áreas de baja densidad, por lo que es flexible con las distribuciones de puntos con formas diversas (a diferencia de K-medias).

Dado un conjunto de puntos (observaciones), el objetivo de DBSCAN es clasificarlos en cuatro tipos: punto núcleo, directamente alcanzable, alcanzable o ruido. Para hacerlo, usa dos parámetros (ε y $minPts$) y sigue los siguientes criterios:

- Un punto p es un *punto núcleo* si, como mínimo, hay $minPts$ puntos en un radio de ε (incluyendo a p).

- Un punto q es *directamente alcanzable* desde p si el punto q está a una distancia menor de p que ε .
- Un punto q es *alcanzable* desde p si existe una secuencia p_1, \dots, p_n con $p_1 = p$ y $p_n = q$ donde cada p_{i+1} es directamente alcanzable desde p_i .
- Todo punto que no entre en las clases anteriores se considera *ruido*.

Si p es un punto núcleo, este forma un grupo con todos los puntos q_i alcanzables desde p . Es decir, todo punto núcleo es parte de un grupo, y todo punto que no es núcleo y que se encuentra a una distancia mayor que ε de cualquier punto núcleo es clasificado como ruido.

Este algoritmo tiene dos parámetros a elegir: $minPts$ controla principalmente la tolerancia del algoritmo ante el ruido y ε es el parámetro crucial para un buen funcionamiento (si ε es demasiado pequeño, la mayoría de puntos serán clasificados como ruido; si es demasiado grande, solo detectará un grupo con todos los puntos). Mientras que $minPts$ lo estimaremos a partir del problema y no hay una gran dependencia de él, necesitamos un método para elegir ε . En este trabajo, utilizaremos el método del codo.

El método del codo [7] para encontrar un valor satisfactorio de ε consiste en calcular, para cada punto, la distancia (euclídea) media a sus k vecinos más cercanos (nosotros tomaremos solo el vecino más próximo) y representarlas en orden ascendente. La idea es que los puntos pertenecientes a un grupo tendrán una distancia pequeña, mientras que el ruido tendrá valores altos (ya que están más aislados). Por tanto, al hacer la representación veremos una zona donde la pendiente cambia rápidamente: el codo. Este es un buen punto para diferenciar entre distancias «grandes» y «pequeñas» en el conjunto de datos, por lo que puede ser un ε válido.

En la Figura 5 podemos ver cómo es la asignación de grupos mediante DBSCAN a los conjuntos de datos de la Figura 3, usando el método del codo para ε y tomando $minPts = 7$, ya que es un valor que no asigna mucho ruido. En la Figura 5a detecta bien los dos grupos con mayor densidad, pero clasifica varios puntos del grupo menos denso como ruido (representado en negro y con etiqueta -1). La mayor diferencia con K-medias la observamos en la Figura 5b, ya que separa satisfactoriamente los tres grupos a pesar de sus formas alargadas.

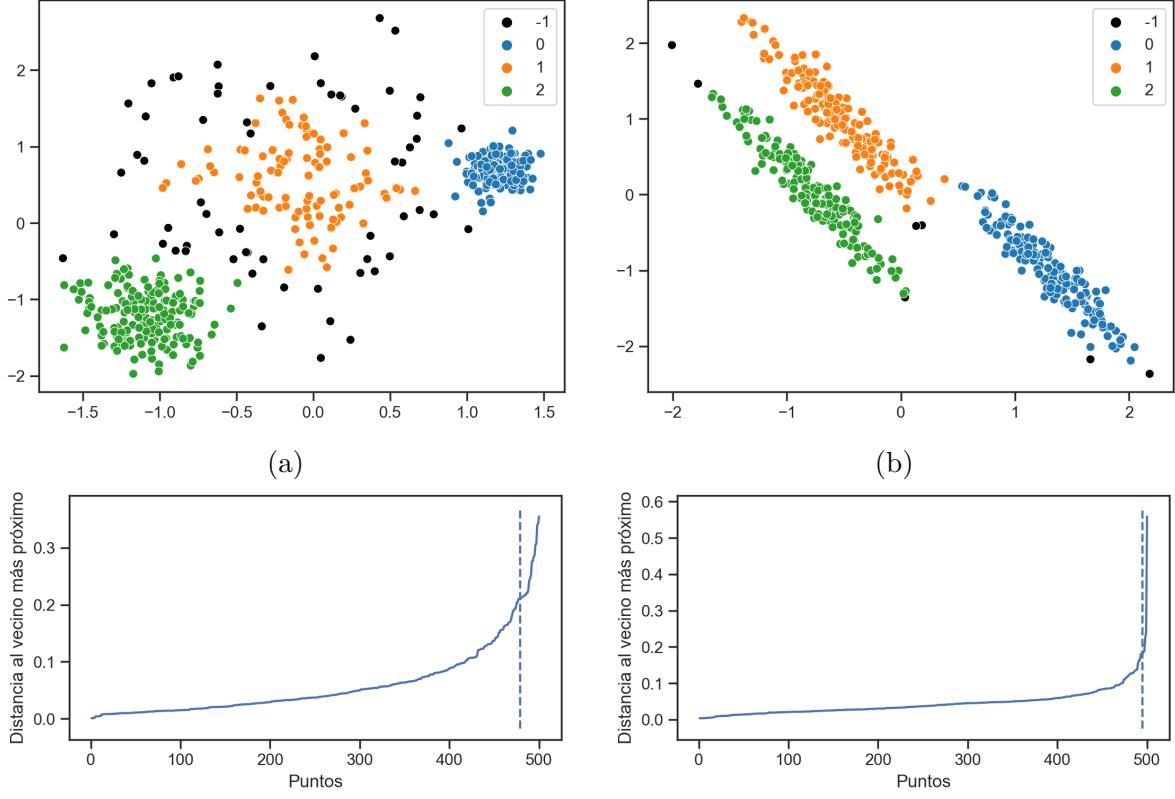
2.4. Agrupamiento cuántico probabilístico

Hemos visto cómo la inspiración en otras disciplinas (en nuestro caso, la física) puede ser útil en la formulación de nuevos algoritmos de aprendizaje automático. Es así como, tomando herramientas de la mecánica cuántica, surge el agrupamiento cuántico.

El agrupamiento cuántico [8] (en inglés: *Quantum Clustering*, QC) hace uso del potencial de la ecuación de Schrödinger para agrupar las observaciones. Sin embargo, el algoritmo original tenía márgenes de mejora en ciertos aspectos, como la elección de los parámetros necesarios o la detección de grupos con formas y densidades variadas.

En respuesta a estos problemas se propone el agrupamiento cuántico probabilístico [9] (en inglés: *Probabilistic Quantum Clustering*, PQC), que dota al algoritmo original de herramientas estadísticas, entre otras mejoras, y que da resultados más precisos en general. Por esta razón, durante el trabajo utilizaremos el PQC únicamente.

A continuación, presentaremos la teoría en la que se basa y las herramientas que usaremos.



(c) Método del codo para la Figura 5a, con $\varepsilon \approx 0,21$.

(d) Método del codo para la Figura 5b, con $\varepsilon \approx 0,18$.

Figura 5: Ejemplos de asignaciones de grupos hechas mediante DBSCAN. En las Figuras 5c y 5d la línea punteada corta el punto con mayor curvatura: el codo. El valor de ε corresponde a la coordenada y de dicho punto.

2.4.1. Ecuación de Schrödinger

Comenzamos tomando la ecuación de Schrödinger independiente del tiempo para una partícula simple no relativista:

$$H\Psi = \left[\frac{-\sigma^2}{2} \nabla^2 + V(\mathbf{x}) \right] \Psi(\mathbf{x}) = E\Psi(\mathbf{x}) \quad (6)$$

donde H es el Hamiltoniano, E es la energía del estado $\Psi(\mathbf{x})$ y σ designa el parámetro de escala de longitud (en nuestro caso será la desviación estándar de las gaussianas, como veremos).

El objetivo es construir una función de onda $\Psi(\mathbf{x})$ total asignando a cada observación una función de onda $\psi_i(\mathbf{x})$, calcular el potencial a partir de (6), aplicar a dicho potencial un descenso del gradiente para encontrar los mínimos locales y asignar los grupos basándonos en probabilidad bayesiana.

Podemos despejar el potencial de (6) para obtener:

$$V(\mathbf{x}) = E + \frac{\sigma^2}{2} \frac{\nabla^2 \Psi(\mathbf{x})}{\Psi(\mathbf{x})} \quad (7)$$

Hay diversas distribuciones de probabilidad que pueden tener las funciones de onda que asignemos a cada observación. En este algoritmo se usan funciones gaussianas normalizadas debido a su suavidad (si queremos calcular el gradiente de (7), necesitamos

que sean diferenciables hasta tercer orden). Además, para una mejor detección de grupos con distribuciones variadas, estas gaussianas tienen matrices de covarianza no esféricas (multivariantes).

Para estimar la matriz de covarianza correspondiente a cada observación, calculamos su matriz de covarianza local Σ_i (8) a partir de la distribución relativa de los k -vecinos más cercanos (en inglés: *k-nearest neighbors, knn*) [10]. La elección de la k (es decir, la cantidad de vecinos próximos que debemos tomar) la explicaremos más adelante.

$$\Sigma_i = \frac{1}{N_k - 1} \sum_{j \in knn}^{N_k} (\mathbf{x}_j - \mathbf{x}_i)^T (\mathbf{x}_j - \mathbf{x}_i) \quad (8)$$

Para calcular la función de onda total, sumamos todas estas gaussianas multivariantes:

$$\Psi(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \psi_i(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{|2\pi\Sigma_i|}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)^T \Sigma_i^{-1} (\mathbf{x}-\mathbf{x}_i)} \quad (9)$$

En el algoritmo original, se tomaban gaussianas de igual desviación estándar para todas las observaciones, obteniendo una expresión más sencilla: $\Psi(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \psi_i(\mathbf{x}) = \sum_{i=1}^n e^{-\frac{-(\mathbf{x}-\mathbf{x}_i)^2}{2\sigma^2}}$. Sin embargo, esta función de onda no representaba con precisión distribuciones de puntos con formas variadas, como espirales o formas alargadas. A pesar de que esto se soluciona con las matrices de covarianza, esta expresión para la función de onda (9) también conlleva algunos problemas:

1. La complejidad de la expresión del potencial aumenta considerablemente.
2. Las matrices de covarianza no invertibles (con elementos en la diagonal cercanos a 0) pueden causar errores.
3. Si las covarianzas son demasiado anisotrópicas, no se producirá suficiente superposición en $\Psi(\mathbf{x})$, generándose así una función de onda menos suave y un número excesivo de mínimos locales.

Para mitigar estos problemas, se representan las matrices de covarianza en su base de autovectores, siendo así diagonales, y se selecciona un valor mínimo para que los elementos de la diagonal no sean cercanos a 0. Además, este valor mínimo también mejora la superposición de las diferentes funciones de onda ψ_i . El valor mínimo, para cada observación i , es el siguiente:

$$\sigma_{th_i}^2 = \frac{\sigma_{k'nni}^2}{p} \quad (10)$$

donde p es la dimensión de las observaciones (el número de características de los vectores \mathbf{x}_i) y $\sigma_{k'nni}$ es la distancia media de los k' vecinos más cercanos a la observación i . Los resultados [9] muestran que esta k' debe coincidir con la k que usamos para calcular las matrices de covarianza (8).

Este parámetro k , dado en porcentaje respecto al total de observaciones (%KNN), es el parámetro que tenemos que elegir para hacer el agrupamiento. Es decir, si tomamos un valor de 20 % y tenemos 10 observaciones, estaríamos tomando para cada punto i los dos vecinos más próximos tanto en (8) como en (10). Al igual que en el resto de algoritmos de agrupamiento, el resultado depende de elegir bien este parámetro; sin embargo, veremos cómo el PQC tiene herramientas para guiarnos en esta elección.

Para calcular el potencial, podemos adaptar el parámetro de escala de longitud de la ecuación de Schrödinger (6) σ_i a las matrices de covarianza, haciendo el cambio $\sigma_i^2 \rightarrow \text{Tr}(\Sigma_i)$. La derivación de $\nabla^2\psi_i$ (7) está basada en las ecuaciones propuestas en [11]:

$$\frac{\partial\psi_i}{\partial\mathbf{x}} = -\psi_i\Sigma_i^{-1}(\mathbf{x} - \mathbf{x}_i) \quad (11)$$

$$\frac{\partial^2\psi_i}{\partial\mathbf{x}\partial\mathbf{x}^T} = -\psi_i(\Sigma_i^{-1}(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T\Sigma_i^{-1} - \Sigma_i^{-1}) \quad (12)$$

$$\nabla^2\psi_i = \text{Tr}\left(\frac{\partial^2\psi_i}{\partial\mathbf{x}\partial\mathbf{x}^T}\right) \quad (13)$$

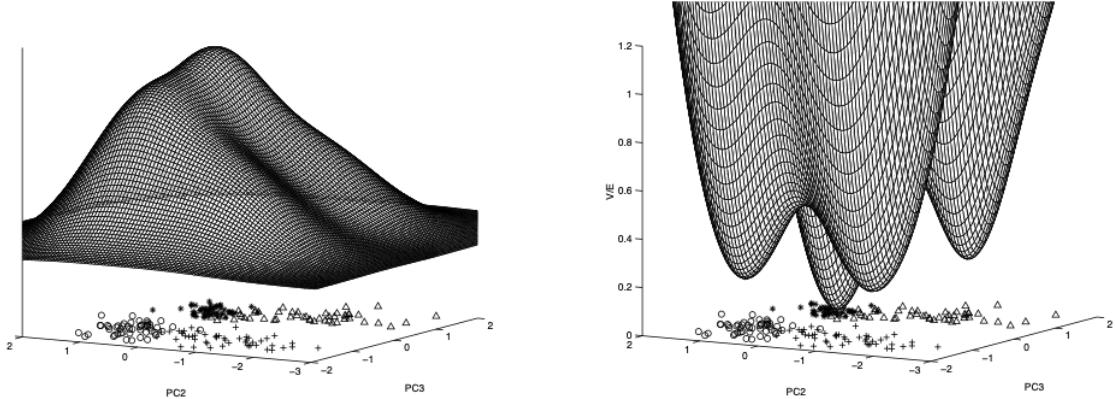
Podemos obtener la expresión del potencial a partir de estas ecuaciones y expresarlo en función de los valores esperados, $\langle F \rangle_\Psi = \frac{\sum_i F_i \psi_i}{\sum_i \psi_i}$:

$$V(x) = E + \frac{\sum_i \frac{\text{Tr}(\Sigma_i)}{2} \psi_i \text{Tr}(\Sigma_i^{-1}(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T\Sigma_i^{-1} - \Sigma_i^{-1})}{\sum_i \psi_i} = \quad (14)$$

$$= E + \left\langle \frac{\text{Tr}(\Sigma_i)}{2} \text{Tr}(\Sigma_i^{-1}(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T\Sigma_i^{-1}) \right\rangle_\Psi - \left\langle \frac{1}{2} \text{Tr}(\Sigma_i) \text{Tr}(\Sigma_i^{-1}) \right\rangle_\Psi$$

donde E es una constante que no afecta al descenso del gradiente que vamos a aplicar y que podemos tomar para que $V \geq 0$.

Para ilustrar el proceso que acabamos de ver, vamos a ver un ejemplo visual. En la Figura 6a vemos la función de onda total $\Psi(\mathbf{x})$, generada por los puntos que se encuentran en el plano inferior (se usan diferentes símbolos para representar los cuatro grupos de observaciones que tiene este conjunto de datos). A partir de esta distribución, se calcula el potencial de la ecuación de Schrödinger, como vemos en la Figura 6b.



(a) Representación de la función de onda total generada. En el plano inferior están representados los puntos \mathbf{x}_i . (b) Representación del potencial de Schrödinger obtenido. Se observan cuatro mínimos, correspondientes a los cuatro grupos.

Figura 6: Ejemplo del uso de la ecuación de Schrödinger para el cálculo del potencial [8].

2.4.2. Asignación de grupos

Una vez generado el potencial y obtenidos los mínimos locales (que serán los centros de los grupos), debemos asignar cada punto a un grupo. Para hacerlo, seguiremos dos pasos: un descenso del gradiente en el potencial (método del QC) y, a partir de esta asignación, haremos la asignación probabilística (característica del PQC).

Para el primer paso, haremos un descenso del gradiente para cada punto. Tomando $\mathbf{y}_i(0) = \mathbf{x}_i$:

$$\mathbf{y}_i(t + \Delta t) = \mathbf{y}_i(t) - \eta(t) \nabla V(\mathbf{y}_i(t)) \quad (15)$$

donde $\eta(t)$ es la tasa de aprendizaje.

En concreto, se aplica el algoritmo Adam [12], una variante del descenso de gradiente estocástico con un término de momento adaptativo que lo hace útil para conjuntos de datos con escasos puntos o valores atípicos. Además, para asegurar la convergencia del algoritmo se imponen dos criterios:

$$\max(|\Delta \mathbf{y}_i|) \leq \epsilon_y \quad \max(\Delta V(\mathbf{y}_i)) \leq \epsilon_V \quad (16)$$

donde ϵ son los valores mínimos (empíricamente, un buen valor para ambos es $\epsilon \approx 0,001$ para datos con una longitud media de aproximadamente 1). En la Figura 7 podemos ver un ejemplo del descenso del gradiente y la asignación a los grupos resultante para un conjunto de datos de prueba bidimensional.

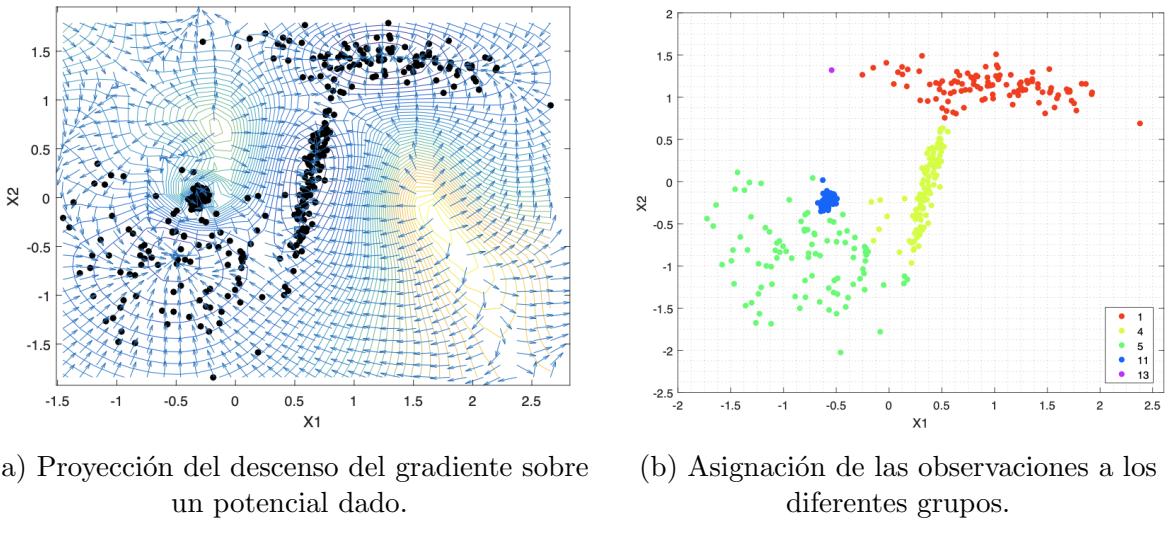


Figura 7: Ejemplo de un descenso del gradiente aplicado sobre los puntos y la consecuente asignación de grupos [9].

Sin embargo, como hemos comentado, en el PQC esta asignación a los grupos no es la definitiva, sino que estos grupos son usados para separar la función de onda en varias subfunciones. Esta asignación de los grupos basada en probabilidad conlleva mejores resultados y, además, dota al algoritmo de varias herramientas útiles.

En el primer paso, a partir del descenso del gradiente, hemos obtenido K grupos. Vamos a usar estos grupos para dividir la función de onda $\Psi(\mathbf{x})$ en K subfunciones, obteniendo así la probabilidad de observar el grupo k en la posición \mathbf{x} , $P(k, \mathbf{x})$:

$$\Psi(\mathbf{x}) = \sum_{k=1}^K \frac{\sum_{i \in k} \psi_i(\mathbf{x})}{n} = \sum_{k=1}^K P(k, \mathbf{x}) = P(\mathbf{x}) \quad (17)$$

donde n es el número de observaciones total y $\#k$ el número de observaciones en el grupo k .

A partir de $P(k, \mathbf{x})$, podemos calcular la probabilidad de k integrando en \mathbb{R} :

$$P(k) = \int_{\mathbb{R}} P(k, \mathbf{x}) d\mathbf{x} = \int_{\mathbb{R}} \frac{\sum_{i \in k}^{\#k} \psi_i(\mathbf{x})}{n} d\mathbf{x} = \sum_{i \in k}^{\#k} \frac{\int_{\mathbb{R}} \psi_i(\mathbf{x})}{n} d\mathbf{x} = \sum_{i \in k}^{\#k} \frac{1}{n} = \frac{\#k}{n} \quad (18)$$

Con estas probabilidades, podemos calcular las probabilidades condicionadas:

$$P(k | \mathbf{x}) = \frac{P(k, \mathbf{x})}{P(\mathbf{x})} = \frac{\sum_{i \in k}^{\#k} \psi_i(\mathbf{x})}{\sum_{k=1}^K \sum_{i \in k}^{\#k} \psi_i(\mathbf{x})} \quad (19)$$

$$P(\mathbf{x} | k) = \frac{P(k, \mathbf{x})}{P(k)} = \frac{\sum_{i \in k}^{\#k} \psi_i(\mathbf{x})}{\#k} \quad (20)$$

Y, con estas probabilidades, podemos definir la segunda asignación de grupos:

$$\text{grupo}(\mathbf{x}) = \arg \max_k P(k | \mathbf{x}) \quad (21)$$

Es decir, dado un punto \mathbf{x} , se asignará el grupo k que maximice $P(k | \mathbf{x})$ o, equivalentemente, $P(k, \mathbf{x})$.

Esta asignación de grupos probabilística solo usa el descenso del gradiente como herramienta al inicio para dividir en subfunciones la función de onda, pero hace la asignación basándose en maximizar la probabilidad $P(k | \mathbf{x})$. Esto hace que pueda haber k' grupos vacíos si $P(k' | \mathbf{x})$ nunca es mayor que el resto de probabilidades en el espacio de las observaciones.

No solo esta asignación probabilística trae consigo una mejora de los resultados (como se comprueba en [9]), sino que nos permite asignar grupos a todos los puntos del espacio, así que podemos clasificar nuevas observaciones en los diferentes grupos sin necesidad de volver a hacer todo el proceso (solo es necesario hacer un descenso del gradiente inicial). Además, aporta otra herramienta: la capacidad de detectar de valores atípicos si la probabilidad de un punto de pertenecer a cualquier grupo es menor que un valor mínimo ϵ dado (es decir, $P(\mathbf{x} | k) < \epsilon \Rightarrow$ valor atípico).

Vamos a ver un ejemplo partiendo de la asignación de grupos inicial de la Figura 7. En las Figuras 8a y 8b vemos la representación de $P(k | \mathbf{x})$ para los cinco grupos iniciales. Podemos observar que el grupo 5 no llega a tener una probabilidad mayor que la de los otros grupos en ningún punto del espacio, por lo que ninguna observación se asignará a este grupo, como vemos en la Figura 8c. Además, podemos calcular la probabilidad $P(\mathbf{x} | k)$ máxima de cada punto del espacio, como vemos en la Figura 8d, y seleccionar un valor mínimo para clasificar valores atípicos.

2.4.3. Elección de parámetros: ANLL

Hemos visto que, en el aprendizaje no supervisado, los resultados de un algoritmo dependen en gran medida de la elección de los parámetros necesarios. Como los conjuntos de datos que se usan no tienen respuestas asignadas, tenemos que buscar un método fiable para la elección. En el PQC, las herramientas probabilísticas nos pueden guiar en la elección del parámetro necesario: $\%KNN$ (el número k de vecinos más próximos, expresado como porcentaje respecto al total de observaciones).

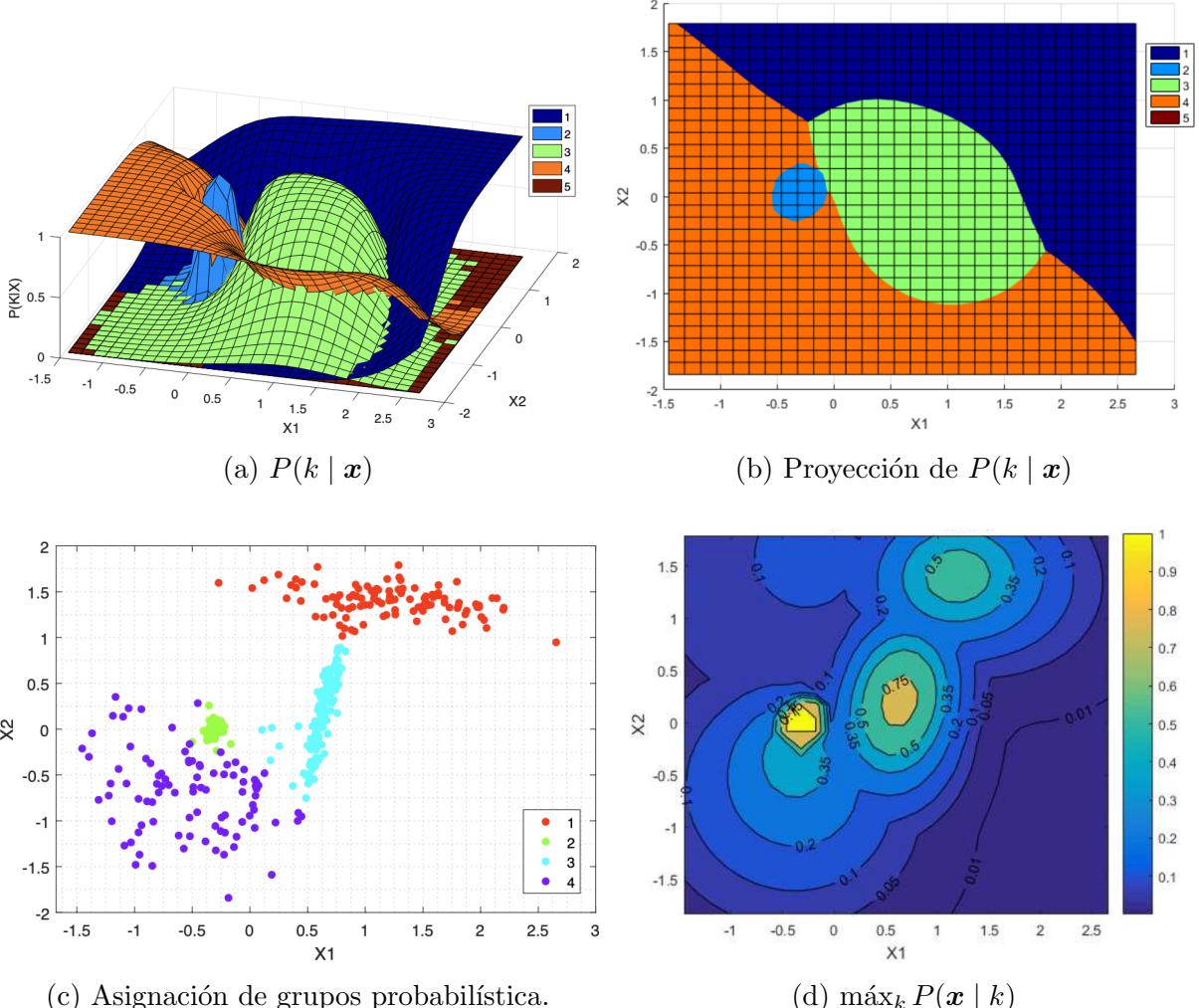


Figura 8: Ejemplo de las probabilidades en el espacio de observaciones y la consecuente asignación de grupos probabilística [9].

Sea k_w el grupo asignado a la observación \mathbf{x}_i (porque la probabilidad $P(k_w | \mathbf{x}_i)$ ha sido superior a las probabilidades para el resto de grupos). Cuanto mejor sea el agrupamiento realizado sobre el conjunto de datos, mayor será, en general, $P(k_w | \mathbf{x}_i)$. Podemos expresarlo en términos de la verosimilitud logarítmica (en inglés: *Log Likelihood*, *LL*):

$$LL(k | \mathbf{x}) = \log \left(\prod_i^n P(k_w | \mathbf{x}_i) \right) = \sum_i^n \log (P(k_w | \mathbf{x}_i)) \quad (22)$$

Normalizando (22) en el rango $[0, 1]$, definimos el *ANLL* (del inglés, *Average Negative Log-Likelihood*):

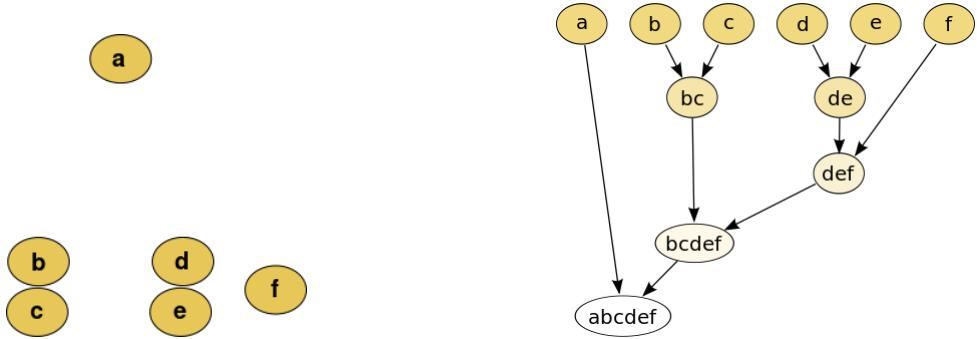
$$ANLL(k | \mathbf{x}) = -\frac{\sum_i^n \log (P(k_w | \mathbf{x}_i))}{N} \quad (23)$$

Un menor *ANLL* se corresponde, en general, con un mejor resultado, por lo que calculando el *ANLL* para diversos $\%KNN$ podemos buscar mínimos locales que nos den agrupamientos satisfactorios. Además, cambios abruptos en el *ANLL* se corresponden con cambios en la estructura de los grupos obtenidos, por lo que podemos obtener información útil sobre nuestros datos.

Hay que tener en cuenta que la solución trivial, es decir, asignar todos los puntos a un solo grupo, tiene $ANLL$ nulo, por lo que debemos buscar estos mínimos locales en un rango de $\%KNN$ adecuado.

2.4.4. Agrupamiento jerárquico

Otra herramienta muy útil que presenta el PQC es la posibilidad de observar la estructura jerárquica de nuestros datos. Un algoritmo jerárquico puede ser aglomerativo (empieza con un cierto número de grupos y va uniendo aquellos que sean similares) o divisorio (empieza con un solo grupo y va separándolos). En la Figura 9 observamos el diagrama de árbol del proceso de un algoritmo aglomerativo, también conocido como dendrograma.



(a) Representación de seis grupos, donde la distancia entre ellos depende de su similitud (por ejemplo, b es más similar a c que a a).

(b) Agrupamiento jerárquico, donde observamos que en cada fase se unen aquellos grupos más similares entre ellos.

Figura 9: Esquema del proceso de un algoritmo de agrupamiento jerárquico aglomerativo en cinco fases.

Para introducir esta herramienta jerárquica, usamos la primera asignación de grupos que hemos hecho con el descenso del gradiente sobre el potencial de Schrödinger. Podemos calcular las posiciones de los centroides de cada grupo tomando la media de las posiciones de los puntos de un grupo y, como conocemos el potencial en estas posiciones, podemos calcular la diferencia de potencial entre el centroide i y el centroide j :

$$\Delta V(i, j) = V_j - V_i \quad (24)$$

Estableciendo una energía mínima E , podemos unir grupos basándonos en si esta energía supera la diferencia de potencial entre centroides de grupos:

$$\Delta V(i, j) \leq E \Rightarrow \text{unir}(\text{grupo}_i, \text{grupo}_j) \quad (25)$$

Si vamos incrementando progresivamente E partiendo de cero, los grupos se irán uniendo hasta que solo haya uno y podremos obtener la jerarquía de grupos de nuestro conjunto de datos. Podemos verlo de forma intuitiva como tomar el potencial de Schrödinger (Figura 6b), ir vertiendo agua sobre cada mínimo y, cuando el agua de un pozo rebose y caiga en otro pozo, estos se unan.

Esta herramienta no solo es útil para aportarnos información sobre nuestro problema, sino que además puede darnos soluciones que no obtendríamos con el agrupamiento que hemos visto antes, aportando una mayor flexibilidad al algoritmo.

2.5. Objetivos

Este trabajo tiene dos objetivos generales: estudiar cómo la física puede ser inspiración para el aprendizaje automático y analizar los algoritmos de agrupamiento en problemas físicos (especialmente, el PQC).

Ya hemos visto cómo el aprendizaje automático se puede beneficiar de conocimientos sobre física y, en concreto, cómo la ecuación de Schrödinger es la base de un algoritmo de agrupamiento.

Ahora, queremos estudiar el agrupamiento cuántico probabilístico en física, planteándonos los siguientes objetivos:

- Analizar qué problemas físicos se pueden beneficiar de algoritmos de agrupamiento.
- Ilustrar las características y funcionalidades del PQC con un problema físico.
- Comprobar el rendimiento del PQC en problemas físicos, comparándolo con otros algoritmos de agrupamiento (K-medias y DBSCAN).
- Estudiar qué características de un problema afectan al rendimiento de los algoritmos de agrupamiento y cómo podemos mejorarlo.

3. Metodología

La implementación de los algoritmos en este trabajo ha sido llevada a cabo en Python, usando la librería Scikit-Learn [13] para los algoritmos de K-medias y DBSCAN, y la librería APQC para el PQC [9].

3.1. Problemas y conjuntos de datos

Hemos elegido dos problemas físicos diferentes que nos ayudarán a cumplir con nuestros objetivos. El primero es un problema sencillo que servirá para ilustrar de forma clara las posibilidades que nos dan los algoritmos de agrupamiento (especialmente, el PQC): agrupar las estrellas de un diagrama de Hertzsprung-Russell. El segundo es un problema de mayor complejidad e interés físico y que servirá para evaluar los rendimientos de los algoritmos de agrupamiento: reconstruir las trayectorias de partículas en un detector. A continuación, presentaremos los problemas y sus correspondientes conjuntos de datos.

3.1.1. Diagrama HR

En este primer problema, el objetivo será aplicar los algoritmos de agrupamiento a un gráfico muy conocido en física: el diagrama de Hertzsprung-Russell. Este diagrama (Figura 10) representa la relación entre las luminosidades de las estrellas y su temperatura, y ha sido de gran utilidad para comprender la evolución estelar.

La mayoría de estrellas del universo ocuparían la región diagonal del gráfico, conocida como secuencia principal, mientras que la otra región densa del diagrama, situada encima de la secuencia principal, corresponde a las estrellas gigantes. Además, tenemos zonas de menor densidad como las enanas blancas, en la parte inferior, o las estrellas supergigantes, en la parte superior. La simplicidad del diagrama, que solo consta de dos dimensiones, nos ayudará a visualizar la acción de los algoritmos.

El conjunto de datos que usaremos se compone de observaciones para 1613 estrellas de secuencia principal y 1563 estrellas gigantes, extraídas aleatoriamente de [15] (provienen

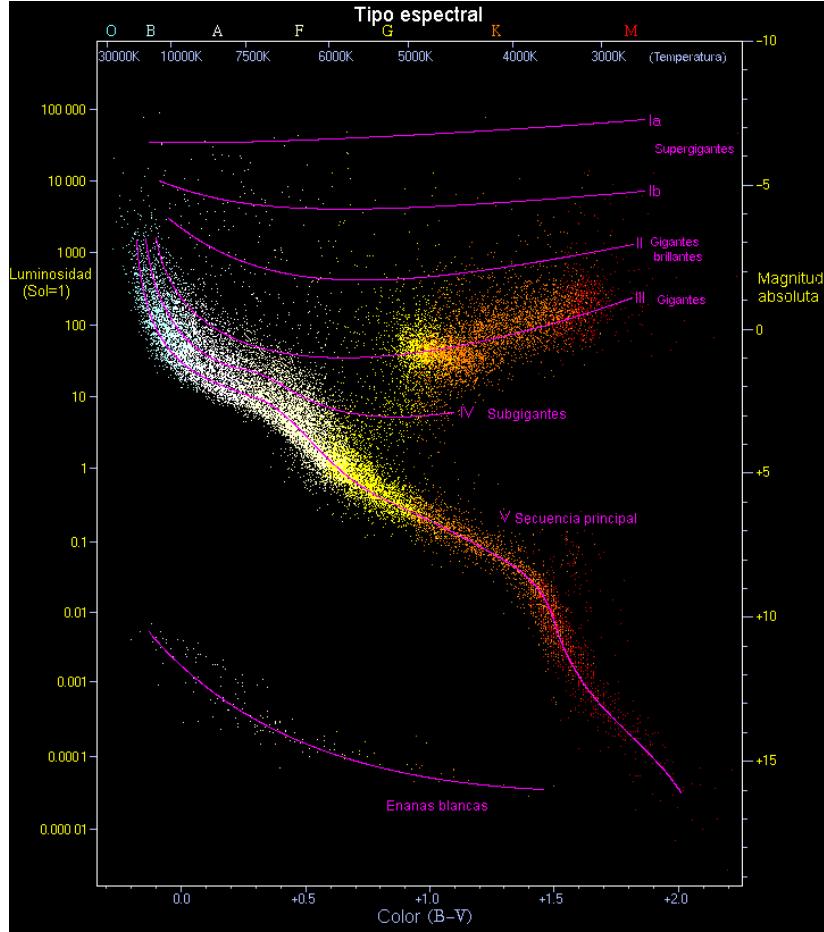


Figura 10: Ejemplo de un diagrama HR [14].

de los catálogos de Hipparcos y Tycho), y 753 enanas blancas, que hemos obtenido a partir de la herramienta en línea SIMBAD [16] para tener otra zona de mayor densidad en el diagrama.

Cada observación consta de las magnitudes aparentes m_b y m_v , la distancia expresada en términos de su paralaje p (en arcosegundos, as) y su tipo (secuencia principal, gigante o enana blanca). A partir de estos datos, podemos calcular el índice de color $B-V$ y la magnitud visual absoluta M_v , que usaremos para el diagrama, a partir de:

$$B-V = m_b - m_v \quad M_v = m_v + 5 (\log_{10} p + 1) \quad (26)$$

Podemos representar $B-V$ frente a M_v (con el eje invertido, por convención) y obtener el diagrama de Hertzsprung-Russell para nuestros datos (Figura 11), sobre el que haremos el agrupamiento.

Para hacer el agrupamiento, antes estandarizaremos las variables $B-V$ y M_v , expresándolas en unidades tipificadas. Para un valor x_i , la unidad tipificada se define como:

$$z_i = \frac{x_i - \mu_x}{\sigma_x} \quad (27)$$

donde μ_x y σ_x son la media y la desviación estándar de x , respectivamente.

Hacemos esto ya que las variables son heterogéneas: $B-V$ está en el rango $[-0.5, 2.0]$ y M_v tiene mayor dispersión, estando en el rango $[-10, 15]$, por lo que contribuirían de forma diferente a la distancia entre puntos.

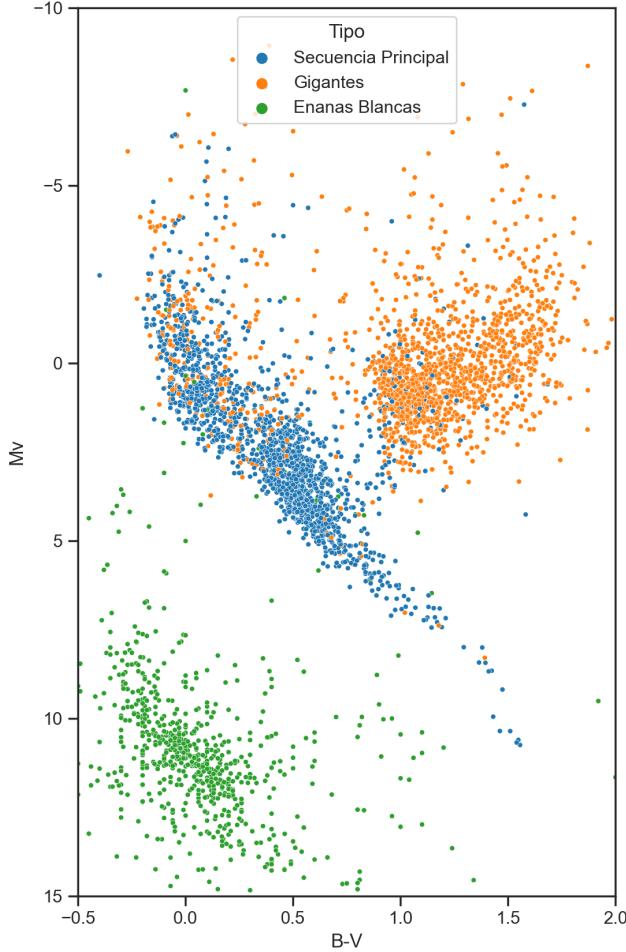


Figura 11: Diagrama HR para nuestro conjunto de datos.

Una vez estandarizadas, aplicaremos los tres algoritmos de agrupamiento que hemos visto. Respecto a la elección de los parámetros, para K-medias tomaremos la K verdadera ($K = 3$), para DBSCAN tomaremos ε siguiendo el método del codo y $minPts = 100$ y para PQC tomaremos el $\%KNN$ que minimice el $ANLL$ (eliendo un mínimo local que dé un resultado lógico). Hay que tener en cuenta que el algoritmo de K-medias parte con ventaja, ya que le estamos dando información sobre el problema (el número real de grupos), mientras que a DBSCAN y PQC no. Puede haber problemas en los que debamos aproximar K y los resultados serán peores, por lo que K-medias no es tan flexible como los otros algoritmos.

3.1.2. Seguimiento de partículas

El siguiente problema consiste en un seguimiento de partículas: tenemos varios puntos en el espacio que corresponden a instantes de las trayectorias de varias partículas y debemos agrupar los diferentes puntos para reconstruir la trayectoria de cada una. Esto emula el problema que tendríamos en un detector de partículas, donde debemos reconstruir la trayectoria completa de una partícula conociendo solo su posición en ciertos momentos.

El problema está inspirado en un desafío propuesto en la comunidad en línea Kaggle por el CERN [17], mientras que el conjunto de datos que usaremos es simulado y está creado en la Radboud University de Nijmegen dentro del marco de una colaboración internacio-

nal para profundizar en las soluciones de este problema [18]. Aunque sea originalmente un problema de aprendizaje supervisado, vamos a afrontarlo desde una perspectiva no supervisada para evaluar los algoritmos de agrupamiento.

El conjunto de datos consta de 100.000 eventos (simulaciones independientes), dentro de cada evento tenemos 20 trayectorias y para cada trayectoria disponemos de 9 puntos. Como los puntos son simulados y solo nos interesa su distribución en el espacio, las unidades de distancia (u.d.) usadas son arbitrarias. En la Figura 12 vemos una representación de cuatro trayectorias para un evento dado, donde cada color corresponde a una partícula. En la Figura 12a vemos la representación en tres dimensiones, mientras que en las Figuras 12b y 12c vemos las proyecciones en los planos XY y XZ , respectivamente.

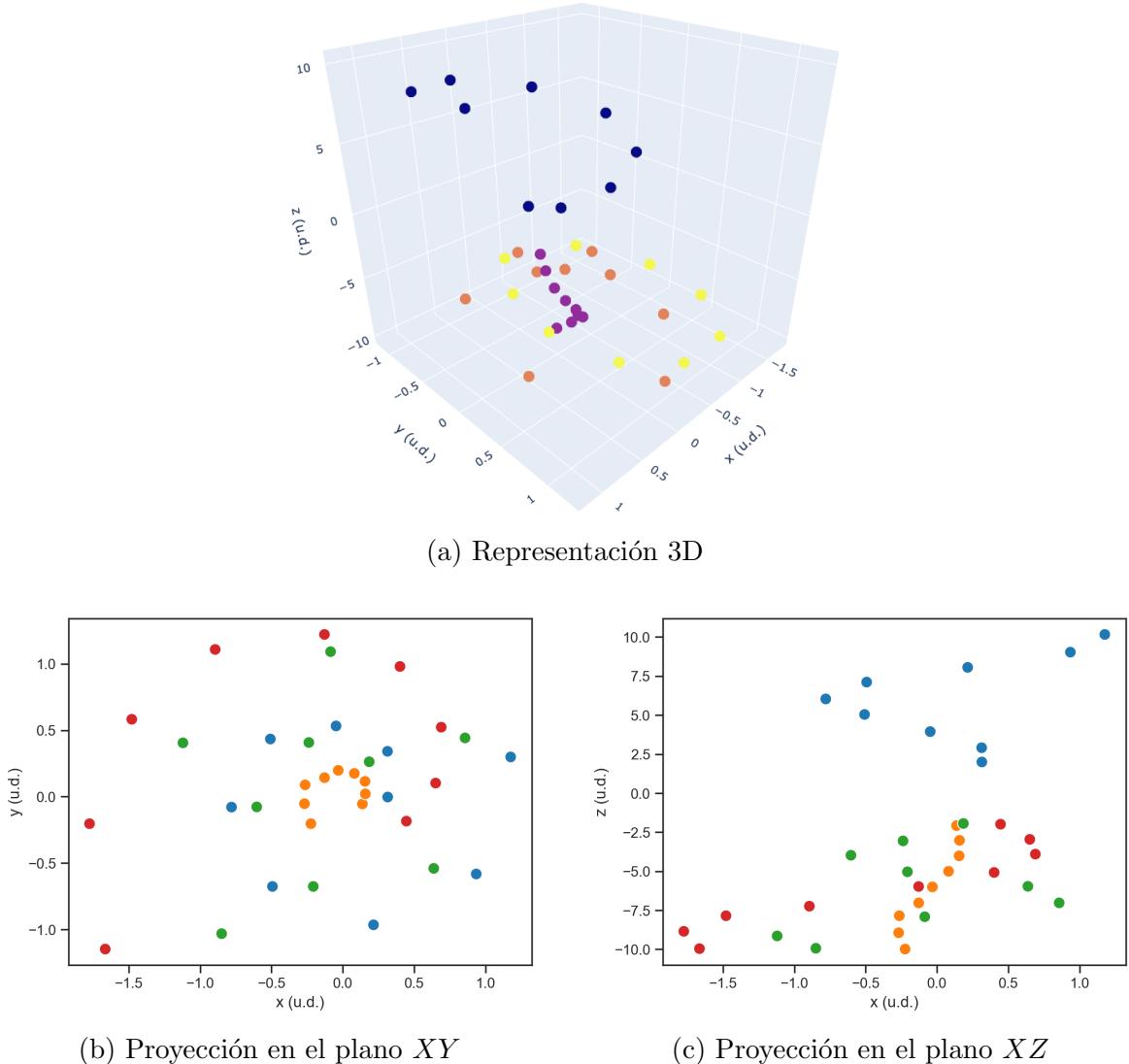


Figura 12: Representaciones de los puntos de cuatro trayectorias en coordenadas cartesianas.

Cada punto del conjunto de datos tiene las siguientes características:

- $event_id \in [0, 99.999]$: identificador numérico del evento.
- (ρ, Φ, z) : coordenadas cilíndricas, donde ρ y z vienen dados en unidades de distancia arbitrarias (u.d.) y Φ en radianes.

- $track_id \in [0, 19]$: identificador de la trayectoria a la que corresponde.

Para un $event_id$ dado, usaremos las coordenadas espaciales para hacer el agrupamiento y comprobaremos nuestros resultados usando $track_id$ (en la sección siguiente, presentaremos cómo evaluaremos los resultados).

Usar coordenadas cartesianas para hacer el agrupamiento no dará un resultado satisfactorio: las diferentes trayectorias no están separadas (como se puede observar en la Figura 12) y, por tanto, los algoritmos no las diferenciarán bien. Sin embargo, representando en coordenadas cilíndricas y esféricas (Figura 13) vemos que las trayectorias pasan a ser líneas rectas.

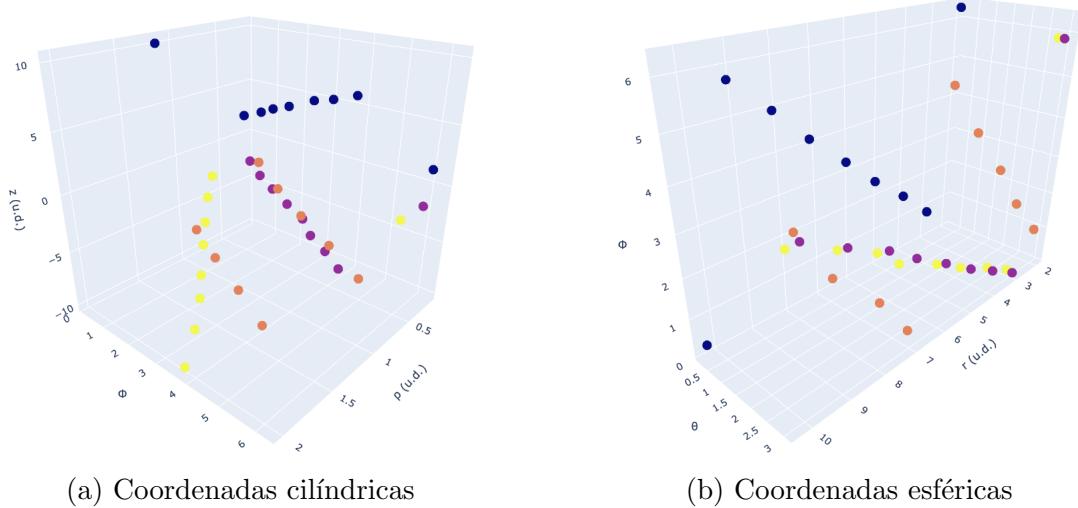


Figura 13: Representaciones de los puntos de cuatro trayectorias en coordenadas cilíndricas y esféricas.

Esto se debe a la forma que tienen las trayectorias de las partículas, ya que podemos observar que siguen, aproximadamente, trayectorias helicoidales de la siguiente forma:

$$\begin{aligned} x &= at \cos(bt) \\ y &= at \sin(bt) \\ z &= ct \end{aligned} \tag{28}$$

donde a , b y c son los parámetros para una trayectoria y $t \in \mathbb{R}$. Esto significa que los puntos que pertenezcan a una trayectoria dada tendrán, aproximadamente, mismos a , b y c , por lo que podemos despejar de las ecuaciones (28) c/a y b/a ,

$$\begin{aligned} \frac{c}{a} &= \frac{z}{\rho} \equiv u \\ \frac{b}{a} &= \frac{1}{\rho} \arccos\left(\frac{x}{\rho}\right) \equiv v \end{aligned} \tag{29}$$

y calcular, para cada punto, estas nuevas coordenadas (u, v) . Si representamos los puntos correspondientes a las Figuras 12 y 13 en este nuevo sistema (Figura 14), observamos cómo los puntos correspondientes a cada partícula forman grupos, en general, poco dispersos, por lo que este nuevo sistema de coordenadas bidimensional puede ser útil.

En este problema usaremos los dos conjuntos de variables que mayor separación entre grupos conllevan: las coordenadas esféricas (r, θ, Φ) y las coordenadas (u, v) . Además,

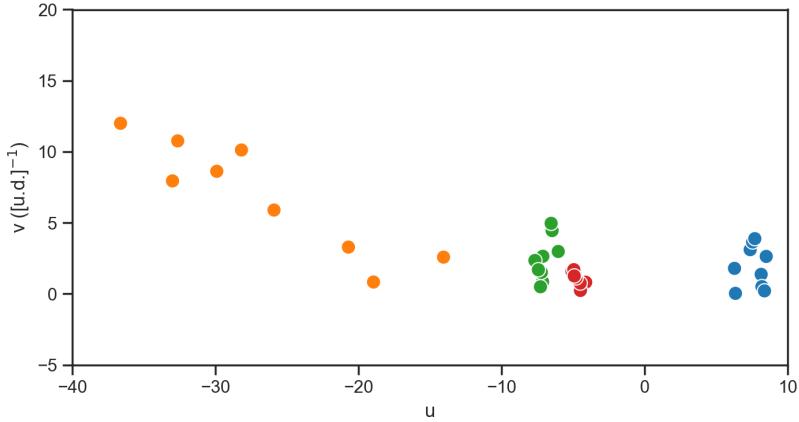


Figura 14: Representaciones de los puntos de cuatro trayectorias en las coordenadas (u, v) .

aunque en el problema anterior las variables usadas para el agrupamiento eran heterogéneas ($B-V$ y M_v) y nos interesaba estandarizarlas usando unidades tipificadas, en este problema no las estandarizaremos, ya que la dispersión que presentan sus variables es adecuada para el agrupamiento.

Para variar la complejidad de este problema y ver cómo cambia el rendimiento de los algoritmos con el número de trayectorias l , actuaremos de la siguiente forma. De los 100.000 eventos, tomaremos una muestra suficientemente grande para obtener resultados fiables (200 eventos diferentes elegidos de forma aleatoria). Dentro de cada evento, tomaremos 10 muestreos aleatorios de trayectorias para cada l y aplicaremos los tres algoritmos de agrupamiento a cada uno, tanto en coordenadas esféricas como en las coordenadas (u, v) . De esta forma, podremos evaluar cómo depende el resultado tanto del algoritmo usado como de l .

Respecto a la elección de parámetros, procederemos de forma similar al anterior problema. Para K-medias, daremos el número de trayectorias ($K = l$); para DBSCAN, usaremos el método del codo para ε y tomaremos $minPts = 5$ y, para PQC, tomaremos aquel $\%KNN$ que minimice el $ANLL$ dentro de un rango adecuado. Sin embargo, como este proceso está automatizado (vamos a hacer un gran número de agrupamientos), no podemos elegir el $\%KNN$ manualmente, por lo que vamos a elegir un rango de $\%KNN$ donde buscar el mínimo local de $ANLL$ en función del número de trayectorias l y del sistema de coordenadas. Para determinar estos rangos de $\%KNN$, hemos observado cómo se comporta el algoritmo variando l para ambos sistemas de coordenadas. Podemos observarlos en la Tabla 1.

l	$\%KNN (u, v)$	$\%KNN (r, \theta, \Phi)$
2	{0,60, 0,65, 0,70}	{0,50, 0,60, 0,70}
3	{0,45, 0,50, 0,55}	{0,40, 0,50, 0,60}
[4, 8]	{0,35, 0,40, 0,45}	{0,30, 0,40, 0,50}
[9, 11]	{0,30, 0,35, 0,40}	{0,20, 0,30, 0,40}
[12, 20]	{0,25, 0,30, 0,35}	{0,20, 0,30, 0,40}

Tabla 1: Valores de $\%KNN$ en función del número de trayectorias l , sobre los que minimizaremos el $ANLL$.

A pesar de que elijamos $\%KNN$ en función de l , esto no le da ventaja al PQC respecto a DBSCAN, ya que este método emula el proceso que seguiríamos si analizáramos cada problema individualmente para elegir el parámetro observando que el resultado fuera lógico. Es decir, lo que estamos haciendo es buscar un mínimo en el $ANLL$ restringiendo el rango de $\%KNN$ para evitar aquellos resultados que no sean válidos (como la solución trivial), lo cual se podría hacer en base a una observación de los datos sin necesidad de conocer l . Automatizando el proceso, podremos obtener resultados más significativos al aumentar la muestra.

3.2. Evaluación de los resultados

Para la evaluación de los resultados obtenidos necesitamos una métrica que puntúe la similitud entre dos agrupamientos diferentes: los grupos verdaderos y los grupos dados por el algoritmo. Esta métrica será el índice de Rand ajustado.

Dado un conjunto de n elementos $S = \{o_1, \dots, o_n\}$ y dos agrupaciones de estos elementos $X = \{X_1, \dots, X_r\}$, $Y = \{Y_1, \dots, Y_s\}$, definimos los siguientes valores:

- a : número de pares de elementos de S que están en el mismo subconjunto de X y en el mismo subconjunto de Y
- b : número de pares de elementos de S que están en diferentes subconjuntos de X y en diferentes subconjuntos de Y
- c : número de pares de elementos de S que están en el mismo subconjunto de X y en diferentes subconjuntos de Y
- d : número de pares de elementos de S que están en diferentes subconjuntos de X y en el mismo subconjunto de Y

El índice de Rand (no ajustado) se define como:

$$IR = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}} \quad (30)$$

No obstante, el índice de Rand, que toma valores entre 0 y 1, no tiene en cuenta las posibles coincidencias aleatorias, por lo que no asigna valores cercanos a cero a agrupamientos aleatorios. Por esa razón, se corrige con el índice de Rand ajustado:

$$IRA = \frac{IR - \langle IR \rangle}{\max(IR) - \langle IR \rangle} \quad (31)$$

donde $\max(IR) = 1$ y $\langle IR \rangle$ es el valor esperado del índice de Rand para un agrupamiento aleatorio que tiene el mismo número de grupos que el que hemos obtenido (se calcula obteniendo IR para un número suficientemente grande de agrupamientos aleatorios).

De esta forma, $IRA \approx 0$ si el agrupamiento es aleatorio y $IRA = 1$ si coincide con el agrupamiento verdadero (además, puede tomar valores negativos si el agrupamiento es peor que el aleatorio). Para implementar esta métrica, usaremos también la librería Scikit-Learn [13].

4. Resultados

4.1. Diagrama HR

En la Figura 15 vemos el resultado de aplicar los tres algoritmos de agrupamiento a nuestro conjunto de datos, con sus correspondientes *IRA*.

El algoritmo K-medias, Figura 15a, ha dado un resultado satisfactorio debido a la forma de los tres grupos que hay: son grupos relativamente separados y convexos. Sin embargo, hay que tener en cuenta que conocíamos la información de cuántos grupos había ($K = 3$), por lo que este algoritmo puede no ser útil si nos enfrentamos a problemas donde no conoczamos K . Además, hemos visto que para distribuciones de puntos más anisotrópicas, su rendimiento sería peor.

El algoritmo DBSCAN, Figura 15b, ha identificado bien los tres grupos. Su menor *IRA*, si lo comparamos con los otros dos algoritmos, se debe a la asignación de muchos puntos como ruido (color negro) y que, por tanto, no pertenecen a ningún grupo. Esto se podría corregir con un ε mayor o aumentando *minPts*.

El algoritmo PQC, Figura 15c, ha separado bien los grupos, pero observamos algo diferente al resto: ha separado las estrellas de la secuencia principal en dos partes (grupos 1 y 3), lo que quiere decir que, al calcular el potencial, se formaron dos pozos en esa zona en lugar de uno. Al igual que en DBSCAN, podríamos corregirlo tomando un $\%KNN$ mayor (que conlleve también un mínimo local en el *ANLL*). Sin embargo, utilizaremos esto para ilustrar cómo funcionan las jerarquías en el PQC.

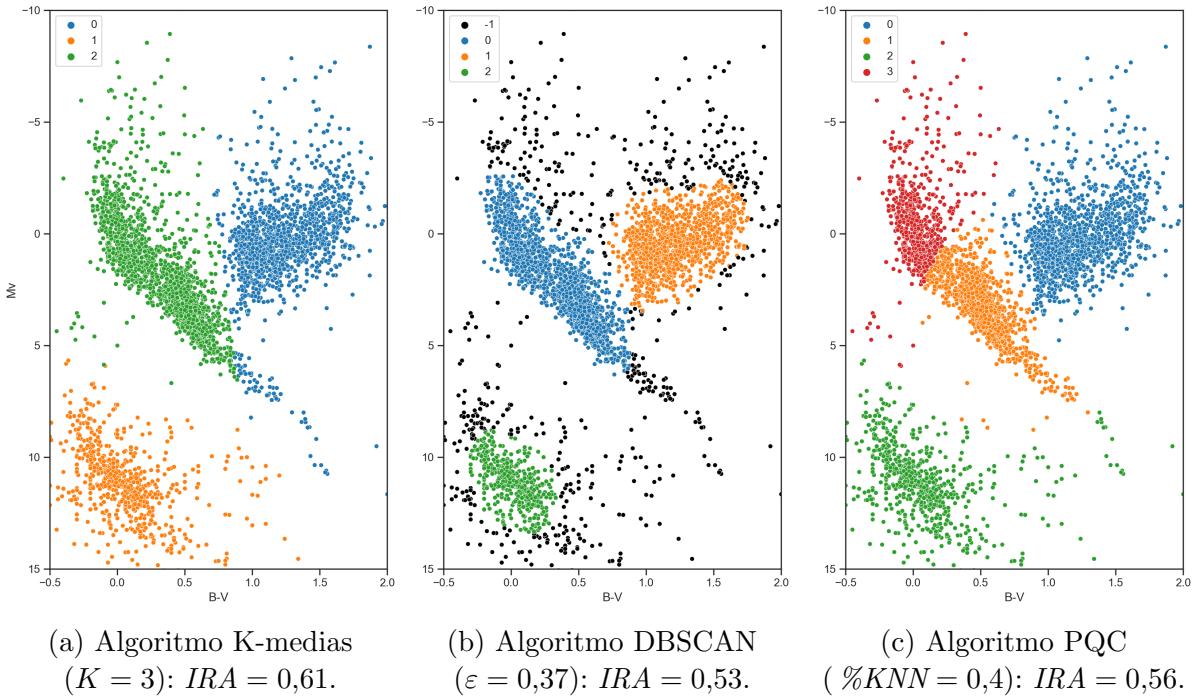


Figura 15: Resultados de los algoritmos de agrupamiento para nuestro Diagrama HR.

Como hemos visto anteriormente, podemos ir aumentando la E en (25) e ir uniendo los grupos para ver su estructura y obtener nuevos agrupamientos. En la Figura 16 podemos observar los agrupamientos resultantes para cada fase, junto con sus E , *IRA* y *ANLL*; y el dendrograma correspondiente. Observamos que hay un descenso muy rápido del *ANLL*,

en la Figura 16d, en $E = 28$, que se corresponde con el agrupamiento de la Figura 16b. Este agrupamiento tiene el mayor *IRA* de todos los que hemos realizado, por lo que no solo hemos comprobado cuál es la jerarquía de nuestro conjunto de datos (las estrellas de secuencia principal son más similares a las gigantes que a las enanas blancas), sino que hemos obtenido un agrupamiento altamente satisfactorio.

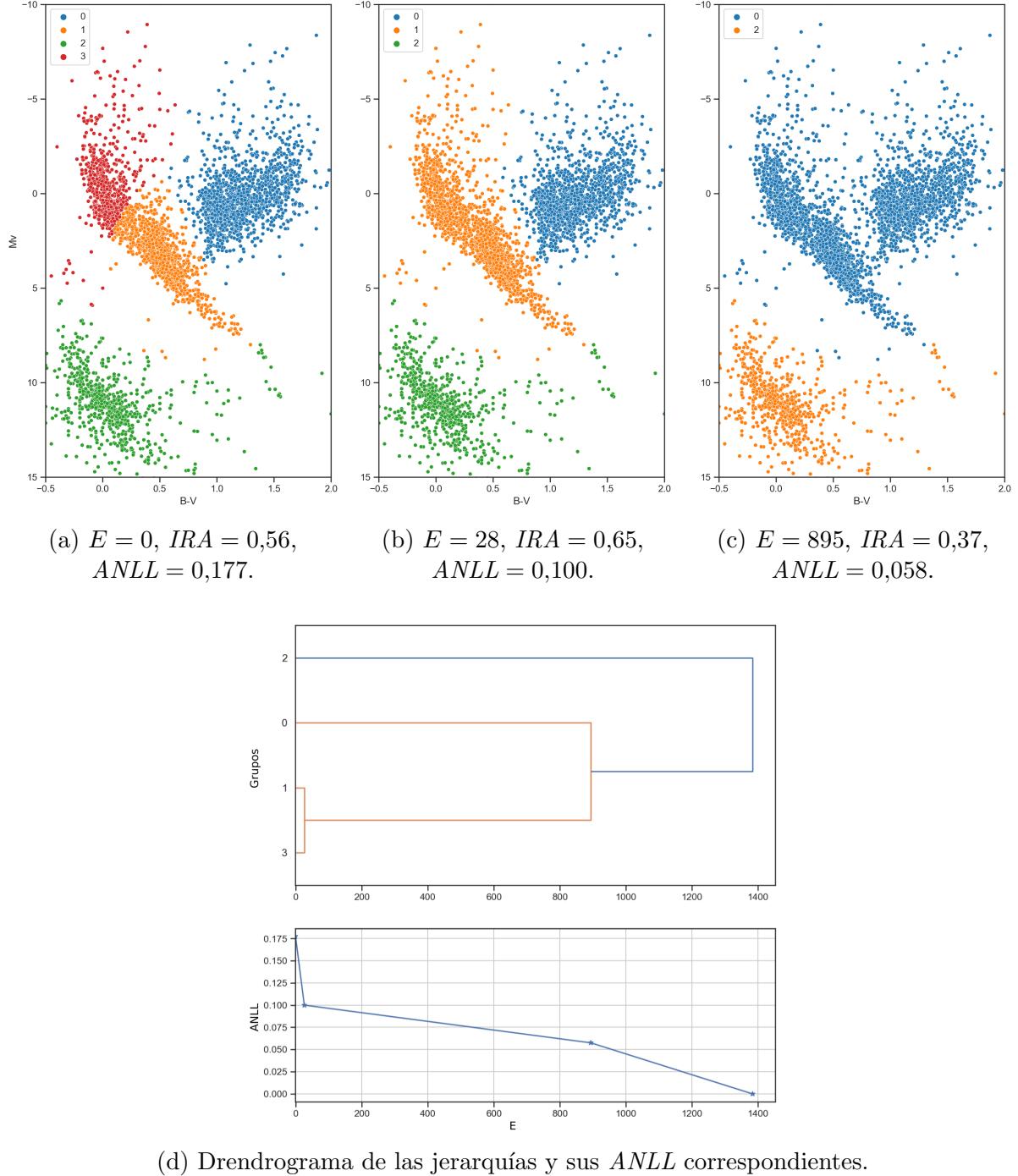


Figura 16: Agrupamiento jerárquico del PQC para nuestro conjunto de datos.

4.2. Seguimiento de partículas

En la Figura 17 vemos representado el *IRA* medio frente al número de trayectorias para ambas coordenadas, esféricas y (u, v) , usando los tres algoritmos de agrupamiento.

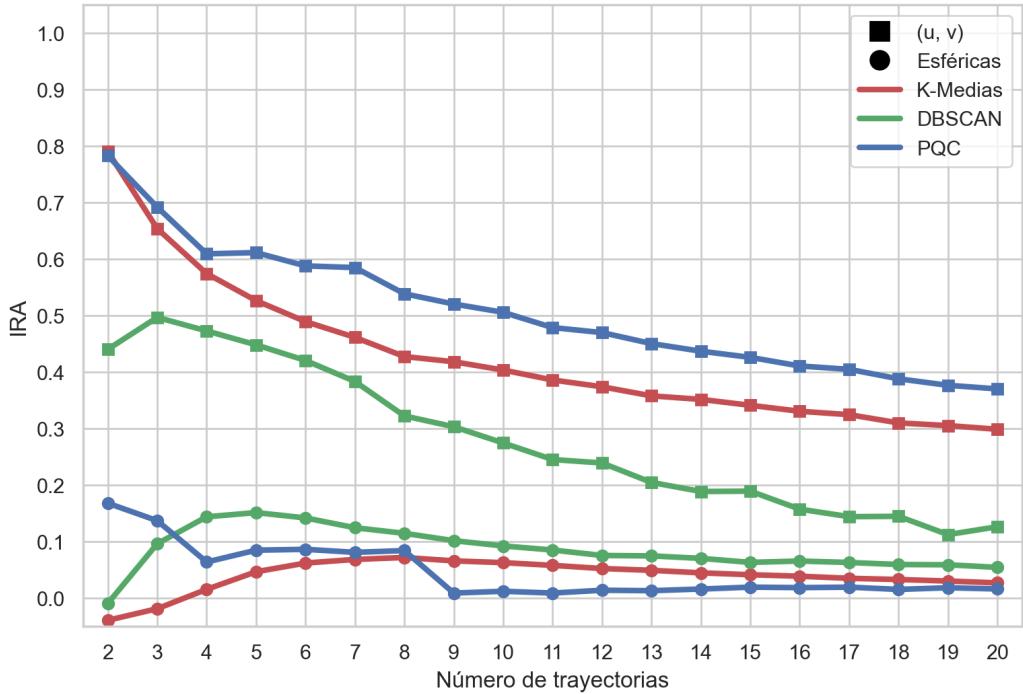


Figura 17: *IRA* frente al número de trayectorias para nuestro conjunto de datos.

En primer lugar, observamos que el *IRA* para las coordenadas esféricas es siempre inferior al de las coordenadas (u, v) que introdujimos, por lo que el tratamiento de los datos (en este caso, las transformaciones de coordenadas) ha sido de gran utilidad. En este problema, hemos podido realizar estas transformaciones ya que las coordenadas seguían trayectorias de la forma (28), por lo que un conocimiento físico del problema es muy importante para el resultado.

En segundo lugar, observamos cómo el rendimiento de todos los algoritmos cae conforme añadimos más trayectorias, ya que estas tienen parámetros similares y hay una mayor superposición de puntos.

En último lugar, respecto a la comparación de rendimientos entre los tres algoritmos, son similares para las coordenadas esféricas. Sin embargo, para las coordenadas (u, v) hay diferencias significativas. El algoritmo K-medias se sitúa entre DBSCAN y PQC, pero hay que recordar que le estábamos dando la información del número de grupos verdadero, por lo que podría no ser tan eficiente para un problema real donde no conoczamos K . Después, el algoritmo DBSCAN rinde peor que el resto, pero tiene dos grandes ventajas: su flexibilidad (en un problema real, podríamos afinar más la elección de ε y $minPts$, mejorando los resultados obtenidos con el método del codo) y su rapidez (comparado con PQC). Finalmente, vemos que el PQC es el algoritmo que mayor *IRA* presenta para todo l (excepto para el caso más simple $l = 2$, donde es similar a K-medias). Además, es flexible como DBSCAN ya que nos guía en la elección de los parámetros a usar (gracias al *ANLL*) y funciona con distribuciones de formas variadas.

5. Conclusiones / *Conclusions*

En este trabajo hemos comprobado la utilidad de la física en la creación de algoritmos de aprendizaje automático estudiando el agrupamiento cuántico probabilístico (PQC). A su vez, hemos aplicado este y otros dos algoritmos de agrupamiento (K-medias y DBSCAN) a dos problemas físicos, obteniendo resultados satisfactorios.

Estos dos problemas físicos se han beneficiado del agrupamiento por dos características principales: son problemas que involucran un número elevado de datos (haciendo necesario el uso de la informática) y cuyo objetivo es clasificar las observaciones en diferentes grupos. Por tanto, es interesante afrontar aquellos problemas físicos que comparten estas características con algoritmos de aprendizaje automático.

El problema del diagrama HR nos ha permitido visualizar las funcionalidades del agrupamiento cuántico probabilístico con un problema sencillo y de gran popularidad en física, viendo como no solo da buenos resultados, sino que nos aporta información más profunda acerca de la estructura jerárquica de los grupos. Estas características y su alta flexibilidad lo hacen útil para el estudio de conjuntos de datos complejos.

Además, en los dos problemas hemos comparado el rendimiento del PQC con los algoritmos K-medias y DBSCAN, siendo el rendimiento del PQC superior en ambos. Este algoritmo inspirado en física no solo es funcional y aporta herramientas útiles, sino que tiene un gran rendimiento, lo que motiva su uso tanto en problemas físicos como no físicos.

Finalmente, hemos comprobado que unos grupos poco definidos o con cierta superposición son causa de un peor rendimiento para los algoritmos de agrupamiento. Sin embargo, si el problema que estamos tratando es físico, podemos ayudarnos de conocimientos en física y matemáticas para transformar dicho conjunto de datos en uno donde haya una mejor separación de grupos; como hemos hecho en el problema de seguimiento de partículas, donde las transformaciones de coordenadas han sido cruciales para lograr un buen resultado.

In this work we have tested the usefulness of physics in the creation of machine learning algorithms by studying probabilistic quantum clustering (PQC). At the same time, we have applied this and two other clustering algorithms (K-means and DBSCAN) to two physical problems, obtaining satisfactory results.

These two physical problems have benefited from clustering because of two main characteristics: they are problems that involve a large number of data (making the use of computer science necessary) and whose objective is to classify the observations into different groups. Therefore, it is interesting to face those physics problems that share these characteristics with machine learning algorithms.

The HR diagram problem has allowed us to visualize the functionalities of probabilistic quantum clustering with a simple and popular problem in physics, seeing how it not only gives good results, but also provides us with deeper information about the hierarchical structure of the groups. These characteristics and its high flexibility make it useful for the study of complex datasets.

Furthermore, in the two problems we have compared the performance of PQC with the K-means and DBSCAN algorithms, with the performance of PQC being superior in both. This physics-inspired algorithm is not only functional and provides useful tools, but also has high performance, which motivates its use in both physical and non-physical problems. Finally, we have found that poorly defined or somewhat overlapping clusters cause worse performance for the clustering algorithms. However, if the problem we are dealing with is

physical, we can help ourselves with knowledge in physics and mathematics to transform that dataset into one where there is a better separation of groups; as we have done in the particle tracking problem, where coordinate transformations have been crucial to achieve a good result.

Referencias

- [1] J. J. Hopfield. «Neural networks and physical systems with emergent collective computational abilities». En: *Proceedings of the National Academy of Sciences* (1982), págs. 2554-2558.
- [2] Stephen G. Brush. «History of the Lenz-Ising Model». En: *Reviews of Modern Physics* 39 (1967), págs. 883-893.
- [3] Ethan Crouse. *Hopfield Networks: Neural Memory Machines*. [Visitado en mayo de 2024]. 2022. URL: <https://towardsdatascience.com/hopfield-networks-neural-memory-machines-4c94be821073>.
- [4] Stuart P. Lloyd. «Least square quantization in PCM». En: *Bell Telephone Laboratories Paper* (1957).
- [5] Gareth James et al. *An Introduction to Statistical Learning*. Springer, 2023.
- [6] Martin Ester et al. «A density-based algorithm for discovering clusters in large spatial databases with noise». En: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Ed. por Evangelos Simoudis, Jiawei Han y Usama M. Fayyad. AAAI Press, 1996, págs. 226-231.
- [7] Michael Hahsler, Matthew Piekenbrock y Derek Doran. «DBSCAN: Fast Density-Based Clustering with R». En: *Journal of Statistical Software* 91.1 (2019), págs. 1-30.
- [8] D Horn y A Gottlieb. «The Method of Quantum Clustering». En: *Advances in Neural Information Processing Systems 14*. Ed. por Thomas G. Dietterich, Suzanna Becker y Zoubin Ghahramani. The MIT Press, 2002, págs. 769-776.
- [9] Raúl V. Casaña-Eslava et al. «A Probabilistic framework for Quantum Clustering». En: *Knowledge-Based Systems* 194 (abr. de 2020). ISSN: 0950-7051. arXiv: [1902.05578 \[stat.ML\]](https://arxiv.org/abs/1902.05578).
- [10] Y. Bengio y P Vincent. «Manifold Parzen Windows». En: *CIRANO Working Papers* 15 (ene. de 2004).
- [11] K. B. Petersen y M. S. Pedersen. *The Matrix Cookbook*. Version 20121115. Nov. de 2012. URL: <http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html>.
- [12] D. P. Kingma y J. B. Adam. «Adam: A Method for Stochastic Optimization». En: *International Conference on Learning Representations (ICLR) 15* (2015).
- [13] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.
- [14] Richard Powell. *Diagrama de Hertzsprung-Russell*. Traducido por Alvaro qc. 2008. URL: <https://commons.wikimedia.org/wiki/File:HRDiagram-es.png>.
- [15] Wing-Fung Ku. *Star Categorization Giants And Dwarfs Dataset*, vinesmsuic. <https://www.kaggle.com/datasets/vinesmsuic/star-categorization-giants-and-dwarfs>. Jul. de 2020.

- [16] M. Wenger et al. «The SIMBAD astronomical database: The CDS reference database for astronomical objects». En: *Astronomy and Astrophysics Supplement Series* 143.1 (abr. de 2000), págs. 9-22. ISSN: 1286-4846. DOI: [10.1051/aas:2000332](https://doi.org/10.1051/aas:2000332).
- [17] Sabrina Amrouche et al. «The Tracking Machine Learning Challenge: Accuracy Phase». En: *The Springer Series on Challenges in Machine Learning*. Springer International Publishing, nov. de 2019, págs. 231-264. ISBN: 9783030291358. DOI: [10.1007/978-3-030-29135-8_9](https://doi.org/10.1007/978-3-030-29135-8_9). URL: <https://www.kaggle.com/c/trackml-particle-identification/>.
- [18] Uraz Odyurt et al. *Novel Approaches for ML-Assisted Particle Track Reconstruction and Hit Clustering*. 2024. arXiv: [2405.17325 \[hep-ex\]](https://arxiv.org/abs/2405.17325).