**Clustering Bayesiano No-Paramétrico**

Este documento se ha redactado para proporcionar bases conceptuales generales y específicas que aporten a una mayor comprensión en este tópico en particular.

Autor: Kevin Blanco Madariaga

1. **Clustering: Definiciones, Limitaciones y Desafíos**

**1.1 Introducción**

El análisis de conglomerados o "clustering" es esencialmente un método estadístico multivariante que busca agrupar elementos y/o variables tratando de lograr la *máxima similitud* en cada grupo y la mayor heterogeneidad entre los grupos. En el ámbito del machine learning se establece como una técnica de aprendizaje no supervisada (Jain, 2010), el cual se ha utilizado en variadas aplicaciones, como por ejemplo en la astronomía para descubrir tipos de estrellas mediante la agrupación de mediciones astrofísicas; geociencias para detectar campos minados o fallas sísmicas a partir de datos espaciales; procesamiento de lenguaje natural en el modelado de tópicos; biomedicina para descubrir grupos de individuos o genes con patrones similares en expresión génica o datos ómicos, entre otros (Wade, 2023). Típicamente, la "*similitud*" en el clustering se define en términos de una métrica de distancia apropiada, que podría ser Euclidiana, Manhattan, Coseno, Jaccard, entre otras; dependiendo del tipo de datos y del problema específico. Los clústeres resultantes pueden ser utilizados para entender la estructura subyacente de los datos, para resumir los datos, o como una etapa de preprocesamiento para otras técnicas de análisis de datos (Tan et al., 2005).

Existen varios tipos de técnicas de clustering que se pueden categorizar en función de su enfoque hacia la agrupación. Entre las principales, se encuentran:

* **Clustering particional:** En esta categoría, los datos se dividen en grupos no superpuestos donde cada dato pertenece exactamente a un grupo. El ejemplo más popular de este tipo es el algoritmo K-means (MacQueen, 1967).
* **Clustering jerárquico:** Este tipo de clustering busca construir una jerarquía de clústeres, ya sea mediante un enfoque "bottom-up" (aglomerativo) o "top-down" (divisivo). Un ejemplo de este tipo es el clustering aglomerativo jerárquico (Manning et al., 2008).
* **Clustering basado en densidad:** Estos algoritmos consideran los clústeres como áreas de alta densidad en el espacio de datos. DBSCAN es un ejemplo de un algoritmo de clustering basado en densidad (Ester et al., 1996).
* **Clustering basado en red:** Estos algoritmos tratan los datos como una red y utilizan teoría de grafos para identificar clústeres. Un ejemplo es el algoritmo de propagación de afinidad (Frey & Dueck, 2007).

A pesar de su utilidad, el clustering no está exento de desafíos. La definición de *"similitud"* puede ser ambigua y depende en gran medida del contexto del problema. La elección de la métrica de distancia, el número de clústeres (en algoritmos particionales) y otros parámetros puede tener un impacto significativo en los resultados de clustering. Además, los algoritmos de clustering pueden ser sensibles a la inicialización y caer en mínimos locales, lo que significa que no siempre producen los mismos resultados en diferentes ejecuciones. La cantidad, la calidad, la complejidad y la diversidad de los datos se han vuelto cada vez más críticas. Aquí se examinan los desafíos más significativos que surgen en el uso de los métodos de clustering.

* **Elección de la métrica de similitud:** La elección de una métrica de similitud adecuada es uno de los desafíos más críticos en el clustering. Las métricas comúnmente utilizadas como la distancia euclidiana, la distancia de Manhattan o la similitud del coseno, no siempre pueden capturar la naturaleza subyacente de los datos. Este problema se exacerba cuando los datos son de alta dimensión o cuando las características son de tipos diferentes, es decir, datos heterogéneos (alta variación).
* **Determinación del número de clusters**: Uno de los principales retos en el clustering es determinar el número óptimo de clusters. La elección de este número puede influir en gran medida en la calidad del resultado. Métodos tradicionales como el método del codo o el índice de Silhouette, muchas veces no logran dar un número óptimo de clusters, especialmente en conjuntos de datos de alta dimensión y complejos.
* **Clustering en alta dimensión:** El clustering en alta dimensión es un problema desafiante debido a la "maldición de la dimensionalidad". A medida que aumenta la dimensión, la distancia entre los puntos tiende a converger, lo que hace que todas las distancias parezcan igualmente similares o disímiles, degradando el rendimiento del algoritmo. Este problema afecta especialmente a los algoritmos basados en distancias.
* **Manejo de ruido y datos atípicos:** La presencia de ruido y datos atípicos puede tener un efecto negativo considerable en los resultados del clustering. Algunos algoritmos son particularmente sensibles a esto, como el K-Means. Las técnicas para el manejo de ruido y datos atípicos son esenciales, pero su implementación puede aumentar la complejidad computacional y el tiempo de ejecución.
* **Clustering de datos categóricos**: Los datos categóricos presentan un desafío particular en el clustering. Muchos algoritmos de clustering se desarrollaron inicialmente para datos continuos y no manejan adecuadamente los datos categóricos. Definir una "distancia" significativa entre los puntos en un espacio categórico es un problema complejo que aún está en investigación.
* **Interpretación y evaluación de los clusters:** La interpretación de los clusters puede ser subjetiva y depende del dominio de aplicación. La evaluación de la calidad de un clustering es un desafío en sí mismo, particularmente en el caso de clustering no supervisado, donde no hay una "verdad" conocida contra la cual comparar los resultados. Es difícil determinar cuándo un agrupamiento es "bueno".
* **Escalabilidad y eficiencia:** En la era del Big Data, los métodos de clustering deben ser escalables y eficientes. Muchos algoritmos de clustering tradicionales no escalan bien con el tamaño del conjunto de datos. A medida que los conjuntos de datos crecen en tamaño y complejidad, el tiempo de computación y los recursos necesarios para el clustering también aumentan, lo que puede hacer que ciertos métodos sean impracticables.
* **Clustering de datos no estructurados:** El clustering de datos no estructurados, como textos y/o imágenes es un desafío debido a su complejidad y a la dificultad de definir una noción de similitud. Las técnicas de clustering que funcionan bien para datos numéricos o categóricos pueden no funcionar tan bien para este tipo de datos.

En general, más allá de sus desafíos, el clustering sigue siendo una herramienta valiosa para la ciencia de datos y el aprendizaje automático, y la investigación en estas áreas continúa ofreciendo nuevas técnicas y enfoques para abordar estos problemas, desde una perspectiva paramétrica y no-paramétrica.

**1.2 Clustering Paramétrico**

Los métodos de clustering paramétricos asumen que los datos se distribuyen según un modelo específico. Este tipo de técnicas de clustering necesita que se especifique de antemano el número de clústeres y se ajustan a los datos según ciertos parámetros, a menudo usando un criterio de optimización. El ejemplo más común de un método paramétrico de clustering es el algoritmo de K-means. Este método parte de una estimación inicial de los centroides de los clústeres y luego asigna cada punto de datos al clúster cuyo centroide es el más cercano. Los centroides se recalculan en función de los puntos asignados, y este proceso se repite hasta que los clústeres ya no cambien o cambien muy poco (MacQueen, 1967). Los métodos de clustering paramétricos, dentro del campo del aprendizaje automático, ofrecen un enfoque para agrupar datos en base a modelos estadísticos definidos. Aunque estos métodos, como las mezclas gaussianas, pueden ser poderosos y precisos, se enfrentan a una serie de desafíos únicos en términos de implementación y optimización. Entre ellos se encuentran:

* **Elección de la distribución subyacente:** Uno de los primeros desafíos en los métodos de clustering paramétricos es determinar la distribución subyacente adecuada. Este desafío surge del hecho de que los modelos paramétricos hacen suposiciones específicas sobre la distribución de los datos. Si la suposición de distribución es incorrecta, el rendimiento del modelo puede degradarse significativamente.
* **Estimación de parámetros:** La estimación de parámetros es otra área desafiante en los métodos de clustering paramétricos. La eficacia de estos métodos depende en gran medida de la precisión con la que se pueden estimar los parámetros del modelo subyacente. La elección de un algoritmo de optimización para la estimación de parámetros, como el algoritmo Expectation-Maximization (EM), puede ser crítica y puede influir en el rendimiento final del modelo.
* **Inicialización de parámetros:** Los métodos de clustering paramétricos suelen requerir una inicialización de los parámetros del modelo, que puede afectar significativamente el rendimiento del algoritmo. Muchos algoritmos de optimización, como el EM, son sensibles a la inicialización y pueden converger a mínimos locales en lugar de a un mínimo global, lo que resulta en un clustering subóptimo.
* **Determinación del número de clusters:** Al igual que con otros métodos de clustering, la determinación del número de clusters es un desafío en los métodos paramétricos. Este problema se magnifica en estos métodos debido a las suposiciones adicionales que hacen sobre la estructura de los datos. Métodos como el criterio de información bayesiano (BIC) o el criterio de información de Akaike (AIC) se utilizan a menudo, pero estos también tienen sus limitaciones y pueden no proporcionar siempre el número óptimo de clusters.
* **Manejo de la dimensionalidad:** Los métodos de clustering paramétricos pueden ser especialmente susceptibles a la denominada "*maldición de la dimensionalidad*". Con un gran número de dimensiones, la estimación de los parámetros del modelo puede volverse muy difícil, y los algoritmos pueden sufrir de sobreajuste. Esto puede llevar a un rendimiento pobre en conjuntos de datos de alta dimensión.
* **Sensibilidad al ruido y a los datos atípicos:** Al igual que con otros métodos de clustering, los métodos paramétricos pueden ser sensibles al ruido y a los datos atípicos. Como estos métodos hacen suposiciones sobre la distribución de los datos, los datos atípicos o el ruido pueden tener un gran impacto en la estimación de los parámetros del modelo y, por ende, en la calidad del clustering.
* **Suposición de independencia:** Muchos métodos de clustering paramétricos hacen la suposición de que los datos en diferentes clusters son independientes. Sin embargo, en muchas aplicaciones del mundo real, esta suposición puede no ser válida. Esto puede llevar a resultados de clustering subóptimos y es un área de investigación activa en la comunidad de aprendizaje automático.
* **Escalabilidad y eficiencia:** Los métodos de clustering paramétricos pueden enfrentar desafíos de escalabilidad. Dado que estos métodos suelen requerir un proceso iterativo para la estimación de parámetros, pueden no ser adecuados para conjuntos de datos extremadamente grandes. Las técnicas de optimización para mejorar la eficiencia y la escalabilidad son un área de investigación activa.

**1.3 Clustering No Paramétrico**

Por otro lado, los métodos de clustering no paramétricos no hacen suposiciones tan fuertes sobre la estructura subyacente de los datos. En lugar de requerir que se especifique de antemano el número de clústeres, los métodos no paramétricos suelen ser capaces de determinar el número adecuado de clústeres a partir de los datos. Un ejemplo de un método no paramétrico de clustering es el algoritmo DBSCAN. En lugar de requerir que se especifique el número de clústeres, DBSCAN considera dos parámetros: el radio espacial máximo ('eps') y el número mínimo de puntos necesarios para formar un clúster ('minPts'). Luego clasifica los puntos de datos como puntos centrales, puntos de borde o ruido en función de estos parámetros, permitiendo la detección de clústeres de formas y tamaños arbitrarios (Ester et al., 1996). Los métodos de clustering no-paramétricos, que incluyen técnicas como el DBSCAN, Mean-Shift, y el Clustering Jerárquico, han ganado popularidad en los últimos años debido a su flexibilidad y adaptabilidad. Sin embargo, estos métodos también enfrentan una serie de desafíos únicos en el contexto del aprendizaje automático. Entre ellos se encuentran:

* **Elección de parámetros de algoritmo:** Aunque los métodos de clustering no-paramétricos no asumen una distribución específica para los datos, a menudo requieren la selección de parámetros específicos del algoritmo, como el radio de vecindad en DBSCAN o la función de kernel y el ancho de banda en Mean-Shift. La elección de estos parámetros puede tener un impacto significativo en los resultados del clustering, y no existe una regla general para la selección de estos parámetros que sea óptima en todos los contextos.
* **Sensibilidad a la densidad:** Los métodos de clustering no-paramétricos suelen ser sensibles a la densidad de los datos. Esto significa que pueden tener dificultades para identificar clusters que tienen densidades de puntos significativamente diferentes. En la práctica, esto puede llevar a la fusión de clusters que deberían ser separados o a la fragmentación de un cluster en varios subgrupos.
* **Manejo de alta dimensión:** Los métodos de clustering no-paramétricos, al igual que otros métodos de clustering, pueden verse afectados por la "maldición de la dimensionalidad". En particular, muchos de estos métodos dependen del concepto de "distancia" entre los puntos, lo cual puede ser problemático en espacios de alta dimensión donde la distancia entre los puntos puede converger.
* **Robustez frente a ruido y datos atípicos:** El ruido y los datos atípicos pueden tener un impacto significativo en los resultados de los métodos de clustering no-paramétricos. Aunque algunos métodos, como DBSCAN, son inherentemente robustos al ruido, otros pueden verse afectados de manera significativa. El desarrollo de técnicas para manejar el ruido y los datos atípicos es un área de investigación activa.
* **Interpretación de los clusters**: A diferencia de los métodos de clustering paramétricos, los métodos no-paramétricos no proporcionan un modelo explícito para los clusters, lo que puede hacer que la interpretación de los resultados sea más desafiante. Esto es especialmente relevante en aplicaciones donde la interpretación de los clusters es importante.
* **Determinación del número de clusters**: Aunque algunos métodos de clustering no-paramétricos, como Mean-Shift, pueden determinar automáticamente el número de clusters, este sigue siendo un desafío en muchos casos. Por ejemplo, en el clustering jerárquico, la elección del punto de corte en el dendrograma puede afectar significativamente el número de clusters identificados.
* **Eficiencia computacional y escalabilidad:** Muchos de estos métodos tienen una complejidad computacional alta, lo que puede hacer que sean impracticables para conjuntos de datos muy grandes. El desarrollo de algoritmos más eficientes y de técnicas de aproximación es una área de investigación activa.

1. **El Enfoque Bayesiano No Paramétrico**

La principal diferencia entre los métodos paramétricos y no paramétricos radica en las suposiciones que hacen sobre los datos. Los métodos paramétricos asumen un modelo específico para la distribución de los datos, lo que puede limitar su capacidad para adaptarse a los datos si estas suposiciones no se cumplen. Por otro lado, los métodos no paramétricos no hacen tales suposiciones, pero pueden ser más desafiantes en términos de elección de parámetros y computacionalmente más costosos. En ambos casos, los enfoques algorítmicos más utilizados son en gran parte heurísticos y no se basan en modelos formales para determinar el número de clústers, porque carecen de medidas de incertidumbre durante el proceso de clustering. Ante eso, un enfoque alternativo que se propone son los métodos de clustering bayesianos no-paramétricos que incorporan información previa sobre los parámetros y permiten evaluar la incertidumbre en la estructura de agrupamiento de los datos, basado en la formalización de un modelo. Siendo de gran utilidad cuando no conocemos a priori el número de clusters en los datos (incertidumbre), o cuando este número puede cambiar con la adición de nuevos datos (densidad/variación).

En esta discusión, nos centraremos principalmente en el Proceso de Dirichlet (DP), una de las técnicas más comunes en el clustering bayesiano no-paramétrico. El Proceso de Dirichlet (DP), desarrollado por Thomas Ferguson en 1973 (Ferguson, 1973), es un proceso estocástico que se utiliza para definir distribuciones de probabilidad sobre espacios de dimensión infinita. En el contexto del clustering, el DP se utiliza para definir una distribución a priori sobre el número y las propiedades de los clusters en los datos. La esencia del clustering bayesiano no-paramétrico utilizando DP radica en su flexibilidad. En lugar de predefinir un número fijo de clusters, este método asume que los datos están generados por un número potencialmente infinito de clusters. En la práctica, sin embargo, sólo un número finito de estos clusters se representará en cualquier conjunto de datos finito (Rasmussen, 2000). Por ejemplo, un modelo DP de mezcla gaussiana (DPGMM) asume que cada punto de datos es generado por una de las posiblemente infinitas gaussianas. El modelo comienza con un solo clúster y, a medida que observa más datos, decide si asigna cada nuevo dato a un clúster existente o inicia un nuevo clúster (Neal, 2000). La decisión se basa en las propiedades de los clusters existentes, la similitud del nuevo dato con los clusters existentes, y un parámetro de concentración que controla la tendencia del modelo a generar nuevos clusters. A continuación se detallan los procesos implicados bajo esta perspectiva:

1. **Definición del Proceso de Dirichlet**

El primer componente clave en el clustering bayesiano no-paramétrico es el Proceso de Dirichlet (DP). Un DP se define como DP(α, ), donde α es un parámetro de concentración y es una distribución de base. Un Proceso de Dirichlet se puede entender como una "distribución sobre distribuciones". Si se toma una muestra G ~ DP(α, ), G es una distribución de probabilidad. Esta distribución G es discreta con probabilidad 1, lo que significa que si se considera una muestra θ ~ G, y se repite este proceso, se observará que muchos de los θ son iguales. Es precisamente esta propiedad la que nos permite usar DP para clustering, donde θ puede representar los parámetros de un clúster.

En otras palabras, en un DP, los parámetros están distribuidos de acuerdo con una distribución base , y el DP es controlado por el parámetro de concentración α. El DP se denota como: DP(α, ). Esto significa que si dibujamos un conjunto infinito de parámetros de la DP, estos parámetros estarán distribuidos de acuerdo con , pero la presencia del parámetro α permite que los parámetros se agrupen, es decir, varios pueden ser iguales, lo que da lugar a clusters en los datos.

1. **Distribución Predictiva**

El aspecto central del DP que facilita el clustering es su distribución predictiva. Si se considera un conjunto de n observaciones {, ..., } muestreadas de G, entonces la distribución de una nueva observación θ\* {dado , ..., } es:

P(θ\* = ) = / (α + N), para i = 1,...,N

P(θ\* es nueva) = α / (α + N)

donde es el número de observaciones que son iguales a . Esto significa que hay una probabilidad proporcional a de que θ\* sea igual a una observación existente , y una probabilidad proporcional a α de que θ\* sea una nueva observación.

1. **Modelo de Mezcla de Dirichlet**

En el clustering bayesiano no-paramétrico utilizando el Proceso de Dirichlet, los datos se modelan como una mezcla de distribuciones, donde la distribución de la mezcla está dada por un Proceso de Dirichlet (DP). Ello se denomina modelo de mezcla de Dirichlet (DPM). En un DPM, cada observación de datos x está asociada con un parámetro θ. La idea es que los x que tienen el mismo θ son parte del mismo cluster. Es decir, primero supongamos que los datos se generan a partir de una distribución de la mezcla. Esto significa que cada observación de datos, denotada como x, se genera de una de K posibles distribuciones, cada una parametrizada por un conjunto de parámetros . Por ejemplo, en una mezcla de gaussianas, incluiría la media y la varianza de la k-ésima gaussiana.

Para cada observación i, tenemos:

~ G

| ~ F() → siendo F alguna distribución de probabilidad.

En otras palabras, podemos expresar esto como:

x ~ f(x | ) → siendo f la función de densidad de probabilidad de la distribución correspondiente.

Entonces, podemos combinar lo anterior para obtener un modelo de mezcla de Dirichlet. En este modelo, cada observación de datos se genera primero seleccionando un parámetro del DP, y luego generando la observación de la distribución parametrizada por . Esto se puede expresar como:

~ DP(α, ) → x ~ f(x | )

1. **Inferencia**

Bajo este enfoque se busca realizar la inferencia inversa. Dado un conjunto de observaciones {,..., }, queremos averiguar qué observaciones pertenecen al mismo clúster. Es decir, se desea inferir los parámetros y la asignación de cada observación a uno de los parámetros. Esto se puede hacer mediante técnicas de muestreo para inferencia bayesiana (como el Muestreo de Gibbs o las cadenas de Markov Monte Carlo), donde iterativamente se muestrea la distribución posterior de cada parámetro y asignación de cluster, dado los datos y todas las demás variables. En la práctica, se trabaja con una versión truncada del DP, donde solo se permite un número finito máximo de clusters. Esto es computacionalmente más factible, y se ha demostrado que proporciona una buena aproximación al DP completo. Por ejemplo, para explicar lo anterior centrémonos en el muestreo de Gibbs para la inferencia en DP. El muestreo de Gibbs es un algoritmo iterativo que selecciona un parámetro y actualiza su valor muestreándolo de la distribución posterior condicional, dadas todas las demás variables y los datos. Dado que el número de clusters en un DP es potencialmente infinito, se introduce una variable auxiliar de asignación de cluster para cada observación, denotada por , que indica a qué cluster pertenece la i-ésima observación. Para cada observación , iteramos a través de las siguientes etapas:

1. Actualizamos el parámetro del clúster al que pertenece , muestreando de la distribución posterior condicional p( | , , z, α), donde representa todos los otros parámetros de clúster.
2. Actualizamos la asignación de cluster de , muestreando de la distribución posterior condicional p( | , θ, α).
3. Se repiten estas etapas hasta que el proceso converja.

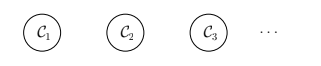
Esta versión truncada del DP, llamada Proceso de Dirichlet Truncado (TDP), pone un límite al número máximo de clusters. Esto es computacionalmente más eficiente y proporciona una buena aproximación al DP completo. Los detalles de la implementación pueden variar, pero en general, se introducen variables auxiliares adicionales para permitir que se añadan y se retiren clusters de forma flexible.

En resumen, el clustering bayesiano no-paramétrico utilizando el Proceso de Dirichlet es un enfoque muy flexible que no requiere especificación del número de clusters por adelantado. Los mismos datos guían hacia la mejor elección del número de clusters. Utilizar el DP implica modelar los datos como una mezcla de distribuciones, donde la distribución de la mezcla está dada por un DP. La inferencia se realiza mediante técnicas de muestreo para inferencia bayesiana, iterativamente actualizando los parámetros de los clusters y las asignaciones de los clusters (considerando un TDP para limitar el número de clusters). Aunque estos métodos proporcionan una gran flexibilidad, también requieren un manejo cuidadoso. En particular, el ajuste de los parámetros, como el parámetro de concentración en el DP, puede ser difícil y tiene un impacto significativo en los resultados del clustering. Además, estos métodos pueden ser computacionalmente intensivos, especialmente en conjuntos de datos de alta dimensión o de gran tamaño. Además, aunque el DP proporciona una forma de modelar la incertidumbre sobre el número de clusters, este modelo asume que los datos son intercambiables, es decir, que el orden de los datos no importa. En muchas aplicaciones del mundo real, esta suposición puede no ser válida. Para superar esta limitación, se han desarrollado numerosas extensiones del DP, como el Proceso de Dirichlet jerárquico (Teh et al., 2006) y el Proceso de Dirichlet dependiente del tiempo (Ahmed and Xing, 2008).

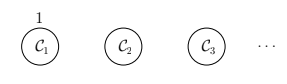
1. **El Proceso del Restaurante Chino (CRP)**

Los modelos de mezcla del proceso de Dirichlet (DPM) se han convertido en una herramienta valiosa en el aprendizaje automático moderno. Las mezclas de DP pueden describirse a través del *proceso del restaurante chino* (CRP), que es una distribución sobre particiones (subpoblaciones) que incorpora la supuesta distribución a priori sobre estructuras de agrupación (Aldous, 1985; Pitman, 2002). El CRP pertenece a la familia de los modelos de mezcla de Dirichlet y describe cómo se pueden formar clusters a medida que los datos llegan secuencialmente (Escobar & West, 1995). De manera metafórica el CRP se explica mediante una secuencia de clientes (datos) sentados en las mesas de un restaurante chino (clusters). En este caso, cada cliente se sienta en una mesa previamente ocupada con una probabilidad proporcional al número de clientes ya sentados allí, y en una nueva mesa con una probabilidad proporcional a un parámetro de concentración. En una mezcla de CRP, los clientes se identifican con puntos de datos, y los datos que están en la misma mesa pertenecen al mismo grupo. Dado que el número de mesas ocupadas es aleatorio, esto proporciona un modelo flexible en el que el número de grupos está determinado por los datos. Los clientes de un CRP son intercambiables, bajo cualquier permutación de su orden, la probabilidad de una configuración particular es la misma, y esta propiedad es esencial para conectar la mezcla de CRP con la mezcla de DP.

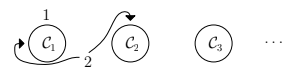
Para efectos de esquematizar este proceso, supongamos que tenemos infinitas "mesas de restaurante" , , . . . Estas mesas son lo suficientemente grandes para que un número arbitrario de personas pueda sentarse en cada mesa:



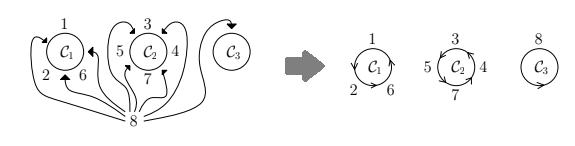
Posteriormente, una infinidad de clientes , , , . . . entran en el restaurante, en una secuencia de uno a la vez. Es decir, el Cliente se coloca en la mesa C1:



Con igual probabilidad de 1/2, se puede colocar el Cliente en la misma mesa que el Cliente o quedar solo en la nueva mesa C2:



Con igual probabilidad de 1/3, se puede colocar el Cliente en la misma mesa que el Cliente o en la mesa del Cliente , o solo en la primera mesa vacía …y así sucesivamente. Es decir, supongamos que los clientes {1, 2, . . . , n − 1} ya están instalados. Con igual probabilidad 1/n, se puede colocar el Cliente n al lado de 1, . . . , o al lado de n − 1, o solo en la mesa vacía (en la imagen n = 8):



Como se mencionó anteriormente, el proceso de Dirichlet es una distribución sobre distribuciones, y la mezcla de DP asume que los parámetros aleatorios que gobiernan las observaciones se extraen de una distribución derivada de un proceso de Dirichlet. Las observaciones son condicionalmente independientes dada la distribución aleatoria, y por lo tanto deben ser marginalmente intercambiables. Si la mezcla de CRP no produjera una distribución intercambiable, no podría ser equivalente a una mezcla de DP. La intercambiabilidad es una suposición razonable en algunas aplicaciones de agrupación, pero en muchas no lo es. El principal beneficio del clustering utilizando CRP es su flexibilidad y adaptabilidad. No es necesario especificar de antemano el número de clusters o categorías, ya que estas se determinan de forma dinámica a medida que se presentan los datos. Esto lo convierte en una herramienta muy útil cuando se trabaja con conjuntos de datos grandes o complejos donde el número de clusters no es conocido a priori, y puede cambiar con el tiempo o con la llegada de nuevos datos. Además, el proceso puede manejar de manera efectiva tanto clusters de tamaños similares como aquellos con tamaños muy diferentes, lo que lo hace más robusto frente a la variabilidad de los datos. Sin embargo, hay que tener en cuenta que, aunque esta flexibilidad es una ventaja, también puede ser una desventaja en ciertas situaciones. Por ejemplo, si hay ruido o datos atípicos en los datos, el CRP puede acabar creando clusters no deseados. Además, la elección del parámetro de concentración puede tener un gran impacto en los resultados, lo que requiere un cuidadoso análisis y afinamiento, para así evitar producir un número excesivo de clusters pequeños, que pueden no ser útiles en algunos contextos (Blei & Jordan, 2006).

**3.1 Algoritmo de CRP**

El CRP es un proceso estocástico que modela una secuencia de decisiones de partición de los datos. Es decir, es un proceso de prioridad para distribuciones de Dirichlet y proporciona una manera elegante de modelar la incertidumbre sobre el número de clusters presentes en los datos (Pitman, 2002; Blackwell & MacQueen, 1973). Utilizando la metáfora CRP, supongamos que hay n clientes en el restaurante y queremos saber a qué mesa se sentará el cliente n+1:

* **Paso 1:** Primero, se define un parámetro de concentración α. Este parámetro controla el grado en que se forman nuevos clusters.

Nota: *El CRP se basa en dos parámetros: El número total de clientes ya sentados (parámetro n) y un parámetro de concentración α, que controla la probabilidad de que un nuevo cliente comience un nuevo cluster.*

* **Paso 2:** El cliente n+1 elige una mesa para sentarse. La probabilidad P de que el cliente n+1 elija la mesa k (donde la mesa k ya tiene clientes sentados) está dada por:

P(mesa k) = / (n + α)

Donde es el número de clientes ya sentados en la mesa k, y n es el número total de clientes ya en el restaurante.

* **Paso 3:** También hay una probabilidad de que el cliente n+1 elija sentarse en una nueva mesa (un nuevo cluster). Esta probabilidad viene dada por:

P(nueva mesa) = α / (n + α)

* **Paso 4:** Estos pasos se repiten para cada nuevo cliente (o punto de datos) que entra en el restaurante.

Este es un proceso de generación de particiones donde la probabilidad de pertenecer a un clúster existente es proporcional al número de miembros ya en el cluster, y la probabilidad de iniciar un nuevo clúster es constante. A partir de este proceso, podemos derivar un conjunto de particiones de nuestros datos, lo que da lugar a un agrupamiento de los mismos. Este proceso de particionamiento es especialmente útil en escenarios donde el número de clusters no se conoce a priori, ya que permite la creación de nuevos clusters a medida que llegan más datos. Finalmente, es importante mencionar que el CRP es un modelo que, en la práctica, suele ser necesario utilizar técnicas como el Muestreo de Gibbs o las cadenas de Markov Monte Carlo (MCMC), para ajustar estos modelos a conjuntos de datos reales.

**3.2 Algoritmo CRP con Muestreo de Gibbs y/o Cadenas de Markov Monte Carlo**

El muestreo de Gibbs y las Cadenas de Markov Monte Carlo (MCMC) son técnicas comúnmente utilizadas para realizar inferencia en modelos de clustering basados en CRP. En el contexto del CRP, ambos se utilizan para actualizar iterativamente las asignaciones de cluster para cada punto de datos. El CRP proporciona un modelo probabilístico para la asignación de puntos de datos a clusters, mientras que el muestreo Gibbs o el MCMC se utilizan para realizar la inferencia en este modelo. Además, en el caso del MCMC, la idea básica detrás de su uso es crear una secuencia o "cadena" de asignaciones de cluster que, en el límite, sigue la distribución de probabilidad deseada. En el caso del CRP, la distribución de probabilidad de interés es la distribución posterior de las asignaciones de cluster dada la información previa y los datos observados. A continuación se presenta una simplificación del algoritmo CRP utilizando muestreo de Gibbs o MCMC:

1. **Inicialización:** Comienza con una asignación inicial de los datos { i } a los clusters. Esta asignación puede ser aleatoria o basada en alguna heurística. Por ejemplo, cada punto de datos { i } podría ser asignado inicialmente a su propio clúster.
2. **Iteración Gibbs y/o MCMC**: En cada iteración del algoritmo usando Gibbs o MCMC, se selecciona un punto de datos { i } al azar y se propone reasignarlo a un nuevo clúster. Las posibilidades para este nuevo cluster incluyen tanto los clusters existentes como un nuevo cluster.
3. **Cálculo de probabilidades:** Calcula la probabilidad de que el punto de datos seleccionado se asigne a cada uno de los clusters posibles, utilizando las reglas de distribución de probabilidad definida por el CRP:

* Para un clúster existente k, la probabilidad es / (n + α), donde es el número de puntos de datos ya asignados a ese cluster y N es el total de puntos de datos.
* Para un nuevo cluster, la probabilidad es α / (n + α), donde α es un parámetro de concentración.

1. **Propuesta y aceptación/rechazo:** Se elige un cluster para la reasignación del punto de datos { i } de acuerdo con estas probabilidades. Además, se calcula la probabilidad de aceptación de la nueva asignación, la cual se basa en la probabilidad de los datos dados los parámetros del modelo para cada clúster. Si la nueva asignación es aceptada, actualiza los clústers; si no, se deja la asignación como estaba.
2. **Repetición:** Se repiten los pasos 2 a 4 durante un número predefinido de iteraciones, o hasta que la asignación de los datos a los clusters converja.
3. **Salida:** Una vez que el algoritmo ha convergido o se ha alcanzado el número máximo de iteraciones, la salida es la asignación final de los datos a los clusters.

A través de este procedimiento que utiliza el muestro de Gibbs o el MCMC se permite explorar el espacio de todas las posibles asignaciones de cluster y, en última instancia, obtener muestras de la distribución posterior. Estas muestras se pueden utilizar para estimar las asignaciones de cluster más probables y otras cantidades de interés. La combinación de CRP con Gibbs o MCMC proporcionan una solución más robusta y flexible para problemas de clustering en los que el número de clusters no se conoce a priori. Sin embargo, como con cualquier método de inferencia bayesiana, su implementación puede ser computacionalmente intensiva y requiere una cuidadosa consideración de las condiciones iniciales y los parámetros del modelo.

1. **Implementación de CRP en Python**

La implementación del CRP en la práctica puede llevarse a cabo utilizando Python. Para efectos de ejemplificar su potencial desarrollo, asumamos que nuestros datos son unidimensionales y que cada clúster se modela como una distribución normal con una media desconocida y una varianza conocida. Utilizaremos la perspectiva de CRP con muestreo Gibbs y los paquetes de python: numpy para cálculos numéricos y matplotlib para la visualización:

Comenzaremos importando las bibliotecas necesarias:

|  |
| --- |
| import numpy as np import matplotlib.pyplot as plt |

Luego, vamos a generar algunos datos de prueba para la demostración. Supongamos que los datos provienen de tres clústeres con diferentes medias.

|  |
| --- |
| np.random.seed(0) # Para reproducibilidad cluster\_means = [-1, 0, 3] n\_data\_per\_cluster = 50 data = np.concatenate([np.random.randn(n\_data\_per\_cluster) + mean for mean in cluster\_means]) |

Ahora, definimos una función para ejecutar un solo paso del muestreo de Gibbs para la computación del CRP:

|  |
| --- |
| def gibbs\_sampling\_step(data, cluster\_assignments, cluster\_sizes, alpha, sigma\_x, sigma\_0):  # Elegimos un dato al azar para reasignar  i = np.random.choice(len(data))  x\_i = data[i]   # Lo quitamos de su clúster actual  k\_old = cluster\_assignments[i]  cluster\_sizes[k\_old] -= 1  if cluster\_sizes[k\_old] == 0: # Si el clúster está vacío, lo eliminamos  del cluster\_sizes[k\_old]   # Calculamos la probabilidad de cada clúster (existente y uno nuevo)  existing\_cluster\_probs = [size / (len(data) - 1 + alpha) for size in cluster\_sizes.values()]  new\_cluster\_prob = alpha / (len(data) - 1 + alpha)  all\_cluster\_probs = existing\_cluster\_probs + [new\_cluster\_prob]   # Elegimos un nuevo clúster para el dato  k\_new = np.random.choice(len(all\_cluster\_probs), p=all\_cluster\_probs)  if k\_new == len(all\_cluster\_probs) - 1: # Si elegimos el nuevo clúster  k\_new = max(cluster\_sizes.keys()) + 1 if cluster\_sizes else 0 # Asignamos un nuevo índice de clúster  cluster\_assignments[i] = k\_new  cluster\_sizes[k\_new] = cluster\_sizes.get(k\_new, 0) + 1   return cluster\_assignments, cluster\_sizes |

Finalmente, inicializamos las asignaciones de clúster al azar y ejecutamos el muestreo de Gibbs para un cierto número de pasos:

|  |
| --- |
| # Parámetros n\_steps = 200 alpha = 1.0 sigma\_x = 1.0 # Varianza conocida de cada clúster sigma\_0 = 1.0 # Varianza del prior para la media de cada clúster  # Inicializar asignaciones de clúster al azar cluster\_assignments = np.random.choice(len(data), size=len(data)) cluster\_sizes = {k: np.sum(cluster\_assignments == k) for k in range(len(data))}  # Ejecutar muestreo de Gibbs for step in range(n\_steps):  cluster\_assignments, cluster\_sizes = gibbs\_sampling\_step(  data, cluster\_assignments, cluster\_sizes, alpha, sigma\_x, sigma\_0)  # Visualizar las asignaciones finales de clúster plt.scatter(data, [0]\*len(data), c=cluster\_assignments) plt.title('Asignaciones finales de clúster') plt.show() |

En general, desarrollar un modelo de clustering en Python usando el CRP con Muestreo de Gibbs y/o Cadenas de Markov, se debe tener especial cuidado en seleccionar las distribuciones a priori adecuadas, inicializar correctamente el estado de la cadena y la selección de parámetros. También es importante considerar el tiempo de convergencia, verificar dicha convergencia y tener un método para evaluar la calidad del modelo de clustering (una desventaja de las técnicas de muestreo de Monte Carlo, como el Muestreo de Gibbs, es que pueden requerir un tiempo de convergencia largo). Finalmente, la utilización eficaz de bibliotecas de Python como NumPy, SciPy, pymc3 y sklearn son importantes para la implementación y evaluación de estos modelos.

1. **Referencias**

Jain, A. K. (2010). "Data clustering: 50 years beyond K-means". Pattern Recognition Letters, 31(8), 651-666.

MacQueen, J. B. (1967, June). "Some methods for classification and analysis of multivariate observations." In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).

Ester, M., Kriegel, H. P., Sander, J., Xu, X. (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 226-231.

Ferguson, T. (1973). A Bayesian Analysis of Some Nonparametric Problems. The Annals of Statistics, 1(2), 209-230.

Rasmussen, C. E. (2000). The infinite Gaussian mixture model. In Advances in neural information processing systems (pp. 554-560).

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. Journal of computational and graphical statistics, 9(2), 249-265.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical dirichlet processes. Journal of the american statistical association, 101(476).

Ahmed, A., & Xing, E. P. (2008). Dynamic non-parametric mixture models and the recurrent Chinese restaurant process: with applications to evolutionary clustering. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on (pp. 219-228). IEEE.

Aldous, D. (1985). Exchangeability and related topics. École d'Été de Probabilités de Saint-Flour XIII — 1983, 1–198.

Blei, D. M., & Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. Bayesian Analysis, 1(1), 121–143.

Escobar, M. D., & West, M. (1995). Bayesian density estimation and inference using mixtures. Journal of the American Statistical Association, 90(430), 577–588.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. ACM Computing Surveys (CSUR), 31(3), 264–323.

Pitman, J. (2002). Combinatorial stochastic processes. Lecture Notes for St. Flour Summer School.

Aldous, D. (1985). Exchangeability and related topics. In École d’été de probabilités de Saint-Flour XIII—1983 (pp. 1-198). Springer.

MacEachern, S.N., & Müller, P. (1998). Estimating Mixture of Dirichlet Process Models. Journal of Computational and Graphical Statistics, 7(2), 223-238.

Wade S. (2023). Bayesian cluster analysis. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, 381(2247), 20220149.

Tan, P., Steinbach, M. and Kumar, V. (2005) Cluster Analysis: Basic Concepts and Algorithms. In: Introduction to Data Mining, Addison-Wesley, Boston.

C. D. Manning, P. Raghavan, and H. Schtze. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. Science (New York, N.Y.), 315(5814), 972–976. https://doi.org/10.1126/science.1136800