Txmins resumen y comentarios importantes:

Se trata de un algoritmo de clustering sin intervención humana donde no se le pasan parámetros, a diferencia que en el tkmeans donde se le tenia que pasar el numero de clusters y el Y para formar el representante.

Se introduce el concepto de Mass Clustering, que se refiere a la necesidad de realizar clustering individualizado y masivo sobre múltiples datasets diferentes.

Este algoritmo esta inspirado en xmeans, xmeans no esta hecho para transacciones

Ventajas de txmeans:

Detecta automáticamente el número de clústeres.

Genera "representative transactions", que resumen los patrones clave de cada clúster.

Usa un enfoque top-down divisivo:

Empieza con un solo clúster.

Lo divide iterativamente en dos sub-clústeres.

Determina los representantes de los clústeres (baskets representativos) mediante una técnica descrita en [12].

Decide si dividir más un clúster usando el Bayesian Information Criterion (BIC) adaptado a datos transaccionales.

Escalable y eficiente para datasets de millones de usuarios.

A continuación desarrolla los trabajos previos:

Captura de pantalla de un teléfono celular

Descripción generada automáticamente

Txmeans extrae centroides, maneja datos raros como comunes , tiene una estrategia top-down no minimiza funciones de costo

Los conceptos técnicos relevantes son entropía que es la medida de desorden usada en funciones de costo para evaluar similitud,

Subspace clustering enfocado en subconjuntos de dimensiones relevantes

Co-clustering agrupa filas y columnas simultaneamnete

Ahora nos plantean el problema

Y plantea el algoritmo que lo soluciona

Algoritmo basado en una estrategia top-down y de divide y vencerá

Estrategia divisiva: parte de un único cluster que parte de todas ls transacciones y de forma iterativa intenta dividir cada cluster en dos subclusteres hasta que ya no se justifique una nueva partición.

No solo se busca obtener los clústeres sino que se extrae un representante de cada uno que sintetice los ítems característicos del grupo

Para dividir un cluster se usa Bayesian Information Criterion (BIC)

El algoritmo funciona de la siguiente manera:

1 se extrae el representante

2 se inicializa una cola q en la que se inserta el par hB y r hb es el conjunto de baskets iniciales y r el representante

2 bucle: se saca de la cola un cluster con su representante, se eliminan los ítems comunes el objetivo es eliminar información redundante que, al ser común a todas las transacciones, no ayuda a discriminar entre diferentes patrones.

Luego se aplica la fuincion que bisecciona el cluster en dos donde se obtienen dos clústeres y sus respectivos representante

Evaluaciojn mediante bic, se calculka el bic del cluster original y del conjunto formado por los dos subclusyeres

Con esta evaluación se decide si se divide en estos dos subclustyeres o no, si es si se reinsertan los ítems comunes en cada uno de los subclusteres y en sus representantes para restaurar el contexto original y se añaden ambos subclusters a la cola, en caso de que se decidad no dividir se considera que ekl cluster actual ya es correcto,

El proceso se repite hasta que la cola quede vacia

4.2 Txmeans Stopping Criterion

En este punto se describe el algoritmo de parada que decide si la divisison de un cluister en dos sub-clusteres es útil en términos de mejorar la homogeneidad interna

Se utiliza el Bic para evaluar el ajuste del modelo antes y después del Split

Antes de proceder al Split se identifican los elementos comunes en as cestas y se eliminan

En el apartado 4.3 va a explicar como divide los clusters en 2.

Calcula el representante calculando los ítems no comunes y luego define i como la unión de los ítems que quedan después de eliminar los comunes y para cada item I se cuanta la frecuencia

Ahora se itera sobre los itmes de I, se selecciona el item con mayor frecuencia, se actualiza el representante añadiendo el item con mayor frecuencia ahora se calcula el SSE, si este sse no mejora al anterior, que el anterior era uuno inicializado a cero, por lo tanto la primera vez lo tiene que mejorar siempre si lo mejora se elimina m y se continua

Ahora te explica el algoritmo par dividir los dos subgrupos.

I=0 y sse alto, se eligen dos representantes iniciales para la división mediante la fuincion donde se seleccionana de manera alejada dos centroides. Luego se van separando todos las transacciones según su distancia a los representantes, como es obvio se ira al cluster donde su representante tiene una distancia menor, posteriormente se recalcula el representante y se calculla sse, se continiua haciendo esto hasta que el sse obtenido sea mayor que el de la ejecución pasada.

Indicadores de rendimiento:

NMI: se usa para medir la similitud entre dos agrupaciones: oscila entre 0 y 1 donde 1 indica que son iguales

Tiene lagunas ya que puede estar influenciada por el tamaño finito de las redes, rNMI que es normalizada

Después calcula la desviación diferencia entre el numero de clusters detectados y el numero de cluster reales

El tiempo de ejecución se mide el tiempo en segundos que tarda el algoritmo en ejecutarse.

Competidores:

1 Practical: Algoritmo sin parámetros basado en una función de costo inspirada en *TF-IDF*. Funcionamiento: Primero escanea los datos y asigna cada transacción (basket) a un cluster existente o crea un nuevo cluster. Luego, ajusta los clusters moviendo transacciones de un cluster a otro. Diferencia con TxMeans: La estructura del algoritmo es completamente distinta. TxMeans sigue una estrategia divisiva, mientras que Practical asigna y reajusta.

2. ATDC Algoritmo divisivo, similar a TxMeans. Funcionamiento: Escanea las transacciones (baskets). Itera entre una fase de particionamiento y una de estabilización. Diferencia con TxMeans: Aunque es divisivo, su metodología de iteración en dos fases es diferente.

3. TKMeans: Versión adaptada de k-means para transacciones.Funcionamiento: Modifica la definición de distancia en k-means para medir disimilitud entre transacciones. Calcula los centroides con la misma técnica que TxMeans. Diferencia con TxMeans: Aunque ambos algoritmos usan centroides, TxMeans sigue un enfoque divisivo, mientras que TkMeans es una adaptación de k-means tradicional.

4. CoolCat: Basado en muestreo aleatorio. Funcionamiento: Trabaja con una muestra aleatoria de las transacciones antes de agruparlas. Diferencia con TxMeans: Ambos usan muestreo, pero CoolCat depende exclusivamente de la muestra inicial, mientras que TxMeans sigue refinando los clusters.

5. Clope Enfoque: Diseñado específicamente para datos de market-basket. Funcionamiento: Requiere un parámetro de repulsión ( 𝑟 r), que es difícil de interpretar. Diferencia con TxMeans: Su parametrización puede ser un desafío, mientras que TxMeans es más automatizado.

Se excluyen algunos algoritmos por ser superados en rendimiento por otros métodos o por estar diseñados para otros tipos de datos: ROCK, SubCAD, LIMBO, Clicks, LargeItem → Superados por Practical, ATDC y Clope.

K-Modes, DHCC → Diseñados para datos categóricos, no transacciones.

PurTreeClust → Se centra en la segmentación de usuarios en lugar de transacciones.

**. Evaluación en Datasets Reales (UCI Repository)**

* Se prueban tres datasets reales: **Mushrooms, Congressional Votes y Zoo**.
* **TxMeans obtiene resultados consistentes y estables**, superando a varios algoritmos en distintas métricas.
* En **Mushrooms**, solo **Clope y TxMeans tienen buenos resultados**.
* En **Zoo**, **TxMeans supera incluso a los métodos con parámetros** correctamente especificados.
* En **Congressional Votes**, el desempeño de **TxMeans es comparable** al de sus competidores.

He realizado varios cambios en el código prpuesto del TXmeans, debido a la diferencia de las versiones de kas librerías. No tenia la posibilidad de instalar las librerías que habían usado ellos debido a incompatibilidades con la versión de Python usada actualmente,

El primer camboio que se ha realizado ha sido sobre la función read\_uci\_data

Donde debido a una incompatibilidad con el versionadop de la liobnreria scipy se ha tenido que vcanmbiar la línea de codigfo  mode\_value = df[k].mode()[0], en esta función se encarga de reemplazar todos lois espacios vacíos en el dataset con el valor mas común de esa columna.

Por otro lado se ha cambiado la función \_first\_iter que usa el modelo para buscar el primer centroide en la primera iteración, en este caso estaba dando error la siguiente línea sample\_baskets\_keys = set(random.sample(list(baskets.keys()), self.random\_sample)) la cual elige de manera aleatoria una transacción de la cesta. Daba error debido a una incompatibilidad con la función random en la variable baskets keys, para solucionarlo se ha pasado a lista y ha funcionado.

Por ultimo se ha modificado la función \_find\_medioids() d = np.inf donde esta línea estaba haciendo reprerencia a una función pasada de numpy.