CLARANS: un método para agrupar objetos

para minería de datos espaciales

Raymond T. Ng y Jiawei Han, Miembro, IEEE Computer Society

Resumen —La minería de datos espaciales es el descubrimiento de relaciones y características interesantes que pueden existir implícitamente en las bases de datos espaciales. Para ello, este trabajo tiene tres contribuciones principales. Primero, proponemos un nuevo método de agrupamiento llamado CLARANS, cuyo objetivo es identificar estructuras espaciales que pueden estar presentes en los datos. Los resultados experimentales indican que, en comparación con los métodos de agrupación existentes, CLARANS es muy eficiente y eficaz. En segundo lugar, investigamos cómo CLARANS puede manejar no solo objetos puntuales, sino también objetos poligonales de manera eficiente. Uno de los métodos considerados, llamado aproximación IR, es muy eficiente para agrupar objetos poligonales convexos y no convexos. En tercer lugar, sobre la base de CLARANS, desarrollamos dos algoritmos de minería de datos espaciales que tienen como objetivo descubrir relaciones entre atributos espaciales y no espaciales.

Términos del Índice —Minería de datos espaciales, algoritmos de agrupamiento, búsqueda aleatoria, geometría computacional.

1 yo NTRODUCCIÓN

re Aquí, la minería en general es la búsqueda de patrones ocultos que pueden existir en grandes bases de datos, de datos espaciales en particular, es el descubrimiento de relaciones y características interesantes que pueden existir implícitamente en las bases de datos espaciales. Debido a las enormes cantidades (generalmente, terabytes) de datos espaciales que pueden obtenerse de imágenes de satélite, equipos médicos, cámaras de video, etc., es costoso y, a menudo, poco realista para los usuarios examinar los datos espaciales en detalle. La minería de datos espaciales tiene como objetivo automatizar dicho proceso de descubrimiento de conocimiento. Por tanto, juega un papel importante en

1. extraer patrones y características espaciales interesantes, capturar
2. relaciones intrínsecas entre datos espaciales y no espaciales,
3. Presentar la regularidad de los datos de forma concisa y a niveles conceptuales superiores, y
4. ayudando a reorganizar las bases de datos espaciales para adaptarse a la semántica de datos, así como para lograr un mejor rendimiento.

Se han realizado muchos estudios excelentes sobre minería de datos, como los descritos en [2], [3], [6], [14], [20], [23], [26]. Agrawal y col. considere el problema de inferir funciones de clasificación a partir de muestras [2] y estudie el problema de las reglas de asociación minera entre conjuntos de elementos de datos [3]. Han y col. proponer un enfoque orientado a atributos para el descubrimiento de conocimientos [14]. Y el libro editado por Shapiro y Frawley incluye muchos estudios interesantes sobre diversos temas en el descubrimiento del conocimiento, como encontrar dependencias funcionales entre atributos [26]. Sin embargo, la mayoría de estos estudios están relacionados con el descubrimiento de conocimientos sobre datos no espaciales y el trabajo más relevante para nuestro enfoque aquí es el que se informa en [23]. Más específicamente, Lu et al. proponen un algoritmo dominante espacial y uno no espacial dominante para extraer relaciones de alto nivel entre datos espaciales y no espaciales. Sin embargo, ambos algoritmos adolecen de los siguientes problemas. Primero, el usuario o un experto debe proporcionar a los algoritmos jerarquías de conceptos espaciales, que pueden no estar disponibles en muchas aplicaciones. En segundo lugar, ambos algoritmos conducen su exploración espacial principalmente fusionando regiones en un cierto nivel de la jerarquía con una región más grande en un nivel superior. Por lo tanto, la calidad de los resultados producidos por ambos algoritmos depende de manera crucial de la adecuación de la jerarquía a los datos dados. El problema para la mayoría de aplicaciones es que es muy difícil saber a priori qué jerarquía será la más adecuada. Descubrir esta jerarquía puede ser en sí mismo una de las razones para aplicar la minería de datos espaciales.

El análisis de conglomerados es una rama de la estadística que, en las últimas tres décadas, se ha estudiado intensamente y se ha aplicado con éxito a muchas aplicaciones. Para la tarea de minería de datos espaciales en cuestión, el atractivo del análisis de conglomerados es su capacidad para encontrar estructuras o conglomerados directamente a partir de los datos dados, sin depender de jerarquías. Sin embargo, el análisis de clústeres se ha aplicado sin éxito en el pasado a la minería de datos general y al aprendizaje automático. Las quejas son que los algoritmos de análisis de conglomerados son ineficaces e ineficientes. De hecho, para que el análisis de conglomerados funcione de manera eficaz, existen los siguientes problemas clave:

. Si existe una noción natural de similitudes entre los "objetos" que se van a agrupar. Para la minería de datos espaciales, nuestro enfoque aquí es aplicar el análisis de conglomerados solo en los atributos espaciales. Si estos atributos corresponden a objetos puntuales, existen nociones naturales de similitudes (por ejemplo, distancias euclidianas o de Manhattan). Sin embargo, si los atributos corresponden a objetos poligonales, la situación es más complicada. Más específicamente, la similitud (o

. RT Ng trabaja en el Departamento de Ciencias de la Computación de la Universidad de Columbia Británica, Vancouver, BC, V6T 1Z4, Canadá. Correo electrónico: rng@cs.ubc.ca.
. J. Han trabaja en la Facultad de Ciencias de la Computación de la Universidad Simon Fraser, Burnaby, BC, V5A 1S6, Canadá.

Manuscrito recibido el 21 de noviembre de 1995; revisado el 6 de diciembre de 2000; aceptado el 9 mar. 2001.
Para obtener información sobre cómo obtener reimpresiones de este artículo, envíe un correo electrónico a: tkde@computer.org y haga referencia al número de registro IEEECS 105188.

distancia) entre dos objetos poligonales se pueden definir de muchas formas, algunas mejores que otras. Pero, las mediciones de distancia más precisas pueden requerir más esfuerzo de cálculo. Por tanto, la cuestión principal es el tipo de agrupación espacial que se está considerando, qué medición logra el mejor equilibrio.

. Si la agrupación de una gran cantidad de objetos se puede realizar de manera eficiente. Los algoritmos tradicionales de análisis de conglomerados no están diseñados para grandes conjuntos de datos, con más de 1.000 objetos, por ejemplo.

Al abordar estos problemas, informamos en este documento:

. el desarrollo de CLARANS, que tiene como objetivo utilizar la búsqueda aleatoria para facilitar la agrupación de una gran cantidad de objetos y
. un estudio sobre la eficiencia y efectividad de tres enfoques diferentes para calcular las similitudes entre objetos poligonales. Son el enfoque que calcula la distancia de separación exacta entre dos polígonos, el enfoque que sobreestima la distancia exacta usando la distancia mínima entre vértices y el enfoque que subestima la distancia exacta usando la distancia de separación entre los rectángulos isotéticos de los polígonos.

Para evaluar nuestras ideas y algoritmos, presentamos resultados, más a menudo experimentales que analíticos, que muestran que:

. CLARANS es más eficiente que los algoritmos existentes PAM y CLARA, los cuales motivan el desarrollo de CLARANS; y calcular la similitud entre dos polígonos utilizando la distancia de separación
. entre los rectángulos isotéticos de los polígonos es el enfoque más eficiente y efectivo.

En [25], presentamos un estudio preliminar de CLARANS y los dos algoritmos de minería de datos espaciales. Pero este artículo se extiende [25] de dos formas principales. Primero, CLARANS y los algoritmos de minería de datos se generalizan para admitir objetos poligonales. Como se motivó anteriormente, agrupar objetos poligonales de manera efectiva y eficiente no es nada sencillo. En segundo lugar, este artículo presenta un análisis más detallado y resultados experimentales sobre el comportamiento de CLARANS y sobre las formas de ajustar CLARANS para aplicaciones específicas.

Desde la publicación de [25], se han desarrollado muchos métodos de agrupamiento, que pueden clasificarse ampliamente en métodos de particionamiento [7], métodos jerárquicos [33], [12], [4], [18], métodos basados en densidad [11], [15] y métodos basados en cuadrículas [31], [29], [1]. En [7], Bradley et al. propone un algoritmo que sigue el marco básico del algoritmo K-means, pero que proporciona escalabilidad al comprimir inteligentemente algunas regiones del espacio de datos. En [33], [12], [4], [18], los métodos jerárquicos propuestos intentan detectar estructuras de agrupamiento anidado, que son frecuentes en algunas aplicaciones. En [11], [15], los métodos propuestos basados en la densidad intentan proporcionar una mejor agrupación para las agrupaciones alargadas; Los métodos de partición suelen ser mucho más adecuados para clústeres esféricos. En [31], [29], [1], espacio de datos para facilitar la agrupación.

Para comparar CLARANS con estos trabajos, hacemos las siguientes observaciones generales:

. Muchas de las técnicas mencionadas anteriormente requieren algunas estructuras de árbol o cuadrícula para facilitar la agrupación. En consecuencia, estas técnicas no se escalan bien con el aumento de la dimensionalidad de los conjuntos de datos. Si bien es cierto que el material discutido en este documento es predominantemente 2D, el algoritmo CLARANS funciona de la misma manera para conjuntos de datos de mayor dimensión. Debido a que CLARANS se basa en una búsqueda aleatoria y no utiliza ninguna estructura auxiliar, CLARANS se ve mucho menos afectado por el aumento de la dimensionalidad.

. Muchas de las técnicas mencionadas anteriormente asumen que la función de distancia es euclidiana. CLARANS, al ser una técnica de búsqueda local, no exige ningún requisito sobre la naturaleza de la función de distancia.

. Muchas de las técnicas mencionadas anteriormente tratan con objetos puntuales; CLARANS es más general y admite objetos poligonales. Una parte considerable de este documento está dedicada al manejo eficaz de objetos poligonales.

. CLARANS es una técnica de agrupamiento de memoria principal, mientras que muchas de las técnicas mencionadas anteriormente están diseñadas para aplicaciones de agrupamiento fuera del núcleo. Admitimos que siempre que se involucran operaciones extensas de E / S, CLARANS no es tan eficiente como los demás. Sin embargo, sostenemos que CLARANS todavía tiene una aplicabilidad considerable. Considere los objetos 2D que se discutirán en este documento. Cada objeto está representado por dos números reales, que ocupan un total de 16 bytes. Agrupar 1.000.000 de objetos requeriría un poco más de 16 Mbytes de memoria principal. Esta es una cantidad fácilmente asequible para una computadora personal, y mucho menos computadoras para minería de datos. El punto aquí es que, dado el muy bajo costo de la RAM, los algoritmos de agrupación en clústeres de la memoria principal, como CLARANS, no están completamente dominados por algoritmos fuera del núcleo para muchas aplicaciones. Finalmente, en una nota similar,

El documento está organizado de la siguiente manera: la Sección 2 presenta PAM y CLARA. La Sección 3 presenta nuestro algoritmo de agrupamiento CLARANS, así como los resultados experimentales que comparan el desempeño de CLARANS, PAM y CLARA. La sección 4 estudia y evalúa experimentalmente los tres enfoques diferentes que calculan las similitudes entre los objetos poligonales. La sección 5 concluye el documento con una discusión sobre los trabajos en curso.

2 C LUSTERING UN LGORITMOS segundo ASED EN PAGS ARTICIONAMIENTO

2.1 Resumen

En los últimos 30 años, el análisis de conglomerados se ha aplicado ampliamente a muchas áreas como la medicina (clasificación de enfermedades), la química (agrupación de compuestos), los estudios sociales (clasificación de hallazgos estadísticos), etc.

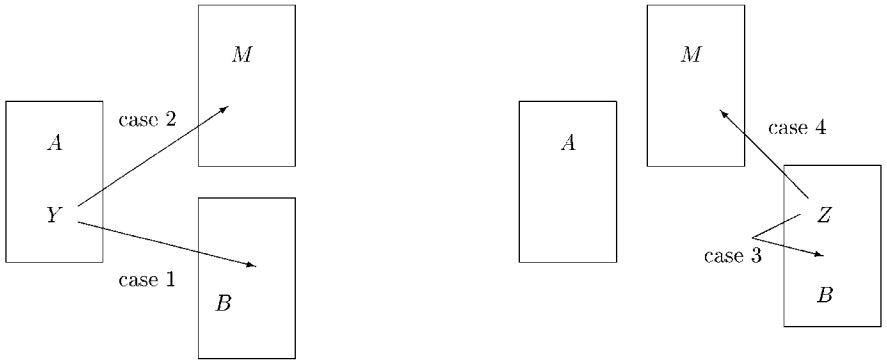

Fig. 1. Cuatro estuches para reemplazar UN con METRO.

El objetivo es identificar estructuras o racimos presente en los datos. Los algoritmos de agrupación existentes se pueden clasificar en dos categorías principales: jerárquico métodos y fraccionamiento
métodos. Los métodos jerárquicos son aglomerativos o divisivos. Dado norte objetos a agrupar, los métodos aglomerativos comienzan con norte agrupaciones (es decir, todos los objetos están separados). En cada paso, se eligen y combinan dos grupos. Este proceso continúa hasta que todos los objetos se agrupan en un grupo. Por otro lado, los métodos de división comienzan poniendo todos los objetos en un grupo. En cada paso, se elige un grupo y se divide en dos. Este proceso continúa hasta norte se producen racimos. Si bien los métodos jerárquicos se han aplicado con éxito a muchas aplicaciones biológicas (por ejemplo, para producir taxonomías de animales y plantas [19]), se sabe que adolecen de la debilidad de que nunca pueden deshacer lo que se hizo anteriormente. Una vez que un método aglomerativo fusiona dos objetos, estos siempre estarán en un grupo. Y una vez que un método divisivo separa dos objetos, estos objetos nunca se reagruparán en el mismo grupo.

En contraste, dado el número k de las particiones que se encuentran, un método de partición intenta encontrar la mejor k particiones [1] del norte objetos. Muy a menudo ocurre que el k Los clústeres encontrados por un método de partición son de mayor calidad (es decir, más similares) que los k agrupaciones producidas por un método jerárquico. Debido a esta propiedad, el desarrollo de métodos de partición ha sido uno de los principales enfoques de la investigación del análisis de conglomerados. De hecho, se han desarrollado muchos métodos de partición, algunos basados en k- medio, algunos en k- medoide

algunos sobre análisis difusos, etc. Entre ellos, hemos elegido k- medoide métodos como base de nuestro algoritmo por las siguientes razones. Primero, a diferencia de muchos otros métodos de particionamiento, el k- medoide Los métodos son muy robustos a la existencia de valores atípicos (es decir, puntos de datos que están muy lejos del resto de los puntos de datos). En segundo lugar, los clústeres encontrados por

k- medoide Los métodos no dependen del orden en que se examinan los objetos. Además, son invariantes con respecto a las traducciones y transformaciones ortogonales de puntos de datos. Por último, pero no menos importante, los experimentos han demostrado que k- medoide Los métodos descritos a continuación pueden manejar conjuntos de datos muy grandes con bastante eficacia. Consulte [19] para obtener una comparación más detallada de k- medoide métodos con otros métodos de particionamiento. En el resto de esta sección, presentamos los dos más conocidos k- medoide métodos en los que se basa nuestro algoritmo.

1. Las particiones aquí se definen de la forma habitual: cada objeto se asigna a exactamente un grupo.

2.2 PAM

PAM (Partitioning Around Medoids) fue desarrollado por Kaufman y Rousseeuw [19]. Encontrar k clusters, el enfoque de PAM es determinar un objeto representativo para cada clúster. Este objeto representativo, llamado medoide está destinado a ser el objeto ubicado más centralmente dentro del clúster. Una vez que se han seleccionado los medoides, cada objeto no seleccionado se agrupa con el medoide al que pertenece más

similar. Más precisamente, si O_j es un objeto no seleccionado y O_{metro} es un medoide (seleccionado), decimos que O_j pertenece al cluster representado por O_{metro} Si re re O_j; O_{metro} Þ ¼ min $_O$ re re O_j; O_{mi} Þ, donde la notación min $_O$ denota el mínimo sobre todos los medoides O_{mi}
y la notación re re O_1; O_2 Þ denota la disimilitud o distancia entre objetos O_1 y O_2. Todos los valores de disimilitud se dan como entradas a PAM. Finalmente, el calidad de un agrupamiento es decir, la calidad combinada de los medoides elegidos) se mide por la disimilitud promedio entre un objeto y el medoide de su grupo. Para encontrar el k medoides, PAM comienza con una selección arbitraria de k objetos. Entonces, en cada paso, un intercambio entre un objeto seleccionado O_{metro} y un no seleccionado objeto O_{pags} se realiza, siempre que dicho intercambio redunde en una mejora de la calidad de la agrupación.

Antes de embarcarnos en un análisis formal, consideremos un ejemplo simple. Supongamos que hay 2 medoides: UN y SEGUNDO.
Y consideramos reemplazar UN con un nuevo medoide METRO. Entonces, para todos los objetos Y que están originalmente en el clúster representado por UN, tenemos que encontrar el medoide más cercano a la luz del reemplazo. Hay dos casos. En el primer caso, Y se mueve al grupo representado por SEGUNDO, pero no al nuevo representado por METRO. En el segundo caso, Y se mueve al nuevo clúster representado por METRO, y el cluster representado por segundo No es afectado. Aparte de reconsiderar todos los objetos Y que están originalmente en UN clúster, también debemos considerar todos los objetos Z que están originalmente en segundo grupo de. A la luz del reemplazo, Z o se queda con SEGUNDO, o se mueve al nuevo clúster representado por METRO. La figura 1 ilustra los cuatro casos.

En el resto de este documento, usamos:

- O_{metro} para denotar un medoide actual que debe ser reemplazado (p. ej., UN en la figura 1),
- O_{pags} para denotar el nuevo medoide para reemplazar O_{metro}(p.ej, METRO en la figura 1),
- O_j para denotar otros objetos no medoides que pueden necesitar o no ser movidos (por ejemplo, Y y Z en la Fig.1), y

. $O_{j;2}$ para denotar un medoide actual más cercano a O_j sin UN y M (p.ej, segundo en la figura 1).

Ahora, para formalizar el efecto de un intercambio entre O_{metro} y O_{pags}, PAM calcula los costos C_{jmp} para todos los objetos no medoides O_j. Según cuál de los siguientes casos O_j es en, C_{jmp} se define de manera diferente.

Caso 1. suponer O_j actualmente pertenece al clúster representado por O_{metro}. Además, deja O_j ser más similar a $O_{j;2}$ que a O_{pags}, es decir, re re $O_{j;}$ O_{pags} Þ re re $O_{j;}$ $O_{j;2}$ Þ, dónde $O_{j;2}$ es el segundo medoide más similar a O_j. Por lo tanto, si O_{metro} es reemplazado por O_{pags} como medoide O_j pertenecería al clúster representado por $O_{j;2}$ (cf. Caso 1 en la Fig.1). Por lo tanto, el costo del swap hasta O_j está preocupado es:

$$C_{jmp} ¼ \text{re re } O_{j;} O_{j;2} Þ \text{ re re } O_{j;} O_{metro} Þ: \qquad \text{re } 1 Þ$$

Esta ecuación siempre da un valor no negativo C_{jmp}, indicando que existe un costo no negativo incurrido en el reemplazo O_{metro} con O_{pags}.

Caso 2. O_j actualmente pertenece al clúster representado por O_{metro}. Pero esta vez, O_j es menos similar a $O_{j;2}$ que a O_{pags}, es decir, re re $O_{j;}$ O_{pags} Þ < re re $O_{j;}$ $O_{j;2}$ Þ. Entonces sí O_{metro} es reemplazado por O_{pags}, O_j pertenecería al clúster representado por O_{pags} (cf. Figura 1). Por lo tanto, el costo de O_j es dado por:

$$C_{jmp} ¼ \text{re re } O_{j;} O_{pags} Þ \text{ re re } O_{j;} O_{metro} Þ: \qquad \text{re } 2 Þ$$

A diferencia de (1), C_{jmp} aquí puede ser positivo o negativo, dependiendo de si O_j es más similar a O_{metro} o para O_{pags}.

Caso 3. suponer que O_j actualmente pertenece a un clúster diferente al representado por O_{metro}. Dejar $O_{j;2}$ ser el objeto representativo de ese cluster. Además, deja O_j ser más similar a $O_{j;2}$ que a O_{pags}. Entonces, incluso si O_{metro} es remplazado por O_{pags}, O_j permanecería en el grupo representado por $O_{j;2}$. Por tanto, el costo es:

$$C_{jmp} ¼ 0: \qquad \text{re } 3 Þ$$

Caso 4. O_j actualmente pertenece al clúster representado por $O_{j;2}$. Pero, O_j es menos similar a $O_{j;2}$ que a O_{pags}. Luego, reemplazando O_{metro} con O_{pags} causaría O_j para saltar al grupo de O_{pags} de la de $O_{j;2}$ Por tanto, el costo es:

$$C_{jmp} ¼ \text{re re } O_{j;} O_{pags} Þ \text{ re re } O_{j;} O_{j;2} Þ; \qquad \text{re } 4 Þ$$

y siempre es negativo. Combinando los cuatro casos anteriores, las muestras, cada iteración es de O re k re 40 þ k Þ $_2$ þ k re nk ÞÞ. costo total de reemplazo O_{metro} con O_{pags} es dado por:

$$TC_{mp} ¼ \sum_j C_{jmp}: \qquad \text{re } 5 Þ$$

Presentamos ahora el algoritmo PAM.

Algoritmo PAM

1. Seleccione k objetos representativos arbitrariamente.
2. Calcular TC_{mp} para todas pares de objetos $O_{metro;}$ O_{pags} donde O_{metro} está seleccionado actualmente, y O_{pags} no es.
3. Seleccione el par $O_{metro;}$ O_{pags} que corresponde a $\min_{O_{metro;}\, O_{pags}} TC_{mp}$. Si el mínimo TC_{mp} es negativo, luego, reemplazar O_{metro} con O_{pags}, y vuelva al paso 2.
4. De lo contrario, para cada objeto no seleccionado, encuentre el objeto representativo similar. Detener.

Los resultados experimentales muestran que PAM funciona satisfactoriamente para conjuntos de datos pequeños (por ejemplo, 100 objetos en 5 grupos [19]). Pero no es eficiente al tratar con conjuntos de datos medianos y grandes. Esto no es demasiado sorprendente si realizamos un análisis de complejidad en PAM. En los pasos 2 y 3, hay k re nk Þ pares de $O_{metro;}$ O_{pags}. Para cada par, computando TC_{mp} requiere el examen de re nk Þ objetos no seleccionados. Por lo tanto, los Pasos 2 y 3 combinados son de O re k re nk Þ $_2$ Þ. Y esta es la complejidad de una sola iteración. Por lo tanto, es obvio que PAM se vuelve demasiado costoso para grandes valores de norte y k. Este análisis motiva el desarrollo de CLARA.

2.3 CLARA

Diseñado por Kaufman y Rousseeuw para manejar grandes conjuntos de datos, CLARA (Clustering LARge Applications) se basa en el muestreo [19]. En lugar de encontrar objetos representativos para todo el conjunto de datos, CLARA extrae una muestra del conjunto de datos, aplica PAM en la muestra y encuentra los medoides de la muestra. El punto es que, si la muestra se extrae de una manera suficientemente aleatoria, los medoides de la muestra se aproximarían a los medoides de todo el conjunto de datos. Para obtener mejores aproximaciones, CLARA extrae múltiples muestras y da el mejor agrupamiento como resultado. Aquí, para mayor precisión, la calidad de un agrupamiento se mide en función de la disimilitud promedio de todos los objetos en el todo conjunto de datos, y no solo de esos objetos en las muestras. Los experimentos informados en [19] indican que cinco muestras de tamaño 40 þ 2 k dar resultados satisfactorios.

Algoritmo CLARA

1. por yo ¼ 1 a 5, repita los siguientes pasos: Extraiga una muestra de 40 þ 2 k objetos
2. aleatoriamente de todo el conjunto de datos, $_2$ y llame al algoritmo PAM para encontrar k medoides de la muestra.
3. Para cada objeto O_j en todo el conjunto de datos, determine de los cuales k medoides es el más parecido a O_j.
4. Calcular la disimilitud promedio de la agrupación obtenido en el paso anterior. Si este valor es menor que el mínimo actual, use este valor como el mínimo actual y conserve el k medoides encontrados en el Paso 2 como el mejor conjunto de medoides obtenido hasta ahora. Regrese al paso 1 para comenzar la siguiente iteración.
5.

Como complemento de PAM, CLARA se desempeña satisfactoriamente para grandes conjuntos de datos (por ejemplo, 1,000 objetos en 10 grupos). Recuerde de la Sección 2.2 que cada iteración de PAM es de O re k re nk Þ $_2$ Þ. Pero, para CLARA, al aplicar PAM solo para Esto explica por qué CLARA es más eficiente que PAM para grandes valores de norte.

3 AC LUSTERING UN LGORITMO segundo ASED EN R ANDOMIZADO S EARCH

En esta sección, presentaremos nuestro algoritmo de agrupación - CLARANS (Agrupación de grandes aplicaciones basadas en RAN- marco dentro del cual podemos comparar PAM y Presentar resultados experimentales que muestran cómo afinar

2. Kaufman y Rousseeuw [19] informan de una heurística útil para dibujar muestras. Aparte de la primera muestra, las muestras posteriores incluyen el mejor conjunto de medoides encontrado hasta ahora. En otras palabras, aparte de la primera iteración, las iteraciones posteriores dibujan 40 þ k objetos para agregar a los mejores k medoides.

CLARANS y que CLARANS supera a CLARA y PAM en términos de eficiencia y eficacia.

3.1 Motivación de CLARANS: una abstracción gráfica

Dado norte objetos, el proceso descrito anteriormente para encontrar k Los medoides pueden verse abstractamente como una búsqueda a través de cierto gráfico. En este gráfico, denotado por GRAMO n; k, un nodo está representado por un conjunto de k objetos F O m; . . . ; O metro gramo, intui indicando claramente que O m; . . . ; O metro son los seleccionados medoides. El conjunto de nodos en el gráfico es el conjunto ff O m; . . . ; O metro gj Q; . . ; O metro . son objetos en el conjunto de datos}.

Dos nodos son vecinos (es decir, conectados por un arco) si sus conjuntos difieren sólo en un objeto. Más formalmente, dos nodos S 1 ¼ f O m; . . . ; O metro gramo y S 2 ¼ f O w; . . ; O w gramo son vecinos si y solo si la cardinalidad de la intersección de S 1; S 2 es k 1, es decir, j S 1 \ S 2 j ¼ k 1. Es fácil ver que cada nodo tiene k re nk Þ vecinos. Dado que un nodo representa un colección de k medoides, cada nodo corresponde a un agrupamiento. Por lo tanto, a cada nodo se le puede asignar un costo que se define como la disimilitud total entre cada objeto y el medoide de su grupo. No es difcil ver que, si objetos O metro; O pags son las diferencias entre vecinos S 1 y S 2 (es decir, O metro; O pags 62 S 1 \ S 2, pero O metro 2 S 1 y O pags 2 S 2), la diferencia de costo entre los dos vecinos está dada exactamente por T mp definido en (5).

Por ahora, es obvio que PAM puede verse como una búsqueda por un mínimo en el gráfico GRAMO n; k. En cada paso, se examinan todos los vecinos del nodo actual. La corriente Luego, el nodo es reemplazado por el vecino con el descenso más profundo en los costos. Y la búsqueda continúa hasta obtener un mínimo. Para grandes valores de norte y k (me gusta norte ¼ 1; 000 y k ¼ 10), examinando todo k re nk Þ vecinos de un nodo lleva mucho tiempo. Esto explica la ineficacia de PAM para grandes conjuntos de datos.

Por otro lado, CLARA intenta examinar menos vecinos y restringe la búsqueda en subgrafos que son mucho más pequeño que el gráfico original GRAMO n; k. Sin embargo, el problema es que los subgrafos examinados se definen enteramente por los objetos en las muestras. Dejar Sa ser el conjunto de objetos en una muestra. El subgrafo GRAMO Sa; k consta de todos los nodos que son subconjuntos (de cardinalidades k) de Sa. Aunque CLARA examina a fondo GRAMO Sa; k vía PAM, el problema es que la búsqueda está completamente confinada en GRAMO Sa; k. Si METRO es el nodo mínimo en el gráfico original GRAMO n; k y si METRO no está incluido en GRAMO Sa; k, METRO nunca se encontrará en la búsqueda de GRAMO Sa; k, independientemente de cuán completa sea la búsqueda. Para reparar esta deficiencia, sería necesario recolectar muchas muestras y procesado.

Como CLARA, nuestro algoritmo CLARANS no verifica a todos los vecinos de un nodo. Pero, a diferencia de CLARA, no restringe su búsqueda a un subgrafo en particular. De hecho, busca en el gráfico original GRAMO n; k. Una diferencia clave entre CLARANS y PAM es que el primero solo comprueba una muestra de los vecinos de un nodo. Pero, a diferencia de CLARA, cada muestra se extrae dinámicamente en el sentido de que ningún nodo correspondiente a objetos particulares se elimina por completo. En otras palabras, mientras CLARA extrae una muestra de nodos al comienzo de una búsqueda, CLARANS extrae una muestra de vecinos en cada paso de una búsqueda. Esto tiene la ventaja de no limitar la búsqueda a un área localizada. Como se mostrará en la Sección 3.3, una búsqueda por CLARANS da agrupaciones de mayor calidad que CLARA, y CLARANS requiere una cantidad muy pequeña de búsquedas. Presentamos ahora los detalles del algoritmo CLARANS.

3.2 CLARANOS

Algoritmo CLARANOS

1. Parámetros de entrada numlocal y maxneighbor. Inicializar yo a 1, y mincost a un gran número.
2. Conjunto Actual a un nodo arbitrario en GRAMO n; k.
3. Conjunto j a 1.
4. Considere un vecino al azar S de Actual, y basado en 5, calcule el diferencial de costos de los dos nodos. Si S tiene un costo menor,
5. establece Actual a S, y vaya al paso 3.
6. De lo contrario, incremente j por 1. Si j maxneighbor, Vamos al paso 4.
7. De lo contrario, cuando j> vecino máximo, comparar el costo de Actual con mincost. Si el primero es menor que mincost, conjunto mincost al costo de Actual y establecer bestnode a Actual.
8. Incremento yo por 1. Si i> numlocal, salida bestnode y detente. De lo contrario, vaya al paso 2.

Los pasos 3 a 6 anteriores buscan nodos con costos cada vez más bajos. Pero, si el nodo actual ya se ha comparado con el número máximo de vecinos del nodo (especificado por maxneighbor) y aún tiene el costo más bajo, el nodo actual se declara como mínimo “local”. Luego, en el Paso 7, el costo de este mínimo local se compara con el costo más bajo obtenido hasta ahora. El menor de los dos costos anteriores se almacena en mincost. El algoritmo CLARANS luego se repite para

buscar otros mínimos locales, hasta numlocal de ellos se han encontrado.

Como se muestra arriba, CLARANS tiene dos parámetros: el número máximo de vecinos examinados (maxneighbor) y el número de mínimos locales obtenidos (numlocal). Cuanto mayor sea el valor de maxneighbor, cuanto más cerca está CLARANS de PAM, y más larga es cada búsqueda de un mínimo local. Sin embargo, la calidad de estos mínimos locales es mayor y es necesario obtener menos mínimos locales. Como muchas aplicaciones de búsqueda aleatoria [16], [17], nos basamos en experimentos para determinar los valores apropiados de estos parámetros.

3.3 Resultados experimentales: Tuning CLARANS

3.3.1 Detalles de los experimentos

Para observar el comportamiento y la eficiencia de CLARANS, ejecutamos CLARANS con conjuntos de datos generados cuyos clusters son conocidos. Para una mejor generalización, utilizamos dos tipos de grupos con características bastante opuestas. El primer tipo de grupos es rectangular y los objetos dentro de cada grupo se generan aleatoriamente. Más específicamente, si se necesita un conjunto de datos de, digamos, 3000 objetos en 20 grupos, primero generamos 20 "cuadros delimitadores" del mismo tamaño. Para que los conglomerados sean menos nítidos, la esquina noreste del yo a casilla y la esquina suroeste de re yo þ 1 Þ th toque de caja. Dado que para nuestra aplicación de minería de datos espaciales, CLARANS se utiliza para agrupar coordenadas espaciales, los objetos en nuestros experimentos aquí son pares de coordenadas x, y. Para cada cuadro delimitador, generamos aleatoriamente 150 pares de coordenadas que se encuentran dentro del cuadro. De manera similar, generamos conjuntos de datos del mismo tipo pero con un número variable de objetos y grupos. En la figura de abajo, el símbolo r nk p.ej,

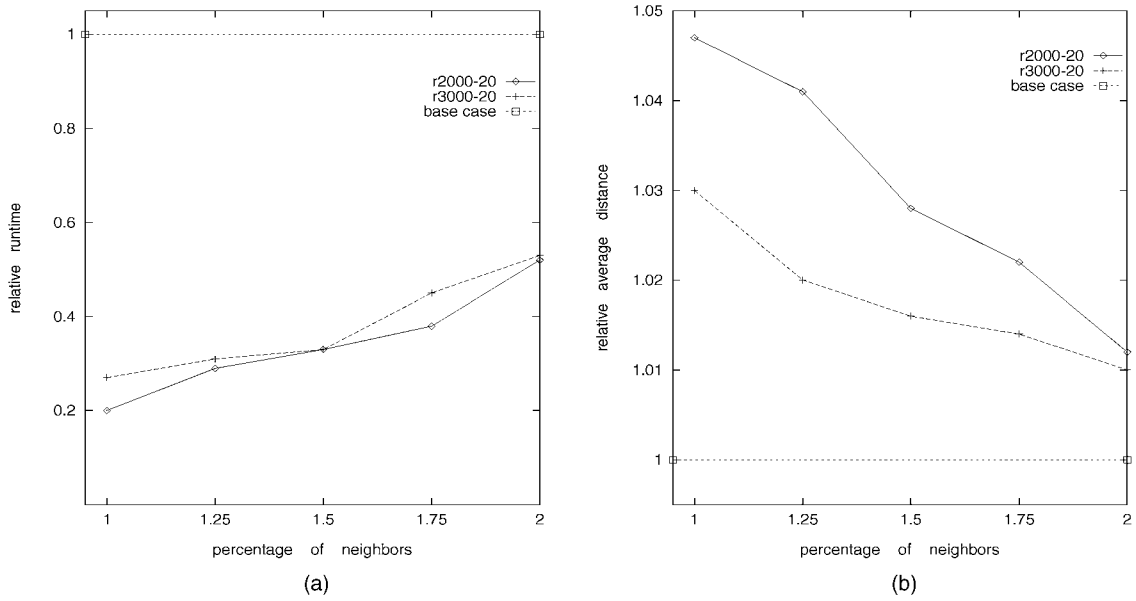

Fig. 2. Determinación del número máximo de vecinos. (a) Eficiencia relativa. (b) Calidad relativa.

r3000-20) representa un conjunto de datos de este tipo con norte puntos en k racimos.

A diferencia del primer tipo, el segundo tipo de grupos con los que experimentamos no contiene puntos aleatorios. Más bien, los puntos dentro de un grupo se ordenan en un triángulo. Por ejemplo, los puntos con coordenadas (0,0), (1,0), (0,1), (2,0), (1,1) y (0,2) forman un grupo triangular de tamaño 6. Para producir un clúster junto al anterior, usamos una traducción del origen (p. Ej., Los puntos (10,10), (11,10), (10,11), (12,10), (11 , 11) y (10,12)). En las figuras siguientes, el símbolo t nk por ejemplo, t3000-20) representa un conjunto de datos organizado de esta manera con norte puntos en k racimos.

Todos los experimentos informados aquí se llevaron a cabo en una estación de trabajo SPARC-LX de tiempo compartido. Debido a la naturaleza aleatoria de CLARANS, todas las cifras relativas a CLARANS son cifras promedio obtenidas al ejecutar el mismo experimento 10 veces (con diferentes semillas del generador de números aleatorios).

3.3.2 Determinación del número máximo de vecinos

En la primera serie de experimentos, aplicamos CLARANS con el parámetro maxneighbor ¼ 250, 500, 750, 1,000 y 10,000 en los conjuntos de datos r nk y t nk, dónde norte varía de 100 a 3000 y k varía de 5 a 20. Para ahorrar espacio, solo resumimos los dos hallazgos principales que conducen a más experimentos:

. Cuando el número máximo de vecinos maxneighbor se establece en 10,000, la calidad de la agrupación producida por CLARANS es efectivamente la misma que la calidad de la agrupación producida por PAM (es decir, maxneighbor ¼ k re nk Þ). Si bien explicaremos este fenómeno en breve, usamos los resultados para maxneighbor ¼ 10; 000 como criterio para evaluar otros valores (menores) de maxneighbor. Más específicamente, los valores de tiempo de ejecución del primer gráfico y los valores de distancia promedio (es decir, la calidad de un agrupamiento) del segundo gráfico en la Fig.2 a continuación son normas configurando maxneighbor ¼ Al 2,000 Estos porcentajes dos líneas horizontales en y valor ¼ 1 en ambos gráficos.

. Como era de esperar, un valor menor de maxneighbor produce una agrupación de menor calidad. Una pregunta que nos hacemos es entonces cuán pequeño puede ser el valor de maxneighbor antes de que la calidad de la agrupación se vuelva inaceptable. De la primera serie de experimentos, encontramos que estos valores críticos parecen ser proporcionales al valor k re nk Þ. Esto nos motiva a realizar otra serie de experimentos con la siguiente fórmula mejorada para determinar el valor de maxneighbor,

dónde minmaxneighbor es un mínimo definido por el usuario valor por maxneighhbor:

- Si k re nk Þ minmaxneighbor luego maxneighbor ¼ k re nk Þ; de otra manera, maxneighbor es igual al valor mayor entre pags% de k re nk Þ y minmaxneighbor.

La fórmula anterior permite a CLARANS examinar a todos los vecinos siempre que el número total de vecinos esté por debajo del umbral. minmaxneighbor. Más allá del umbral, el porcentaje de vecinos examinados cae gradualmente del 100 por ciento a un mínimo de pags%. Los dos gráficos de la Fig.2 muestran el tiempo de ejecución relativo y la calidad de CLARANS con

minmaxneighbor ¼ 250 y pags variando del 1 al 2 por ciento.

Si bien los gráficos solo muestran los resultados de los conjuntos de datos rectangulares con 2000 y 3000 puntos en 20 grupos, estos gráficos son representativos, ya que las apariencias de los gráficos para conjuntos de datos pequeños y medianos y para los conjuntos de datos triangulares son muy similares.

La figura 2a muestra que cuanto menor es el valor de pags, menor es la cantidad de tiempo de ejecución que requiere CLARANS. Y como era de esperar, la Fig.2b muestra que un valor más bajo de pags produce un agrupamiento de menor calidad (es decir, una distancia promedio más alta (relativa)). Pero, la característica muy sorprendente que se muestra en la Fig.2b es que la calidad todavía está dentro del 5 por ciento de la producida por

TABLA 1
Tiempo de ejecución relativo y calidad para el conjunto de datos r2000-20

numlocal	1	2	3	4	5
relative runtime	0.19	0.38	0.6	0.78	1
relative average distance	1.029	1.009	1	1	1

ajuste maxneighbor ¼ 10; 000 (o por PAM). Como ejemplo, si un máximo de pags ¼ 1: 5% de los vecinos se examinan, la calidad está dentro del 3 por ciento, mientras que el tiempo de ejecución es solo del 40 por ciento. Lo que eso significa es que examinar un 98,5 por ciento más de vecinos, aunque lleva mucho más tiempo, solo produce resultados marginalmente mejores. Esto es consistente con nuestra declaración anterior de que CLARANOS con maxneigh ¼ 10; 000 da la misma calidad que PAM, que es efectivamente la misma que

ajuste maxneighbor ¼ k re nk Þ ¼ 20 re 3000 20 Þ ¼ 59; 600.

La razón por la que es necesario examinar tan pocos vecinos para obtener agrupaciones de buena calidad se puede ilustrar mejor con la abstracción de gráficos presentada en la Sección 3.1. Recuerde que cada nodo tiene k re nk Þ vecinos, lo que hace que el gráfico sea muy
conectado. Considere dos vecinos S_1; S_2 del nodo actual, y suponga que S_1 constituye un camino que conduce a un cierto nodo mínimo S. Incluso si S_1 se pasa por alto al no ser examinado y S_2 se convierte en el nodo actual, todavía quedan

numerosos caminos que conectan S_2 a S. Por supuesto, si todas estas rutas no son estrictamente descendentes (en costo) y pueden
incluir "colinas" en el camino, S nunca será alcanzado desde
S_2. Pero nuestros experimentos parecen indicar que la posibilidad de que exista una colina en cada El camino es muy pequeño.

Para mantener un buen equilibrio entre tiempo de ejecución y calidad, creemos que pags valor entre 1,25 por ciento y 1,5 por ciento es muy razonable. Para todos nuestros experimentos posteriores con CLARANS, elegimos el valor pags ¼ 1:25%.

3.3.3 Determinación del número de mínimos locales

Recuerde que el algoritmo CLARANS tiene dos parámetros: maxneighbor y numlocal. Habiendo tratado el primero, aquí nos enfocamos en determinar el valor de numlocal. En esta serie de experimentos, ejecutamos CLARANS con numlocal ¼
1; . . . ; 5 en conjuntos de datos r nk y t nk para valores pequeños, medianos y grandes de norte y k. Para cada ejecución, registramos el tiempo de ejecución y la calidad de la agrupación. La Tabla 1 (que es típica de todos los conjuntos de datos) muestra el tiempo de ejecución relativo y la calidad del conjunto de datos r2000-20. Aquí, todos los valores están normalizados por aquellos con numlocal ¼ 5.

Como era de esperar, los tiempos de ejecución son proporcionales al número de mínimos locales obtenidos. En cuanto a la calidad relativa, hay una mejora de numlocal ¼ 1 a
numlocal ¼ 2. La realización de una segunda búsqueda de un mínimo local parece reducir el impacto de la aleatoriedad "desafortunada" que puede ocurrir en una sola búsqueda. Sin embargo, el establecimiento numlocal mayor que 2 no es rentable, ya que hay poco aumento en la calidad. Esta es una indicación de que un mínimo local típico es de muy alta calidad. Creemos que este fenómeno se debe en gran parte, como se discutió anteriormente, a la naturaleza peculiar del gráfico abstracto que representa las operaciones de CLARANS. Para todos nuestros experimentos posteriores con CLARANS, usamos la versión que encuentra dos mínimos locales.

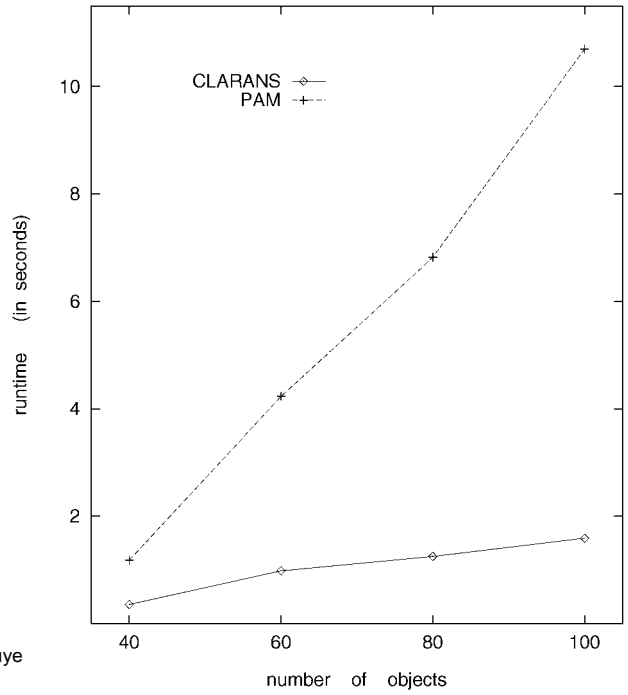

Fig. 3. Eficiencia: CLARANS versus PAM.

3.4 Resultados experimentales: CLARANS versus PAM

En esta serie de experimentos, comparamos CLARANS con PAM. Como se discutió en la Sección 3.3.2, para conjuntos de datos grandes y medianos, es obvio que CLARANS, si bien produce agrupaciones de calidad muy comparable, es mucho más eficiente que PAM. Por lo tanto, nuestro enfoque aquí fue comparar los dos algoritmos en pequeños conjuntos de datos. Aplicamos ambos algoritmos a conjuntos de datos con 40, 60, 80 y 100 puntos en cinco grupos. La Fig. 3 muestra el tiempo de ejecución de ambos algoritmos. Tenga en cuenta que, para todos esos conjuntos de datos, las agrupaciones producidas por ambos algoritmos son de la misma calidad (es decir, la misma distancia promedio). Por tanto, la diferencia entre los dos algoritmos está determinada por su eficacia. Es evidente en la Fig. 3 que, incluso para conjuntos de datos pequeños, CLARANS supera significativamente a PAM. Como era de esperar, la brecha de rendimiento entre los dos algoritmos crece,

3.5 Resultados experimentales: CLARANS versus CLARA

En esta serie de experimentos, comparamos CLARANS con CLARA. Como se discutió en la Sección 2.3, CLARA no está diseñado para pequeños conjuntos de datos. Por lo tanto, realizamos este conjunto de experimentos en conjuntos de datos cuyo número de objetos excede
100. Y los objetos se organizaron en diferentes números de grupos, así como en los dos tipos de grupos descritos en la Sección 3.3.1.

Cuando realizamos esta serie de experimentos ejecutando CLARA y CLARANS como se presentó anteriormente, CLARANS siempre puede encontrar agrupaciones de mejor calidad que las encontradas por CLARA. Sin embargo, en algunos casos, CLARA puede llevar mucho menos tiempo que CLARANS. Por lo tanto, nos preguntamos si CLARA produciría agrupaciones de la misma calidad si se le diera la misma cantidad de tiempo. Esto nos lleva a la siguiente serie de experimentos en los que les dimos a CLARANS y CLARA la misma cantidad de tiempo. Figura 4

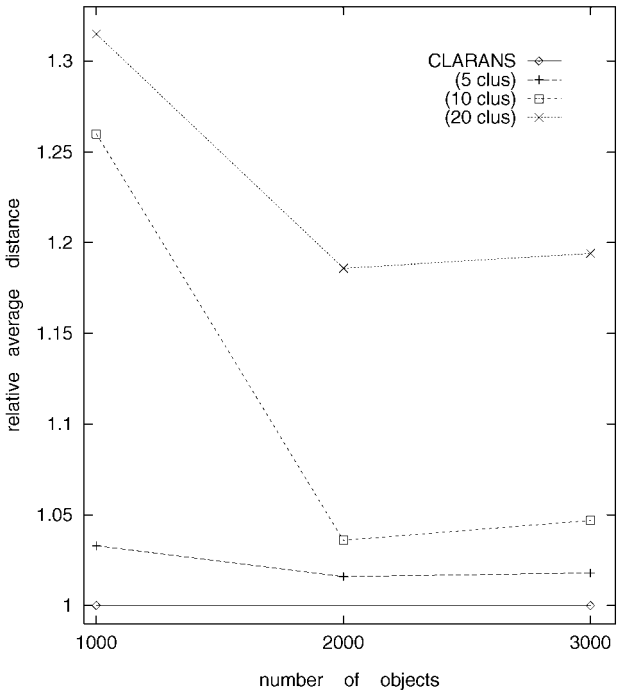

Fig. 4. Calidad relativa: mismo tiempo para CLARANS y CLARA.

muestra la calidad de los agrupamientos producidos por CLARA, normalizada por el valor correspondiente producido por CLARANS.

Con la misma cantidad de tiempo, CLARANS claramente supera a CLARA en todos los casos. La brecha entre CLARANS y CLARA aumenta del 4 por ciento cuando k,
el número de grupos, es del cinco al 20 por ciento cuando k es 20. Esta ampliación de la brecha como k Los aumentos se pueden explicar mejor al observar los análisis de complejidad de CLARA y CLARANS. Recuerde de la Sección 2.3 que cada iteración de CLARA es de O re k 3 þ nk Þ. Por otro lado, recuerde de la Sección 3.3.2 que el costo de CLARANS es básicamente linealmente proporcional al número de objetos. 3 Por lo tanto, un aumento en

k impone un costo mucho mayor a CLARA que a CLARANS.

La comparación de complejidad anterior también explica por qué, para un número fijo de clusters, cuanto mayor es el número de objetos, más estrecha es la brecha entre CLARANS y CLARA. Por ejemplo, cuando el número de objetos es
1,000, la brecha es tan alta como 30 por ciento. La brecha se reduce a alrededor del 20 por ciento a medida que aumenta el número de objetos a
2.000. Dado que cada iteración de CLARA es de O re k 3 þ nk Þ, el primer trimestre k 3 domina el segundo término. Por lo tanto, para un fijo k,
CLARA es relativamente menos sensible a un aumento en norte. Por otro lado, dado que el costo de CLARANS es aproximadamente linealmente proporcional a norte, un aumento en norte impone un costo mayor a CLARANS que a CLARA. Esto explica por qué, para un fijo k, la brecha se estrecha a medida que el número de objetos aumenta. No obstante, la conclusión que se muestra en la Fig. 4 es que CLARANS vence a CLARA en todos los casos.

En resumen, hemos presentado evidencia experimental que muestra que CLARANS es más eficiente que PAM y CLARA para conjuntos de datos pequeños y grandes. Nuestros resultados experimentales para conjuntos de datos medianos (no incluidos aquí) conducen a la misma conclusión.

4 Clustering Convex Pagsólogon Objects

4.1 Motivación

Como se describe en la Sección 3.3, todos los experimentos presentados hasta ahora asumen que cada objeto se representa como un punto, en cuyo caso se pueden usar métricas de distancia estándar como la distancia de Manhattan y la distancia euclidiana para calcular la distancia entre dos objetos / puntos. Sin embargo, en la práctica, numerosos objetos espaciales que podríamos querer agrupar son de naturaleza poligonal, por ejemplo, centros comerciales, parques. La pregunta central entonces es cómo calcular la distancia entre dos objetos poligonales de manera eficiente y efectiva para fines de agrupación. Una forma obvia de aproximar objetos poligonales es representar cada objeto por un punto representativo, como el centroide del objeto. Sin embargo, en general, los objetos que se agrupan pueden tener tamaños y formas muy variados. Por ejemplo, una casa típica en Vancouver puede tener un tamaño de lote de 200 metros cuadrados y una forma rectangular, mientras que Stanley Park en Vancouver tiene un tamaño de unos 500.000 metros cuadrados y una forma irregular que abraza la costa. Simplemente representando cada uno de estos objetos por su centroide, o cualquier punto único,

fácilmente produciría agrupaciones de mala calidad.

Dado el argumento anterior, uno puede preguntarse si es suficiente representar un objeto poligonal por múltiples puntos en el objeto, por ejemplo, puntos en el límite del objeto. Pero, para objetos grandes como Stanley Park, dos de sus puntos representativos pueden estar a 5.000 metros de distancia entre sí. Si estos dos puntos representativos se envían a CLARANS como individual

puntos / objetos, no hay garantía de que estén en el mismo grupo. Esto daría lugar a que Stanley Park se asignara a más de un clúster, lo que violaría el requisito de partición de los algoritmos de clúster. 4

Esto motiva por qué, en esta sección, estudiamos cómo CLARANS (y para el caso, CLARA y PAM) pueden aumentarse para permitir que los objetos poligonales convexos, en su totalidad, se agrupen. La pregunta clave es cómo calcular de manera eficiente la distancia entre dos polígonos. Para responder a esta pregunta, estudiamos tres enfoques diferentes. El primero se basa en calcular la distancia de separación exacta entre dos objetos poligonales convexos. El segundo enfoque utiliza la distancia mínima entre vértices para aproximar la distancia de separación exacta. El tercer enfoque se aproxima utilizando la distancia de separación entre rectángulos isotéticos. Analizamos los pros, los contras y las complejidades de estos enfoques. Por último, pero no menos importante, proponemos una optimización del rendimiento que se basa en memorizar las distancias calculadas. Al final de esta sección,

3. Hay un aspecto aleatorio y un aspecto no aleatorio en la ejecución de CLARANOS. El aspecto no aleatorio corresponde a la parte que encuentra el diferencial de costo entre el nodo actual y su vecino. Esta parte, como se define en (5), es linealmente proporcional al número de objetos en el conjunto de datos. Por otro lado, el aspecto aleatorio corresponde a la parte que busca un mínimo local. Como los valores para trazar los gráficos son valores promedio de 10 corridas, que tienen el efecto de reducir la influencia del aspecto aleatorio, los tiempos de ejecución de CLARANS utilizados en nuestros gráficos están dominados en gran medida por el aspecto no aleatorio de CLARANS.

4. Tenga en cuenta que existen algoritmos de agrupación en clústeres que permiten que los clústeres superposición. Sin embargo, aquí solo nos ocupamos del tipo más estándar de agrupaciones, generalmente conocidas como crujiente agrupamiento, donde cada objeto se asigna a exactamente un grupo.

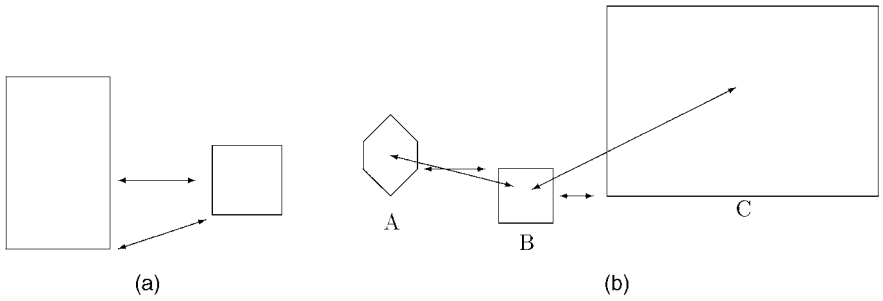

Fig. 5. Aproximación de MV frente a distancia de separación frente a distancia del centroide. (a) Aproximación de MV versus distancia de separación y (b) distancia de centroide versus distancia de separación.

4.2 Cálculo de la distancia de separación exacta

En geometría de coordenadas, la distancia entre un punto PAGS y una linea L se define como la distancia mínima, en este caso la distancia perpendicular, entre el punto y la línea, es decir,

min F re re PAGS; Q Þ j Q es un punto en L gramo. Así, dados dos polígonos UN; SEGUNDO, es natural para nosotros definir la distancia entre estos dos polígonos como la distancia mínima entre cualquier par

de puntos en UN; SEGUNDO, es decir, min F re re PAGS; Q Þ j PAGS; Q son puntos en UN; segundo respectivamente gramo. Esta distancia es exactamente la misma que la distancia mínima entre cualquier par de puntos en los límites de UN; SEGUNDO. Esto se llama separación distancia entre los dos polígonos [9], [27].

Necesitamos dos pasos clave para calcular la distancia de separación entre dos polígonos convexos UN; SEGUNDO. Primero, necesitamos determinar si UN y segundo tener alguna intersección. Esto se puede hacer en O re log n þ log m Þ tiempo donde Nuevo Méjico denota el número de vértices UN y segundo tener [27]. (Un vértice de un polígono es el punto de intersección entre dos bordes del polígono). Si los dos polígonos se intersecan, entonces la distancia de separación es cero. De lo contrario, calculamos la distancia de separación entre los dos límites. Esto nuevamente se puede hacer en O re log n þ log m Þ tiempo [9], [21].

Si bien existen diferentes algoritmos que calculan la distancia de separación y que tienen la misma O re log n þ log m Þ complejidad, hemos elegido e implementado uno de los algoritmos más eficientes que se informa en [21]. En lugar de probar todos los vértices y aristas en los límites de los dos polígonos, este algoritmo primero identifica dos cadenas de vértices y segmentos de línea consecutivos (una cadena de cada polígono) que se "enfrentan" entre sí en el sentido de que un vértice en cualquier cadena puede "Ver" al menos un vértice en la otra cadena. Si PAGS es un vértice en UN y Q un vértice en SEGUNDO, Nosotros decimos eso PAGS y Q "Verse" entre sí si el segmento de línea que se une PAGS

y Q no se cruza con el interior de ninguno de los polígonos. [5]
Así, por definición, la distancia de separación de los dos polígonos debe ser la distancia mínima entre cualquier par de puntos (no necesariamente vértices) en las dos cadenas. Al tomar una estrategia de búsqueda binaria, el algoritmo encuentra la última distancia.

4.3 Aproximación por la distancia mínima entre vértices

Uno forma de aproximar la distancia de separación exacta entre dos polígonos es encontrar la distancia mínima entre los vértices de los polígonos, es decir, min F re re PAGS; Q Þ j PAGS; son vértices de UN; segundo respectivamente gramo. En adelante, nos referiremos a este aproximación como el Aproximación MV. Obviamente, la aproximación MV requiere una complejidad de tiempo de O re Nuevo Méjico Þ.
Aunque desde el punto de vista de la complejidad, esta aproximación es inferior al algoritmo descrito anteriormente que calcula la distancia de separación exacta en la práctica, generalmente supera al algoritmo exacto, a menos que los valores de norte y metro son moderadamente altos, por ejemplo, superiores a 20. Para muchas aplicaciones de minería de datos espaciales, es suficiente representar la mayoría de los objetos espaciales con menos de 20 vértices y aristas. Esto justifica el uso de la aproximación MV para optimizar el rendimiento. La sección 4.6.2 proporcionará resultados experimentales sobre la eficiencia de la aproximación de MV.

La Fig. 5a muestra un ejemplo simple que demuestra que la distancia de separación entre dos polígonos no necesita ser igual a la distancia mínima entre vértices. Sin embargo, es fácil ver que la distancia de separación no puede exceder la distancia mínima entre vértices. Por tanto, la aproximación MV siempre sobreestima la distancia de separación real. Desde el punto de vista de agrupar objetos por sus aproximaciones MV, la pregunta clave es si tales sobreestimaciones afectarían la (calidad de) las agrupaciones.

Recuerde de la Sección 4.1 que argumentamos que usar solo el centroide para representar un objeto podría producir agrupaciones de mala calidad. Y la distancia entre los centroides de UN y segundo siempre sobreestima la distancia de separación real. En este punto, uno puede preguntarse si la sobreestimación por la distancia del centroide y la aproximación de MV tienen efectos similares. La diferencia clave es que el primero, dependiendo en gran medida de los tamaños y formas de los polígonos, da aproximaciones "no uniformes", mientras que el segundo es más consistente en sus aproximaciones. En la figura 5b, segundo está más cerca de C

que a UN basado en sus distancias de separación exactas. Considerando que la distancia del centroide entre UN y segundo está bastante cerca de la distancia de separación que se aproxima, la distancia del centroide entre segundo y C es muchas veces mayor que la distancia de separación real entre segundo y C. De hecho, por sus distancias centroides, segundo está más cerca de UN que a C, invirtiendo así el orden inducido por sus distancias de separación. En general, si una colección de objetos tiene tamaños y formas muy variados, la aproximación de la distancia del centroide produciría agrupamientos deficientes (en relación con los agrupamientos producidos al utilizar las distancias de separación). Por otro lado, para el ejemplo mostrado en la Fig.5b, la aproximación MV conserva el orden que segundo está más cerca de C que a A. Ciertamente, un

5. Esta definición ignora cualquier objeto que pueda encontrarse entre UN y SEGUNDO.

La aproximación es una aproximación y no es difícil construir situaciones en las que la aproximación de MV puede ser inexacta. Sin embargo, al probar la aproximación en numerosos objetos poligonales que se pueden encontrar en mapas reales, hemos verificado que la aproximación MV es razonable y es mucho menos susceptible a variaciones en tamaños y formas que la aproximación centroide distancia. La sección 4.6.4 dará resultados experimentales sobre la calidad de los agrupamientos producidos por la aproximación MV.

4.4 Aproximación por la distancia de separación entre rectángulos isotéticos

Otra forma de aproximar la distancia de separación exacta entre dos polígonos UN; segundo es 1) calcular isotético
rectángulos $o_{UN;}$ $o_{segundo}$ y 2) calcular la distancia de separación entre yo $_{UN}$ y yo SEGUNDO.
Dado un polígono UN, el isotético
rectángulo o_{UN} es el rectángulo más pequeño que contiene UN, y cuyas aristas son paralelas a los ejes xo y.
De ahora en adelante, nos referiremos a esta aproximación a la distancia de separación exacta como Aproximación IR.

Para cualquier polígono dado UN, un rectángulo delimitador mínimo de UN se define como el rectángulo más pequeño, en el área, que contiene A. Como tal, un rectángulo delimitador mínimo no necesita tener sus bordes paralelos a los ejes x o y. Precisamente por esto, es relativamente costoso calcular un rectángulo delimitador mínimo. Por el contrario, el rectángulo isotético, aunque posiblemente tenga un área mayor que la de un rectángulo delimitador mínimo, se puede obtener fácilmente al encontrar el mínimo y el máximo de la

s F ets F v j v es la coordenada x de un vértice del polígono gramo, y
w j w es la coordenada y de un vértice del polígono gramo. Así,
El cálculo del paso 1 de la aproximación IR requiere una cantidad trivial de tiempo.

Al igual que calcular la distancia de separación exacta entre dos polígonos, como se describe en la Sección 4.2, calcular la distancia de separación exacta entre dos rectángulos isotéticos requiere dos pasos. En el primer paso, donde se verifica la posible intersección, solo se necesita un tiempo constante para los rectángulos isotéticos, pero el tiempo logarítmico al número de vértices de los polígonos. De manera similar, en el siguiente paso donde se calcula la distancia de separación real (necesaria cuando los dos rectángulos o polígonos no se cruzan), es tiempo constante para rectángulos isotéticos, pero tiempo logarítmico para polígonos. En particular, para rectángulos isotéticos, es suficiente llamar repetidamente un procedimiento que calcula la distancia entre un punto y un segmento de línea. Por lo tanto, el Paso 2 de la aproximación IR se puede realizar de manera eficiente.

De la eficiencia a la efectividad, tal como evaluamos en la sección anterior cómo la aproximación de MV afecta la calidad de las agrupaciones, aquí deberíamos hacer una pregunta similar. Por un lado, la aproximación IR es diferente de la aproximación MV en que la primera siempre subestima la distancia de separación real entre los polígonos originales. Esto se debe a que el rectángulo isotético de un polígono contiene el polígono. Por otro lado, al igual que la aproximación MV, siempre que la aproximación IR subestime todos los polígonos agrupados de manera bastante uniforme, los agrupamientos producidos serían de calidad comparable a los producidos utilizando la separación exacta distancias. La sección 4.6.4 proporcionará resultados experimentales que comparen la calidad de las agrupaciones producidas por estos dos enfoques.

Hasta ahora, hemos representado la aproximación IR (y la aproximación MV) como una optimización del rendimiento para calcular la distancia de separación exacta. En realidad, hay otra ventaja que ofrece la aproximación IR (y la aproximación MV). Es decir, no requiere que el polígono original sea convexo. Como se describió anteriormente, la definición del rectángulo isotético de un polígono se aplica igualmente bien a los polígonos convexos y no convexos. Por lo tanto, cuando se integra con la aproximación IR, CLARANS se puede utilizar para agrupar polígonos. Por el contrario, el método descrito en la Sección 4.2 solo funciona para polígonos convexos, por lo que también restringe CLARANS a polígonos convexos.

4.5 Memorización de distancias exactas y aproximadas

Recuerde de las secciones anteriores que, para cualquier par de objetos
$O_{metro;}$ $O_{j,}$ la distancia re re $O_{metro;}$ O_{j} Þ entre los dos objetos puede ser referenciada numerosas veces en una ejecución de
CLARANOS. Cuando los objetos son simplemente puntos, esta distancia se puede calcular dinámicamente cada vez que se necesite. Esto es suficiente porque calcular la distancia de Manhattan o euclidiana es una operación simple que tarda microsegundos en completarse.

Sin embargo, la situación es muy diferente cuando los objetos que se agrupan son polígonos. Como se mostrará experimentalmente en la Sección 4.6.2, calcular la similitud entre dos polígonos, incluso cuando se usa la aproximación IR o la aproximación MV, todavía toma milisegundos

completar. Desde la distancia re re $O_{metro;}$ O_{j} Þ puede ser necesario repetidamente, tiene sentido, una vez calculado, memorizar
la distancia. Esto aseguraría que la distancia entre cada par de objetos poligonales se calcule como máximo una vez. Claramente, esta estrategia de memorización cambia el espacio por la eficiencia del tiempo. La sección 4.6.5 proporcionará resultados experimentales que evalúen si la memorización es valiosa y si vale la pena esta compensación.

4.6 Evaluación experimental

4.6.1 Detalles de los experimentos

Recuerde que tanto la aproximación MV como la aproximación IR son aplicables para calcular la distancia de separación entre polígonos no convexos. Pero, debido a que el método descrito en la Sección 4.2 solo funciona para polígonos convexos, todos nuestros experimentos están restringidos a objetos poligonales que son convexos. Para generar polígonos aleatorios de diferentes formas, tamaños y orientaciones, usamos el generador de polígonos convexos desarrollado por Snoeyink para probar algunas de las ideas reportadas en [21]. El generador calcula y genera norte puntos uniformemente espaciados en una elipse con centro (ctrx; ctry), comenzando con el ángulo compensar, y tener mayor

y ejes menores paralelos a los ejes xey con radios $r_{1;}$ $r_{2,}$
respectivamente, donde norte; ctrx; ctry; compensar; $r_{1;}$ r_{2} son todas las entradas al generador. Usamos un generador de números aleatorios para
crear todas estas entradas, de tal manera que todos los polígonos generados se encuentran en una región rectangular con longitud l y ancho
w. Variando los valores de l y w podemos generar conjuntos de polígonos con diferentes densidades y, por tanto, con diferentes

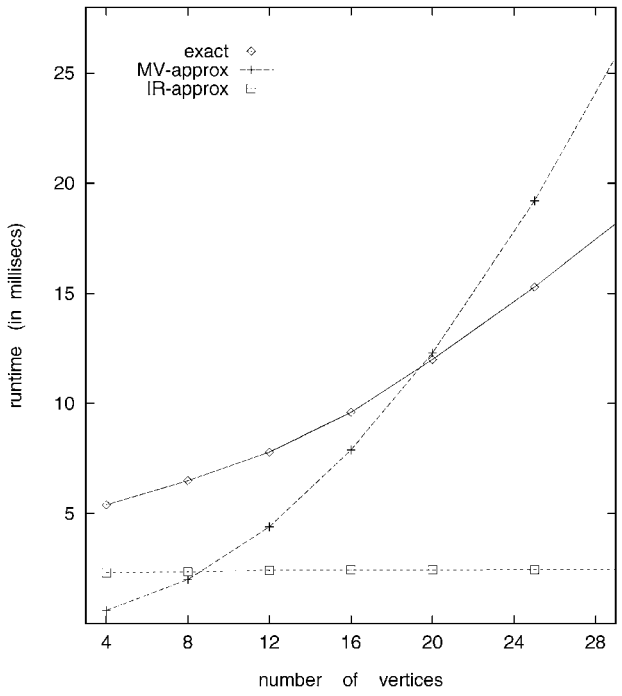

Fig. 6. Eficiencia de las aproximaciones.

proporciones de polígonos que se superponen con algunos otros polígonos del conjunto.

Todos los experimentos informados aquí se llevaron a cabo en una estación de trabajo SPARC-LX de tiempo compartido. Siempre que se utilizó CLARANS, todas las cifras relativas a CLARANS, debido a su naturaleza aleatoria, fueron cifras promedio obtenidas al ejecutar el mismo experimento 10 veces.

4.6.2 Eficiencia: distancia exacta versus aproximación IR versus aproximación MV

En esta serie de experimentos, usamos polígonos con diferentes números de vértices y registramos la cantidad promedio de tiempo de ejecución necesaria para calcular la distancia de separación exacta, su aproximación IR y su aproximación MV para un par de polígonos. La figura 6 muestra los resultados con los tiempos de ejecución registrados en milisegundos.

El enfoque de aproximación de IR es el claro ganador ya que siempre supera al enfoque de distancia de separación exacta, que supera al enfoque de aproximación de MV por un amplio margen cuando el número de vértices es alto, y eso, incluso cuando el número de vértices es pequeño, ofrece un rendimiento competitivo con el enfoque de aproximación MV. Recuerde de la sección 4.4 que se necesitan dos pasos para calcular la aproximación IR. El primer paso es calcular los rectángulos isotéticos, que tiene una complejidad de

O re norte Þ. El segundo paso es calcular la distancia de separación entre los rectángulos isotéticos, que tiene una complejidad de
O re 1 Þ. La planitud de la curva para la aproximación IR en la Fig. 6 muestra claramente que el segundo paso domina al primero. Por lo tanto, en conjunto, la aproximación de IR no varía a medida que aumenta el número de vértices.

Por el contrario, tanto la distancia de separación exacta como la aproximación MV requieren tiempos de ejecución más altos a medida que aumenta el número de vértices. Cuando el número de vértices es menor que 20, calcular la distancia de separación exacta requiere más tiempo que hace la aproximación MV. Pero lo contrario es cierto cuando el número de vértices excede 20. En otras palabras, el tiempo de ejecución para la aproximación MV crece más rápido que para calcular la distancia de separación exacta. Esto es consistente con los resultados de complejidad presentados en las Secciones 4.2 y 4.3.

4.6.3 Efectividad de agrupamiento: distancia exacta versus aproximación IR versus aproximación MV

Si bien la serie anterior de experimentos no implica agrupaciones, en esta serie de experimentos, CLARANS se integró con las tres formas diferentes de calcular distancias entre polígonos. Debido a que el número de vértices de polígonos es un parámetro que afecta el rendimiento de los tres enfoques, ejecutamos CLARANS con conjuntos de norte- cara

polígonos donde norte varía de 4 a 20, y con una mezcla aleatoria de polígonos, cada uno de los cuales tiene entre 4 y 20 aristas. Para esta serie de experimentos, nos centramos tanto en la eficacia como en la eficiencia de la agrupación.

Con respecto a la efectividad de las dos aproximaciones en relación con el enfoque de distancia exacta, recuerde que la aproximación de IR siempre subestima la distancia exacta, mientras que la aproximación de MV siempre la sobreestima. Por lo tanto, no es apropiado medir simplemente la efectividad con base en la distancia aproximada promedio entre un polígono y el medoide de su grupo. En cambio, comparamos los grupos producidos por el enfoque de distancia exacta y los grupos producidos por las dos aproximaciones. Más precisamente, para un polígono en particular UN, calculamos la relación:

la distancia exacta entre UN y METRO $_{0\ UN}$ dividido por el exacto distancia entre UN y METRO $_{UN,}$

dónde METRO $_{0\ UN}$ es el medoide del racimo UN está asignado a usando la aproximación y METRO $_{UN}$ es el medoide correspondiente usando la distancia exacta. Cuanto más cerca de 1 sea la relación, más precisa la aproximación.

Como resulta que para casi todos los conjuntos de polígonos con los que experimentamos, más del 90 por ciento de los polígonos satisfacen METRO $_{UN}$ ¼ METRO $_{A.}$ Es decir, a pesar de las aproximaciones, las estructuras de agrupamiento están muy conservadas. Más lejos-
Además, la Tabla 2 muestra el promedio de razón antes mencionado en todos los polígonos A.

La calidad de las agrupaciones producidas por la aproximación IR y la aproximación MV es casi idéntica a la calidad de la agrupación producida utilizando la distancia de separación exacta, que difiere en aproximadamente un 2-3 por ciento. Esto muestra claramente que las dos aproximaciones son muy efectivas, ya que subestiman o sobreestiman las distancias reales de manera tan consistente que los grupos se conservan. Por lo tanto, se justifica el uso de la aproximación IR y la aproximación MV como formas de optimizar el rendimiento.

4.6.4 Eficiencia de agrupamiento: distancia exacta versus aproximación IR versus aproximación MV

A continuación, consideramos la eficiencia de las dos aproximaciones en relación con el enfoque de distancia exacta. La Fig. 7 muestra los tiempos que necesita CLARANS para agrupar un número variable de polígonos que tienen 10 aristas y un número variable de polígonos que tienen entre 4 y 20 aristas. En ambos casos, la aproximación IR y la aproximación MV superan considerablemente el enfoque de distancia de separación exacta. En

TABLA 2
El promedio de la relación entre todos los polígonos UN

Approach	Average Ratio
MV-approximation	1.02
IR-approximation	1.03

En particular, la aproximación IR es siempre la más eficiente, requiriendo solo entre el 30 y el 40 por ciento del tiempo necesario para la aproximación de distancia exacta.

Otros conjuntos de polígonos con los que experimentamos, incluidos algunos que tienen densidades variables, también dan la misma conclusión. Por lo tanto, dado que la aproximación IR es capaz de generar agrupaciones de calidad casi idéntica a las producidas por el enfoque exacto, la aproximación IR es la elección definitiva para CLARANS. Otros experimentos que llevamos a cabo indican que se puede sacar la misma conclusión si se utilizaran PAM y CLARA para agrupar objetos poligonales.

4.6.5 Efectividad de la memorización de distancias calculadas

Si bien los gráficos de la Fig. 7 identifican el enfoque de aproximación IR como el claro ganador, los resultados de rendimiento de la aproximación son decepcionantes. Por ejemplo, se necesitan
1,000 segundos para agrupar 100 polígonos. Recuerde de la Fig. 6 que la aproximación de IR para un par de polígonos toma de dos a tres milisegundos. Para 100 polígonos, hay 100 * 100/2 =
5,000 pares de polígonos. Estas 5.000 distancias tardan un total de 10 a 15 segundos en calcularse. Por lo tanto, lo que está sucediendo es que la distancia entre cada par de polígonos se calcula en promedio de 60 a 100 veces. Esto argumenta con mucha fuerza por qué las distancias calculadas deben memorizarse como una optimización del rendimiento. La Fig.8 muestra los resultados de aplicando memorización al mismo conjunto de polígonos usados en la Fig. 7b. De hecho, con la memorización, el tiempo necesario para agrupar 100 polígonos con la aproximación IR cae a unos 10 segundos, como se estimó anteriormente. Se obtienen ganancias de rendimiento similares si se utilizan la distancia de separación exacta o la aproximación de MV.

4.7 Resumen

Con respecto a cómo calcular las similitudes entre objetos, nuestros resultados experimentales indican claramente que el enfoque de aproximación IR es la elección. Si bien solo se aproxima a la distancia de separación exacta, el enfoque de aproximación IR puede producir agrupaciones de calidad casi idéntica a las agrupaciones producidas por el enfoque de distancia exacta. El factor decisivo es entonces la eficacia del método de aproximación IR. Tiene la propiedad deseable de que su tiempo de cálculo no varía con el número de vértices de los objetos poligonales y que supera el enfoque de distancia exacta típicamente de tres a cuatro veces. Además, se puede utilizar tanto para objetos poligonales convexos como no convexos.

Si bien el material de esta sección se centra en colecciones de objetos que son todos polígonos, los resultados se pueden generalizar fácilmente a colecciones heterogéneas de objetos, donde algunos objetos son polígonos y los restantes son puntos. Para tales colecciones, la similitud entre un punto y un polígono se puede definir como la distancia entre el punto y el rectángulo isotético del polígono. Esta distancia se puede calcular en tiempo constante.

5 Conclusiones

En este artículo, presentamos un algoritmo de agrupamiento llamado CLARANS que se basa en una búsqueda aleatoria. por

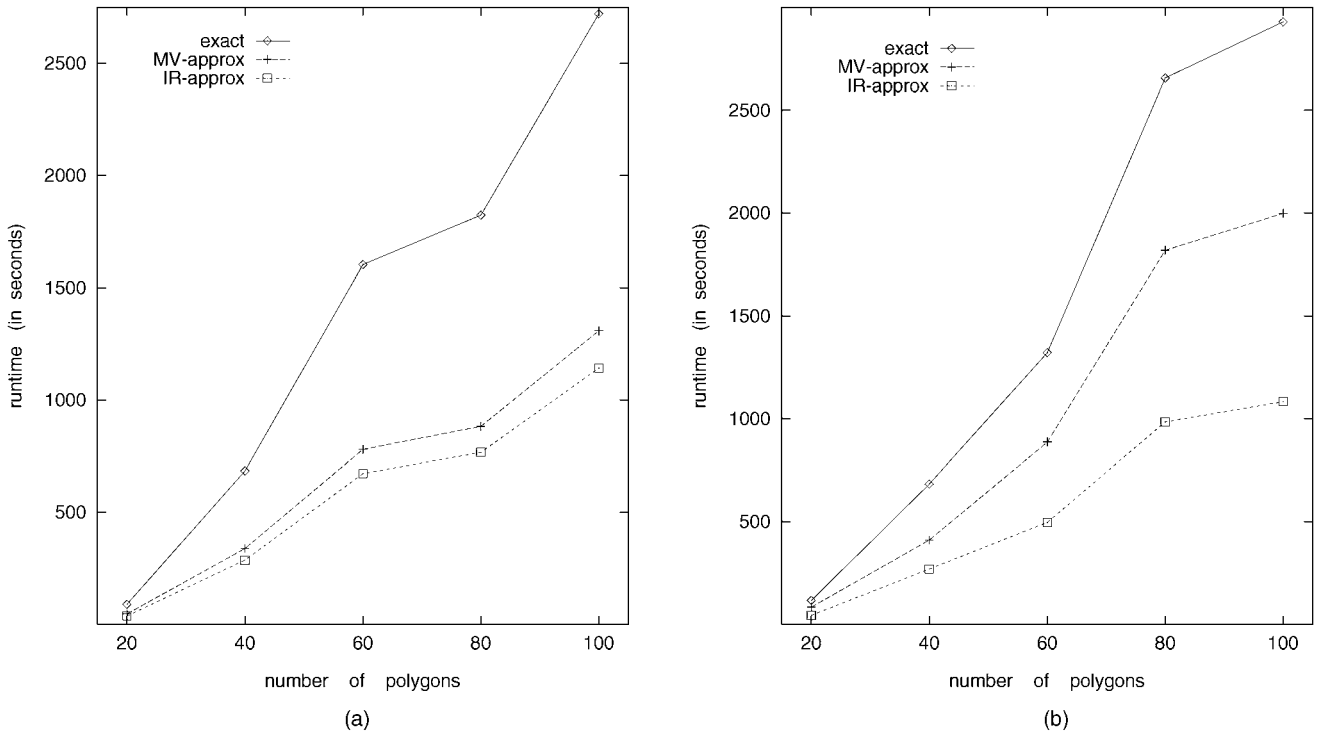

Fig. 7. Eficiencia de agrupamiento de las aproximaciones. (a) Polígonos de 10 lados. (b) Polígonos de 4 a 20 lados.

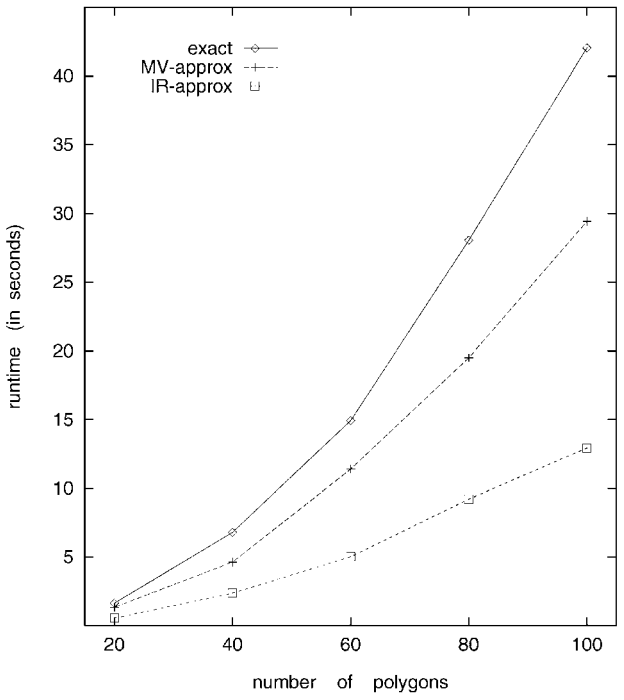

Fig. 8. Eficiencia de agrupamiento con memorización de distancia.

pequeños conjuntos de datos, CLARANS es algunas veces más rápido que PAM; la brecha de rendimiento para conjuntos de datos más grandes es aún mayor. En comparación con CLARA, CLARANS tiene la ventaja de que el espacio de búsqueda no está localizado en un subgrafo específico elegido a priori, como en el caso de CLARA. En consecuencia, cuando se le da la misma cantidad de tiempo de ejecución, CLARANS puede producir agrupaciones de mucha mejor calidad que las generadas por CLARA.

También hemos estudiado cómo los objetos poligonales pueden ser agrupados por CLARANOS. Hemos propuesto tres formas diferentes de calcular la distancia entre dos polígonos. La complejidad y los resultados experimentales indican que la aproximación IR es algunas veces más rápida que el método que calcula la distancia de separación exacta. Además, los resultados experimentales muestran que, a pesar del tiempo de ejecución mucho menor, la aproximación IR es capaz de encontrar agrupaciones que son de calidad casi tan buena como las producidas utilizando las distancias de separación exactas. En otras palabras, la aproximación IR puede proporcionar una ganancia de eficiencia significativa, pero sin pérdida de efectividad.

En el trabajo en curso, estamos desarrollando un paquete que utiliza la agrupación en clústeres como base para proporcionar muchas operaciones de minería de datos espaciales. Los algoritmos de minería de datos espaciales SD (CLARANS) y NSD (CLARANS) descritos en [25] son dos ejemplos. También existen operaciones de minería con mapas [32].

UN AGRADECIMIENTOS

La investigación de RT Ng fue patrocinada parcialmente por subvenciones NSERC OGP0138055 y STR0134419, y subvenciones IRIS-3.

La investigación de J. Han fue parcialmente apoyada por la subvención OGP03723 de NSERC y las subvenciones NCE / IRIS-3.

R EFERENCIAS

[1] R. Agrawal, J. Gehrke, D. Gunopulos y P. Raghavan, "Agrupación subespacial automática de datos de alta dimensión para aplicaciones de minería de datos", Proc. 1998 ACM-SIGMOD, págs. 94-105,
1998.

[2] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer y A. Swami, "An Interval Classifier for Database Mining Applications", Proc. 18ª Conf. Bases de datos muy grandes, págs. 560–573, 1992.

[3] R. Agrawal, T. Imielinski y A. Swami, "Reglas de asociación minera entre conjuntos de elementos en grandes bases de datos", Proc. 1993 ACM Grupo de Interés Especial sobre Gestión de Datos, págs. 207–216, 1993.

[4] M. Ankerst, M. Breunig, H.-P. Kriegel y J. Sander, "ÓPTICA: puntos de ordenación para identificar la estructura de agrupación", Proc. 1999 ACM Grupo de Interés Especial sobre Gestión de Datos, págs. 49–60, 1999.

[5] WG Aref y H. Samet, "Estrategias de optimización para el procesamiento de consultas espaciales", Proc. 17ª Conf. Bases de datos muy grandes, págs. 81-90,
1991.

[6] A. Borgida y RJ Brachman, "Loading Data into Description Reasoners", Proc. 1993 ACM Grupo de Interés Especial sobre Gestión de Datos, págs. 217-226, 1993.

[7] P. Bradley, U. Fayyad y C. Reina, "Scaling Clustering Algorithms to Large Databases", Proc. Cuarta Conf. Int'l Descubrimiento de conocimiento y minería de datos, págs. 9-15, 1998.

[8] T. Brinkhoff y H.-P. Kriegel, B. Seeger, "Procesamiento eficiente de combinaciones espaciales mediante árboles R", Proc. 1993 ACM Grupo de Interés Especial sobre Gestión de Datos, págs. 237-246, 1993.

[9] D. Dobkin y D. Kirkpatrick, "A Linear Algorithm for Determining the Separation of Convex Polyhedra", J. Algoritmos,
vol. 6, no. 3, págs. 381–392, 1985.

[10] M. Ester, H. Kriegel y X. Xu, "Descubrimiento de conocimientos en grandes bases de datos espaciales: técnicas de enfoque para una identificación de clases eficiente", Proc. Cuarto Symp Int'l. Grandes bases de datos espaciales (SSD '95), págs. 67–82, 1995.

[11] M. Ester, H. Kriegel, J. Sander y X. Xu, "Un algoritmo basado en la densidad para descubrir grandes clústeres en grandes bases de datos espaciales con ruido", Proc: Segunda Conf. Int. Descubrimiento de conocimiento y minería de datos, 1996.

[12] S. Guha, R. Rastogi y K. Shim, "CURE: An Efficient Clustering Algorithm for Large Databases", Proc. 1998 Grupo de interés especial de ACM sobre gestión de datos, págs. 73–84, 1998.

[13] O. Günther, "Computación eficiente de uniones espaciales", Proc. Novena Conf. Ing. De datos, págs. 50-60, 1993.

[14] J. Han, Y. Cai y N. Cercone, "Descubrimiento del conocimiento en bases de datos: un enfoque orientado a atributos", Proc. 18ª Conf. Bases de datos muy grandes, págs. 547–559, 1992.

[15] A. Hinneburg y DA Keim, "Un enfoque eficiente para la agrupación en clústeres en grandes bases de datos multimedia con ruido", Proc. 1998 Conf. Internacional. Descubrimiento de conocimiento y minería de datos, págs. 58–65,
1998.

[dieciséis] Y. Ioannidis y Y. Kang, "Algoritmos aleatorios para optimizar consultas de unión grandes", Proc. 1990 ACM Grupo de Interés Especial sobre Gestión de Datos, págs. 312–321, 1990.

[17] Y. Ioannidis y E. Wong, "Optimización de consultas mediante recocido simulado", Proc. 1987 Grupo de interés especial de ACM sobre gestión de datos, págs. 9-22, 1987.

[18] G. Karypis, E.-H. Han y V. Kumar, "CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling",
Computadora, vol. 32, no. 8, págs. 68–75, agosto de 1999.

[19] L. Kaufman y PJ Rousseeuw, Encontrar grupos en los datos: una introducción al análisis de conglomerados. John Wiley & Sons, 1990.

[20] D. Keim, H. Kriegel y T. Seidl, "Supporting Data Mining of Large Databases by Visual Feedback Queries", Proc. 10ª Conf. Ing. De datos, 1994.

[21] D. Kirkpatrick y J. Snoeyink, "Poda y búsqueda tentativas para calcular puntos fijos con aplicaciones a la computación geométrica", Proc. Noveno ACM Symp. Geometría Computacional, págs. 133–
142, 1993.

[22] R. Laurini y D. Thompson, Fundamentos de los sistemas de información espacial. Prensa académica, 1992.

[23] W. Lu, J. Han y B. Ooi, "Descubrimiento del conocimiento general en grandes bases de datos espaciales", Proc. Taller de sistemas de información geográfica del Lejano Oriente, págs. 275-289, 1993.

[24] G. Milligan y M. Cooper, "Un examen de los procedimientos para determinar el número de clústeres en un conjunto de datos", Psicometrika
vol. 50, págs. 159-179, 1985.

[25] R. Ng y J. Han, "Métodos de agrupación en clústeres eficientes y eficaces para la minería de datos espaciales", Proc. 20ª Conf. Bases de datos muy grandes, págs. 144-155, 1994.
[26] G. Piatetsky-Shapiro y WJ Frawley, Descubrimiento de conocimiento en bases de datos. Prensa AAAI / MIT, 1991.
[27] F. Preparata y M. Shamos, Geometría Computacional. Nueva York: Springer-Verlag, 1985.
[28] H. Samet, El diseño y análisis de estructuras de datos espaciales. Addison-Wesley, 1990.
[29] G. Sheikholeslami y S. Chatterjee, A. Zhang, "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases", Proc. 1998 Conf. Bases de datos muy grandes, págs. 428–439, 1998.
[30] H. Spath, Disección y análisis de conglomerados: teoría, programas FORTRAN, ejemplos. Ellis Horwood Ltd., 1985.
[31] W. Wang y J. Yang, R. Muntz, "STING: Un enfoque de cuadrícula de información estadística para la minería de datos espaciales", Proc. 23a Conf. Bases de datos muy grandes, págs. 186-195, 1997.
[32] Y. Yu, "Encontrar características sólidas, comunes y discriminatorias de clústeres a partir de mapas temáticos", Tesis de maestría, Departamento de Ciencias de la Computación, Univ. de Columbia Británica, 1996.
[33] T. Zhang y R. Ramakrishnan, M. Livny, "BIRCH: un método eficiente de agrupación de datos para bases de datos muy grandes", Proc. Grupo de Interés Especial de ACM sobre Gestión de Datos, págs. 103-114, 1996.

Raymond T. Ng recibió el doctorado en ciencias de la computación de la Universidad de Maryland, College Park, en 1992. Desde entonces, ha sido profesor asistente y asociado en la Universidad de Columbia Británica. Sus intereses de investigación actuales incluyen minería de datos, bioinformática y sistemas de bases de datos multimedia. Ha publicado más de 80 artículos en revistas y conferencias, y ha formado parte de muchos comités de programas de ACM-SIGMOD, VLDB y SIG-KDD.

Jiawei Han recibió el doctorado en ciencias de la computación de la Universidad de Wisconsin en Madison en 1985. Actualmente es el director del Laboratorio de Investigación de Sistemas de Bases de Datos Inteligentes y profesor en la Escuela de Ciencias de la Computación de la Universidad Simon Fraser, Canadá. Ha realizado investigaciones en las áreas de sistemas de bases de datos, minería de datos, almacenamiento de datos, minería de datos geoespaciales, minería web, base de datos deductiva y orientada a objetos.

sistemas e inteligencia artificial, con más de 150 publicaciones en revistas o conferencias. Actualmente es líder de proyecto de las Redes de Centros de Excelencia de Canadá (NCE) / proyecto IRIS-3 “Construcción, consulta, análisis y minería de almacenes de datos en Internet” (1998–
2002) y el arquitecto jefe del sistema DBMiner. Ha servido o está sirviendo actualmente en los comités de programa para más de 50 conferencias y talleres internacionales. También ha formado parte de los consejos editoriales de Transacciones IEEE sobre conocimiento e ingeniería de datos, minería de datos y descubrimiento de conocimiento, y el Revista de Sistemas de Información Inteligentes. Es miembro de la IEEE Computer Society, ACM, ACM-SIGMOD, ACM-SIGKDD y AAAI.