Algoritmos de agrupamiento

D. Pascual [1], F. Pla [2], S. Sánchez [2]

1. Departamento de Computación
Universidad de Oriente
Avda. Patricio Lumumba S/N. 90500 Santiago de Cuba, Cuba
dpascual@csd.uo.edu.cu

2. Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I
Avda. Sos Baynat S/N. 12071 Castellón, España
{pla, sanchez}@lsi.uji.es

Resumen

En el presente trabajo mostramos algunos de los resultados del estudio realizado sobre diferentes técnicas de agrupamiento, las que se encuentran dentro del campo de estudio del Reconocimiento de Patrones, haciendo hincapié en las técnicas basadas en densidad. Mostramos el estudio comparativo de tres algoritmos empleando bases de datos reales y artificiales.

Palabras clave: Algoritmos de agrupamiento, Algoritmos basados en densidad.

1 Introducción

Debido al desarrollo alcanzado con los microprocesadores se pueden manejar grandes volúmenes de datos como puntos en espacios de una alta dimensionalidad, éstos aparecen en varias esferas de la vida real, en bases de datos de corporaciones financieras, telecomunicaciones, medicina, imágenes de satélite, etc., numerosas son las aplicaciones en las que se requiere del manejo de bases de datos espaciales con el objetivo de conocer acerca de la identificación de grupos, para descubrir importantes distribuciones del espacio en estudio, lo cual puede ser resuelto con el empleo de algún algoritmo de agrupamiento conveniente, por lo que el estudio, aplicación y creación de nuevos algoritmos constituye un desafío importante en la actualidad.

El Reconocimiento de Formas constituye un amplio conjunto de técnicas para el tratamiento de datos entre las que se puede mencionar: la selección y extracción de características, la clasificación de un objeto en un grupo dado y la división de los datos en grupos (agrupamiento). Uno de los enfoques en el Reconocimiento de Formas, en función del tipo de espacio de representación utilizado es el

Reconocimiento Estadístico de Formas que se apoya en la Teoría de Decisión, éste asume que el espacio de representación tiene una estructura de espacio vectorial y/o métrico y no se supone ninguna relación estructural entre las distintas características. Dentro de este enfoque, se distingue entre las aproximaciones paramétrica y no paramétrica. En el primer caso, se asume un conocimiento a priori sobre la forma funcional de las distribuciones de probabilidad de cada clase sobre el espacio de representación, las cuales vendrán determinadas por un conjunto finito y normalmente fijo de parámetros, las fronteras de decisión estarán definidas por dichas distribuciones de clases. La aproximación no paramétrica no supone ninguna forma de las distribuciones de probabilidad sobre el espacio de representación, de modo que el único conocimiento a priori será el correspondiente a la información inducida a partir de un conjunto de muestras, las fronteras de decisión estarán determinadas por las muestras del conjunto de entrenamiento. En este trabajo nos interesa el Reconocimiento Estadístico de Formas no paramétrico, pero haremos una pequeña alusión a la aproximación paramétrica con el estudio de las mixturas finitas.

El problema del agrupamiento puede definirse como sigue: dados n puntos en un espacio n-dimensional particionar los mismos en k grupos tales que los puntos dentro de un grupo son más similares que cada uno a los de los otros grupos, dicha similaridad se mide atendiendo a alguna función distancia (función de disimilaridad) o alguna función de similaridad.

También es importante para el mejor funcionamiento de los algoritmos de agrupamiento la detección de ruido para evitar la influencia negativa de éstos en la formación de los grupos, así como la estimación del número correcto de grupos a determinar.

Los métodos de agrupamiento no paramétricos pueden dividirse en tres grupos fundamentales: jerárquicos [1-4], particionales [1] y basados en densidad [5-9]. En el presente trabajo mostramos los resultados del estudio realizado acerca de los algoritmos de agrupamiento existentes y con más detalles los basados en densidad así como algunos experimentos realizados para comparar algunos de ellos. El trabajo está estructurado de la siguiente forma: en la sección 2 se describen varios algoritmos de agrupamiento, en la sección 3 se muestran algunos experimentos realizados para comparar tres algoritmos y en la sección 4 las conclusiones.

2 Algoritmos de agrupamiento

El problema de formar grupos en un conjunto de datos es muy importante para el conocimiento del comportamiento de una población de la cual solo se tiene una

cantidad n de sus elementos. La solución de estos problemas se realiza mediante la creación de algoritmos de agrupamiento.

Entre los métodos de agrupamiento paramétricos se encuentran las mixturas finitas, éstas son una poderosa herramienta para modelar densidades de probabilidad de conjuntos de datos univariados y multivariados, modelan observaciones las cuales se asume que han sido producidas por un conjunto de fuentes aleatorias alternativas e infieren los parámetros de estas fuentes para identificar qué fuente produjo cada observación, lo que lleva a un agrupamiento del conjunto de observaciones.

Los métodos de agrupamiento no paramétricos pueden dividirse en tres grupos fundamentales: jerárquicos, particionales y basados en densidad.

Los algoritmos jerárquicos son aquellos en los que se va particionando el conjunto de datos por niveles, de modo tal que en cada nivel generalmente , se unen o se dividen dos grupos del nivel anterior, según si es un algoritmo aglomerativo o divisivo.

Los algoritmos particionales son los que realizan una división inicial de los datos en grupos y luego mueven los objetos de un grupo a otro según se optimice alguna función objetivo.

Los algoritmos basados en densidad enfocan el problema de la división de una base de datos en grupos teniendo en cuenta la distribución de densidad de los puntos, de modo tal que los grupos que se forman tienen una alta densidad de puntos en su interior mientras que entre ellos aparecen zonas de baja densidad.

2.1 Mixturas finitas

Sea $Y=[Y_1, Y_2, \ldots, Y_d]^T$ una variable aleatoria d-dimensional con $y=[y_1, y_2, \ldots, y_d]^T$ representando un resultado particular de Y. se dice que Y sigue una distribución de mixtura finita con k componentes si su función de densidad de probabilidad se puede escribir por:

$$p(y/\theta) = \sum_{m=1}^{k} \alpha_m \, p(y/\theta_m) \qquad (1)$$

donde $\alpha_1, \alpha_2, \ldots \alpha_k$ son probabilidades mezclantes (probabilidades a priori) que nos indican el grado de importancia de cada uno de los k modos, cada Θ_m es el vector de parámetros que define la m-ésima componente, $\Theta = \{\Theta_1, \ldots, \Theta_k, \alpha_1, \ldots,$

$\alpha_k\}$ es el conjunto completo de parámetros necesarios para especificar la mixtura, por supuesto las α_m deben satisfacer:

$\alpha_m \geq 0$ para todo m=1,…, k y $\sum_{m=1}^{k} \alpha_m = 1$ (2).

La opción usual para obtener los estimados de los parámetros es el algoritmo Expectation-Maximization (EM) [10], que produce una secuencia de estimados de Θ aplicando alternativamente dos pasos (E-paso y M-paso) hasta obtener un máximo local de: $\log p(Y/\theta)$, siendo Y un conjunto de n muestras independientes e idénticamente distribuidas.

Este algoritmo tiene varias limitaciones: es un método local, por tanto es sensible a la inicialización; puede converger a la frontera del espacio de parámetros donde la verosimilitud es no acotada llevando a estimados sin sentido; si el número de componentes es muy grande puede sobre-entrenar los datos pues estos son incompletos y por tanto se puede obtener una forma más irregular de lo que verdaderamente es, mientras que una mixtura con pocas componentes no es lo suficientemente flexible para aproximar al verdadero modelo; la finalización también es una limitación, pues llega un momento donde el proceso deja de evolucionar por lo que se supone que alcanza la localización óptima pero esto no nos asegura la verdadera distribución.

En el algoritmo que se describe en [10] se implementa el criterio de Mínima longitud de un Mensaje usando una variante del EM, así reduce las limitaciones del mismo, es robusto con respecto a la inicialización, elimina el problema de la convergencia a la frontera del espacio de parámetros, comienza por un valor de k grande k_{max} hasta otro menor k_{min} y obtiene buenos resultados debido a que emplean la variante de considerar solamente las componentes de probabilidad no cero para obtener los estimados de los parámetros. Se puede usar para cualquier tipo de modelo de mixtura, en sus experimentos lo emplearon para mixturas de gaussianas y a diferencia del EM este algoritmo lo que hace es minimizar la función objetivo:

$$L(\theta, y) = \frac{N}{2} \sum_{m:\alpha_m>0} \log\left(\frac{n\alpha_m}{12}\right) + \frac{k_{nz}}{2}\log\left(\frac{n}{12}\right) + \frac{k_{nz}(N+1)}{2} - \log(p(y/\theta)$$

donde N es la dimensión del parámetro Θ_m y k_{nz} el total de probabilidades no nulas.

2.2 Algoritmos jerárquicos

Las estrategias jerárquicas más conocidas son Single Link (SL), Average Link (AL) y Complete Link (CL). De manera muy breve estos algoritmos se describen como:

- **(SL)**: En cada paso se unen los dos grupos cuyos elementos más cercanos tienen la mínima distancia.
- **(AL)**: En cada paso se unen los dos grupos tal que tienen la mínima distancia promedio entre sus puntos.
- **(CL)**: En cada paso se unen los dos grupos tal que su unión tiene el diámetro mínimo o los dos grupos con la menor distancia máxima entre sus elementos.

Algoritmo Chameleon

Tiene dos fases [2]. Durante la primera fase construye el grafo de los k vecinos más cercanos y usa un algoritmo de particionamiento de grafo para agrupar los puntos en subgrupos.

Durante la segunda fase, usa un algoritmo jerárquico aglomerativo para encontrar los clusters genuinos combinando repetidamente estos subgrupos. En esta segunda fase determina el par de subgrupos más similares tomando en cuenta su interconectividad y cercanía, éstas expresan las características internas de los subgrupos, el modelo no es estático, si no que es capaz de adaptarse a las características internas de los subgrupos según estos van cambiando.

Este algoritmo es diferente del algoritmo SL porque permite unir varios pares de subgrupos en la misma iteración.

Un nuevo criterio de aislamiento de grupos

En este algoritmo se integra un criterio de aislamiento de grupos en un algoritmo de agrupamiento jerárquico aglomerativo [11].

Define el incremento de la disimilaridad o gap entre dos grupos C_i y C_j, el criterio de aislamiento es el siguiente: dados dos grupos C_i y C_j candidatos para unir, si $gap_i \geq t_i$ ($gap_j \geq t_j$) aislar el cluster C_i (C_j) y continuar la estrategia de agrupamiento con el resto de los patrones. Si ninguno de los grupos excede el límite del gap unirlos. Emplea un umbral dinámico (t_i) que va variando a lo largo del algoritmo y en dependencia del grupo C_i, esto lo diferencia del clásico

algoritmo SL en el que el umbral para decidir la unión de los dos grupos más cercanos es fijo.

El algoritmo comienza con cada patrón en un grupo, en cada nivel del algoritmo se determina el par de grupos más similares según si tienen los dos puntos más similares entre los puntos de grupos diferentes y se aplica el criterio de aislamiento, los clusters son entonces o unidos o aislados (uno o ambos).

2.3 Algoritmos particionales

Estos algoritmos asumen un conocimiento a priori del número de clusters en que debe ser dividido el conjunto de datos, llegan a una división en clases que optimiza un criterio predefinido o función objetivo. Entre los algoritmos que emplean esta técnica podemos mencionar: K-means [1], ISODATA [4], CLARANS, [2] etc.

Algoritmo K-Means (Mac Queen 1967)

Es uno de los más simples y conocidos algoritmos de agrupamiento, sigue una forma fácil y simple para dividir una base de datos dada en k grupos (fijados a priori).

La idea principal es definir k centroides (uno para cada grupo) y luego tomar cada punto de la base de datos y situarlo en la clase de su centroide más cercano. El próximo paso es recalcular el centroide de cada grupo y volver a distribuir todos los objetos según el centroide más cercano. El proceso se repite hasta que ya no hay cambio en los grupos de un paso al siguiente [1].

El problema del empleo de estos esquemas es que fallan cuando los puntos de un grupo están muy cerca del centroide de otro grupo ver ejemplo en [2], también cuando los grupos tienen diferentes tamaños y formas.

Algoritmo CURE

Constituye un algoritmo híbrido entre los dos enfoques jerárquico y particional [4], que trata de emplear las ventajas de ambos y de eliminar las limitaciones. En éste, en lugar de usar un solo punto como representante de un grupo se emplea un número c de puntos representativos del grupo. La similaridad entre dos grupos se mide por la similaridad del par de puntos representativos más cercanos, uno de cada grupo.

Para tomar los puntos representativos selecciona los c puntos más dispersos del grupo y los atrae hacia el centro del mismo por un factor de contracción α, en cada

paso se unen los dos grupos más cercanos y una vez unidos se vuelve a calcular para éste su centro y los c puntos representativos.

Con este algoritmo se encuentran grupos de diferentes tamaños y formas, con ese método de sacar c puntos representativos y atraerlos hacia el centro del grupo CURE maneja los puntos ruido y outliers presentes en la base de datos.

2.4 Algoritmos basados en densidad

Estos algoritmos usan diversas técnicas para determinar dichos grupos las que pueden ser por grafos, basadas en histogramas, kernels, aplicando la regla k-NN, empleando los conceptos de punto central, borde o ruido. Entre ellos podemos mencionar los algoritmos DBSCAN, OPTICS, KNNCLUST y SNN.

El primer algoritmo que emplea este enfoque para dividir el conjunto de datos es DBSCAN [5], en este aparecen los conceptos: punto central, borde y ruido los que serán empleados para determinar los diferentes clusters. Otros algoritmos basados en densidad que siguen la línea de DBSCAN son: OPTICS y GDBSCAN. Veamos algunos algoritmos.

Algoritmo DBSCAN

Es el primer algoritmo basado en densidad [5], se definen los conceptos de punto central (puntos que tienen en su vecindad una cantidad de puntos mayor o igual que un umbral especificado), borde y ruido.

El algoritmo comienza seleccionando un punto p arbitrario, si p es un punto central, se comienza a construir un grupo y se ubican en su grupo todos los objetos denso-alcanzables desde p. Si p no es un punto central se visita otro objeto del conjunto de datos. El proceso continúa hasta que todos los objetos han sido procesados. Los puntos que quedan fuera de los grupos formados se llaman puntos ruido, los puntos que no son ni ruido ni centrales se llaman puntos borde.

De esta forma DBSCAN construye grupos en los que sus puntos son o puntos centrales o puntos borde, un grupo puede tener más de un punto central.

Algoritmo OPTICS

La motivación para la realización de este algoritmo se basa en la necesidad de introducir parámetros de entrada en casi todos los algoritmos de agrupamiento existentes que en la mayoría de los casos son difíciles de determinar, además en conjuntos de datos reales no existe una manera de determinar estos parámetros

globales, el algoritmo OPTICS trata de resolver este problema basándose en el esquema del algoritmo DBSCAN creando un ordenamiento de la base de datos para representar la estructura del agrupamiento basada en densidad, además puede hacer una representación gráfica del agrupamiento incluso para conjuntos de datos grandes, este algoritmo esta descrito en [6].

Algoritmo KNNCLUST

Propone utilizar como regla basada en densidad la de los k vecinos más cercanos (k-NN) [7] para tratar bases de datos de alta dimensionalidad como imágenes de satélites. La regla k-NN ha sido extensamente usada en muchos métodos de clasificación donde existe un conjunto de objetos etiquetados [12].

El algoritmo comienza asignando cada punto a un cluster individual. Se calcula para cada punto sus k vecinos más cercanos.

Para cada punto x de la base de datos se aplica la regla k-NN y x se asigna a un grupo según esta regla, este proceso se repite hasta que ninguno de los objetos cambia de grupo en dos iteraciones sucesivas.

En este algoritmo se determina el número de grupos de manera automática. Necesita la entrada de un solo parámetro: la cantidad de vecinos.

Algoritmo SNN

La motivación para la creación de este algoritmo es la existencia de bases de datos de alta dimensionalidad tales como textos y series de tiempo, así como la existencia de grupos de diferentes formas y tamaño [8]. El enfoque de los vecinos más cercanos compartidos fue primero introducido por Jarvis y Patrick , una idea similar fue presentada más tarde en ROCK [13,14].

Este algoritmo primero encuentra los vecinos más cercanos de cada punto de la base de datos y define la similaridad entre cada par de puntos en términos de cuántos vecinos más cercanos los dos puntos comparten. Usando esta definición de similaridad elimina ruido y outliers, identifica puntos centrales y construye grupos alrededor de éstos, el uso de una definición de similaridad con los vecinos más cercanos compartidos elimina problemas de diferentes densidades mientras que con el uso de los puntos centrales maneja los problemas de forma y tamaño. Para los pesos de los enlaces entre dos puntos en el grafo de los vecinos más cercanos compartidos (SNN) se toma en cuenta el ordenamiento de los vecinos más cercanos. Encuentra de manera natural la cantidad de grupos.

Algoritmo DENCLUE

Es un algoritmo de dos fases [9]. Dados N puntos de una base de datos D provista de una distancia d, y $x \in D$, define función de densidad, gradiente y punto atractor. En la primera fase divide el hiper-rectángulo del conjunto de datos en hipercubos de aristas de longitud 2σ. Determina cuales son los hipercubos más poblados y los hipercubos que están conectados.

En la segunda fase, considera solamente los hipercubos más poblados y los conectados a hipercubos más poblados para determinar los grupos, para cada x en estos hipercubos determina el valor de la función de densidad pero considerando solamente aquellos puntos x´ tales que su distancia al centro del hipercubo al que x pertenece sea menor o igual que 4σ, similarmente halla el gradiente y el atractor de densidad para x para el que la función de densidad sea mayor o igual que un valor ξ y clasifica a x en la clase de su atractor.

Con este algoritmo se obtiene el número de grupos de manera natural.

También existen algoritmos híbridos de las técnicas particionales y basados en densidad [15].

2.5 Otros enfoques en los algoritmos de agrupamiento

Actualmente hay otros enfoques empleando la técnica de multiclasificadores, entre ellos podemos mencionar un algoritmo que emplea el algoritmo K-means y la técnica jerárquica SL [16].

Este algoritmo tiene dos etapas, en la primera se realizan N divisiones de la base de datos en subgrupos empleando el algoritmo K-means y en la segunda proponen un mecanismo de votación para combinar los resultados de los agrupamientos según una nueva medida de similaridad entre patrones bajo la idea de que si dos patrones pertenecen a un mismo grupo ellos serán colocados en el mismo grupo en diferentes agrupamientos. Toman la co-ocurrencia de todos los pares de patrones en el mismo grupo como votos para su asociación y detectan grupos consistentes en la matriz de asociación usando la técnica SL. Para cada patrón que no pertenezca a ningún grupo forma con ellos grupos unitarios.

3 Resultados Experimentales

En esta sección queremos mostrar los resultados de algunos de los experimentos realizados para comparar tres algoritmos: K-means, CURE y DBSCAN,

empleamos tres bases de datos artificiales (bases 1, 2 y 3) y cinco bases de datos reales (Heart, Diabetes, Phoneme, Satimage y Liver) tomadas del UCI Machina Learning Database Repository [18].

En la figura 1a, se muestra un buen resultado al aplicar el algoritmo CURE a la base de datos # 1, en la que el conjunto de datos está dividida en tres grupos, pero en el caso de la base # 2 no puede distinguir los 4 grupos según se ve en la figura 1b.

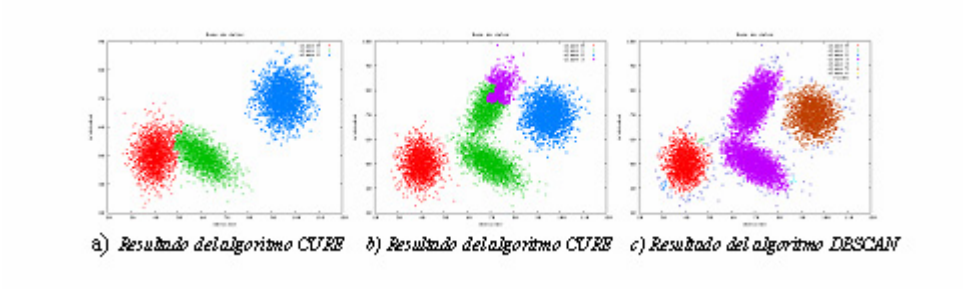

Figura1. Resultados de las bases de datos # 1 y # 2

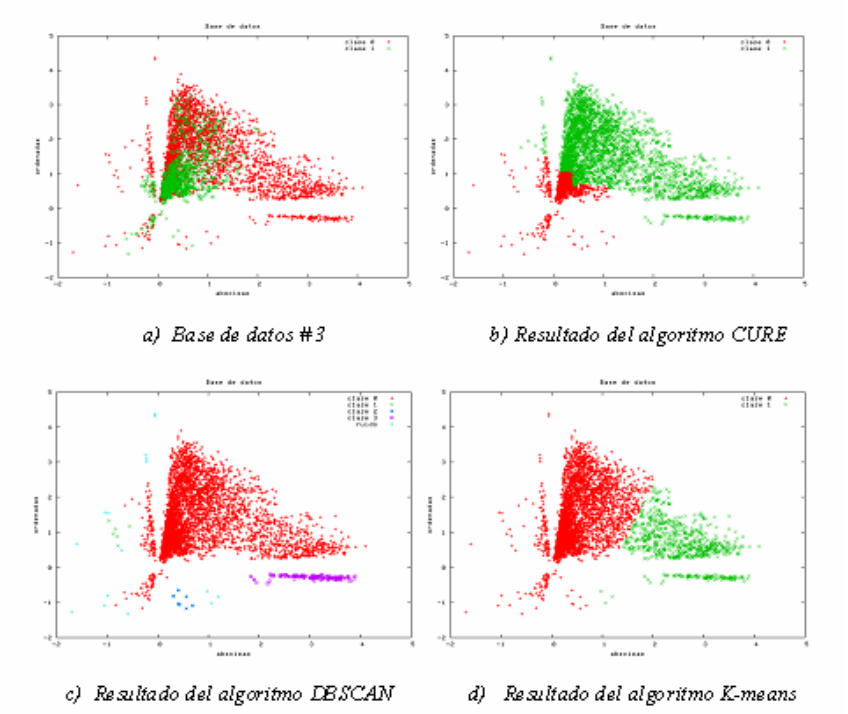

Figura2. Resultados de la base de datos # 3

El algoritmo DBSCAN no es capaz de separar los grupos que aparecen en color rojo y verde en la figura 1a, ni en el caso de la base # 2. Esto es debido al hecho de que DBSCAN reúne a todos los elementos dentro de la vecindad de un punto con dicho punto en un mismo grupo siempre que ese punto tenga al menos una cantidad prefijada de puntos de la base de datos en su vecindad, por lo que no puede separar grupos tan cercanos como los que se ven en las figuras 1a y 1b, K-means detecta bien todos los grupos para ambas bases de datos, este algoritmo detecta bien generalmente los clusters globulares, a diferencia de CURE por el hecho de la cantidad de puntos representativos a tener en cuenta. Los resultados de la base de datos # 3 se muestran en la figura 2.

El algoritmo que mejores resultados obtuvo fue CURE (figura 2b), este algoritmo es bastante efectivo para baja dimensionalidad, los algoritmos DBSCAN y K-means como se ve en las figuras 2c y 2d respectivamente no pueden distinguir los dos clusters por la mezcla tan grande que hay entre sus elementos.

Bases de datos reales: En la tabla siguiente mostramos los resultados de los experimentos realizados con bases de datos reales, se observa que el algoritmo que más alto índice de clasificación correcta presenta es K-means, seguido por CURE, en el caso de DBSCAN algunas veces no se obtiene el número de clases verdadero, como ocurre con las bases de datos Heart y Satimage.

	K-means	DBSCAN	CURE
Heart	79.62	-	62.96
Diabetes	66.79	57.87	65.62
Phoneme	70.65	70.46	70.65
Satimage	69.26	-	37.668
Liver	57.97	2.72	57.97

Tabla 1. Resultados de los experimentos realizados con bases de datos reales

4 Conclusiones y trabajo futuro

Con este trabajo hemos presentado los resultados del estudio realizado sobre algoritmos de agrupamiento para las diferentes técnicas existentes.

Mostramos al menos un algoritmo como ejemplo en cada una de las técnicas y hacemos hincapié en los algoritmos basados en densidad por ser de mayor interés para nosotros.

Implementamos tres de los algoritmos, dos de ellos (DBSCAN y K-means) muy citados en los trabajos sobre algoritmos de agrupamiento.

Realizamos experimentos con bases de datos artificiales y reales para hacer un estudio comparativo entre los algoritmos antes mencionados.

Como trabajo futuro pensamos crear un algoritmo basado en densidad con el objetivo de aplicarlo en el tratamiento de imágenes.

Referencias

[1] R. O. Duda, P. E. Hart, and D. Stork, "Pattern Classification", 2001, Wiley series in Probabilistic and Statistic, John Wiley and Sons, Second Edition.

[2] G. Karypis, E. H. Han. And V. Kumar, "Chameleon: A hierarchical clustering algorithm using dynamic modeling", IEEE Computer, 1999, 32(8), pag. 68-75.

[3] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data Clustering method for very large databases", In Proceedings of ACM SIGMOD Conference on Management of Data, 1996, pag. 103-114.

[4] S. Guha, R. Rastogi, and K. Shim**, "**CURE: An efficient clustering algorithm for large databases", In Proceedings of ACM SIGMOD´98, 1998, pag. 73-84.

[5] M. Ester, H.P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", In Proceedings of International Conference on Knoowledge Discovery and Data Mining (KDD´96), 1996, pag 226-231.

[6] M. Ankerst, M. Bruenig, H. P. Kriegel, and J. Sander, "OPTICS: ordering points tu identify the clustering structure", In Proceedings of ACM SIGMOD International Conference on Management of Data, 1999, pag. 49-60.

[7] Thanh N. Tran. "Knn Density-Based Clustering for High Dimensional Multispectral Images".

[8] L. Ertoz, M. Steinbach, V. Kumar, "Finding Clusters of Different Sizes, Shapes and Densities in Noise", 2003,

[9] A. Hinneburg, and D. A. Keim, "An efficient Approach to Clustering in Large Multimedia Databases with Noise", in Proceedings Knowledge Discovery and Data Mining, 1998, pag. 58-65.

[10] Mario A. Figueiredo and A. K. Jain, "Unsupervised Learning of Finite Mixture Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, Vol 24, Nro 3, pag 381-396.

[11] A. L.N. Fred y J. M. N. Leitao, "A New Cluster Isolation Criterion Based on Dissimilarity Increments", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, Vol 25. Nro 8, pag 944-958.

[12] Vázquez,F.;Sánchez, J.S.; Pla, F.;"A stochastic approach to Wilson's editing algorithm", Pattern Recognition and Image Analysis, Lecture Notes in Computer Science, Springer-Verlag, ISBN 3-540-26154-0, 2005, Vol. 3523, pag 35-42.

[13] R. A. Jarvis and E. Patrick, "Clustering using a similarity measure based on shared nearest neighbors", IEEE Transaction on Computers, 1973, C-22(11).

[14] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes", In Proceedings of IEEE Conference on Data Engineering, 1999.

[15] M. Dash, H. Liu y X. Xu, "1+1>2: Merging Distance and Density Based Clustering".

[16] A. L. N. Fred, and A.K. Jain, "Data Clustering Using Evidence Accumulation",

[17] Merz, C.J., Murphy., Murphy, P.M.: UCI Repository of Machine Learning Database. Department of Information and Computer Science, University of California, Irvine, CA (1998).