UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO

FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

"IMPLEMENTACIÓN DE ALGORITMOS GENÉTICOS PARA LA SEGMENTACIÓN DE IMÁGENES SATELITALES POR CONGLOMERADOS DE LA REGIÓN PUNO – 2013"

TESIS

PRESENTADA POR:

Bach. MELITON APAZA TITO

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO

PUNO – PERÚ

2014

UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

"IMPLEMENTACIÓN DE ALGORITMOS GENÉTICOS PARA LA SEGMENTACIÓN DE IMÁGENES SATELITALES POR CONGLOMERADOS DE LA REGIÓN PUNO - 2013"

TESIS

PRESENTADA POR:

Bach. MELITON APAZA TITO

A la Coordinación de Investigación de la Facultad de Ingeniería Estadística e Informática de la Universidad Nacional del Altiplano – Puno, para optar el Título Profesional de:

INGENIERO ESTADÍSTICO E INFORMÁTICO

APROBADA POR:

PRESIDENTE DEL JURADO : M.Sc. ALEJANDRO APAZA TARQUI

PRIMER MIEMBRO : M.Sc. REMO CHOQUEJAHUA ACERO

SEGUNDO MIEMBRO : M.Sc. ANGEL J. QUISPE CARITA

DIRECTOR DE TESIS : Dr. VLADIMIRO IBAÑEZ QUISPE

ASESOR DE TESIS : M.Sc. JHON R. HUANCA SUAQUITA

ÁREA: Informática
TEMA: Ingeniería del software

DEDICATORIA

La presente tesis la dedico a toda mi familia, principalmente a mis padres que ha sido un pilar fundamental en mi formación como profesional, por brindarme la confianza, consejos, oportunidad y recursos para lograrlo.

AGRADECIMIENTOS

A la UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO por darme la oportunidad de estudiar y proporcionarme los recursos necesarios para mi formación profesional.

Asimismo, estoy agradecido a mis padres, Teófilo y Victoria, que me enseñaron a entender el valor de la educación en la vida de las personas, algo por lo que siempre les estaré eternamente agradecido. A mi hermano Eduardo, mis hermanas: Lucila, Lourdes, a mis tíos Fortunato, Brígida y también le estoy agradecido a Roxana por su apoyo incondicional.

Finalmente a las personas que directa e indirectamente apoyaron a la concretización del presente trabajo de tesis.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo de investigación fue realizado en la Región Puno, cuyo objetivo es desarrollar una aplicación para la segmentación de imágenes basada en conglomerados, denominado Algoritmos Genéticos K-medias (AGKM).

Esta aplicación fue propuesta debido al deficiente método de selección del valor de inicialización del algoritmo K-medias al tomar un número de conglomerados inicial de forma aleatoria o por cálculo de la observación visual, esto puede influir en el desempeño del algoritmo, haciendo que tenga una separación inadecuada o demore más tiempo en la búsqueda del número de conglomerados. Se implementó usando la metodología de los Algoritmos Genéticos (AGs) y K-medias, el primero tiene la finalidad de encontrar un número de conglomerados existentes en la imagen y el segundo realiza el proceso de separación. La métrica usada para evaluar la eficiencia de este algoritmo es el valor de la entropía en las imágenes, los resultados obtenidos son sometidos a una prueba estadística que nos indica que existe una ligera mejoría.

Concluyendo que el AGKM ofrece una ligera mejoría con respecto al algoritmo K-medias tradicional en la segmentación de imágenes satelitales para la Región Puno.

Palabras Clave: Algoritmos Genéticos, K-medias, segmentación, kernel, gaussiana, procesamiento digital de imágenes, conglomerados, imágenes satelitales, función de densidad.

ABSTRACT

This research entitled "Implementation of Genetic Algorithm for the Segmentation of Satellite Imagery Cluster of Puno Region - 2013" was developed in the Puno Region, the target was developmed an application of image segmentation based on clusters, called Genetic Algorithms K-means (AGKM).

This application was proposed because the seleccition method of initialization value is limited to K-medias algorithm to take a number of initial cluster of random method or visual inspection, this influence in the performance of algorithm doing that algorithm to have inadequate separation or run slowly. It is implemented using the methodology of Genetic Algorithms and K-medias, the first aims to find an adequate number of segments which divide an image and provide this value to the second algorithm to finally make the segmentation process. The metric used to evaluate the efficiency of this algorithm is the value of the entropy in the images, the results are subjected to a statistical test indicates that there is a slight improvement .

Concluding that AGKM offers a slight improvement over the traditional K-means algorithm in the segmentation of satellite images for the Puno Region - 2013 .

Keywords: Genetic Algorithm, K-means, segmentation, kernel, gaussian, digital image processing, clustering, satellite images, density function.

INTRODUCCIÓN

El procesamiento digital de imágenes es un área que explora el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad, facilitar la búsqueda de información. Una de las técnicas es el algoritmo K-medias que está siendo utilizado en la segmentación de imágenes satelitales, esta implementación la podemos encontrar en el software de sistemas de información geográfica ArcGIS producido por ERSI.

El algoritmo K-medias es un método de división de información, utilizado para las diversas aplicaciones en las que se busca como resultado un número determinado de grupos (objetos semánticos, en el caso de segmentación de imágenes), el algoritmo fue propuesto por primera vez por Stuart Loyd en 1957 pero publicado en 1982 como una técnica para modulación por impulsos codificados (Lloyd, 1982).

La técnica de agrupamiento es no jerárquica, por lo que se fija un número k de conglomerados al inicio de su ejecución y se asignan elementos a un conglomerado en función de la distancia, empleando para ello funciones como la distancia euclidiana. Otra de sus características importantes es que emplea la media estadística para el cálculo de los nuevos conglomerados. Concretamente, lo que se calcula del conglomerado es su centroide, esto es el punto resultante de la media de todos los elementos asignados al conglomerado en cuestión.

En este contexto se observa que el algoritmo K-medias presenta una deficiencia en el método de inicialización, debido a que la asignación del número de conglomerados inicial es de forma aleatoria o por calculo visual. Por este

motivo es que se plantea como método de inicialización el valor encontrado por los Algoritmos Genéticos, esta solución explora un conjunto de datos sin tener previo conocimiento, teniendo como guía apenas una función de evaluación.

El concepto de análisis de conglomerados (“clusters”), que viene siendo utilizado con éxito en diversas áreas de investigación como mineración de datos, procesamiento digital de señales, análisis de datos meteorológicos, etc. Su principal objetivo es encontrar el número de conglomerados dentro de un conjunto de datos, generando de esta manera clases, en los que se puede reagrupar los datos.

Una técnica muy eficiente en el análisis de conglomerados es la utilización de la función de densidad de probabilidad estadística (Coto, 2003). Después de estimar la función de densidad de los datos es posible presentarlos gráficamente. Los picos de esta función representan el número de conglomerados (Pinto, 1998).

Los AGs son en esencia algoritmos de búsqueda basados en mecanismos de selección natural y genética. Sin las limitaciones encontradas en los métodos tradicionales, los AGs se muestran muy eficientes para buscar soluciones óptimas, o aproximadamente óptimas, en una gran variedad de problemas (Mendes Filho, 1998). Los AGs fueron utilizados con éxito en la predicción de número de conglomerados en un conjunto de datos unidimensionales (Pinto, 1998).

CAPÍTULO I

PLAN DE INVESTIGACIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

En el presente trabajo de tesis se aborda la técnica de segmentación de imágenes satelitales usando el algoritmo K-medias, este algoritmo presenta deficiencias en el método de inicialización al tomar un número de conglomerados inicial de forma aleatoria o por cálculo de la observación visual, esto puede influir en el desempeño del algoritmo, haciendo que demore más tiempo de lo debido en la búsqueda del número de conglomerados adecuados, o que no converja nunca. Es por este motivo que se propuso los Algoritmos Genéticos para la búsqueda del número inicial adecuado de conglomerados que nos permita segmentar la imagen de forma satisfactoria, para medir el desempeño de los algoritmos implementados se utilizó el indicador de entropía para cada imagen satelital antes y después del tratamiento con AGKM (Algoritmos Genéticos K-medias) y K-medias.

El algoritmo K-medias presenta una complejidad computacional difícil, existen heurísticos que hacen que converja rápidamente a un óptimo local. También hay que tener en cuenta que se requiere de un método de inicialización y se suelen usar dos principalmente:

- El método de Forgy, consistente en la toma de k observaciones de forma aleatoria como centros del clúster y luego empezar con el paso de asignación.

- El método de partición aleatoria, en el que se asigna aleatoriamente un clúster a cada observación y se sigue con el paso de actualización.

K-medias converge cuando al realizar el paso de asignación ningún elemento cambia de clúster. Este suele ser el algoritmo de parada más común, pero también se puede establecer un número máximo de iteraciones o cierto grado mínimo de convergencia, con lo cual no se conseguirá el óptimo local pero sí una aproximación.

Por lo que se plantea la siguiente pregunta ¿Los Algoritmos Genéticos K-medias (AGKM) propuesto en este trabajo es más eficiente en la segmentación de imágenes satelitales para la región Puno – 2013 frente al algoritmo K-medias tradicional?

1.2 JUSTIFICACIÓN DEL PROBLEMA

El análisis de imágenes comprende todos los métodos y técnicas que se utilizan para extraer información de una imagen. El primer paso para ello es la segmentación de imágenes que se ocupa de descomponer una imagen en sus partes constituyentes, como los objetos de interés y el fondo, basándose en ciertas características locales que nos permiten distinguir un objeto del fondo y objetos entre sí.

El resultado de la segmentación de una imagen es un conjunto de segmentos que cubren en conjunto a toda la imagen, o un conjunto de las curvas

de nivel extraídas de la imagen, cada uno de los píxeles de una región es similar en alguna medida, como el color, la intensidad o la textura. Regiones adyacentes son significativamente diferentes con respecto a las mismas características.

Los principales tipos de segmentación de imágenes son:

- Segmentación basada en características.
- Segmentación basada en transiciones.
- Segmentación basada en modelos.
- Segmentación basada en homogeneidad.

En este trabajo se trató la segmentación basada en características, útil cuando no se conoce las características de las regiones que buscamos o, incluso, si no se sabe cuántas categorías se tiene, cada región del espacio de características se define mediante un patrón de centroide, cada vector de características se asigna a la región del centroide más próximo. Este tipo también se le conoce como segmentación por conglomerados (Clustering). La exploración de una imagen de color requiere un tratamiento multidimensional debido a que se maneja una matriz tridimensional (tres canales) para nuestro caso, estas tres dimensiones son conocidas como RGB (Red, Green, Blue) que permiten generar una gama de colores, donde los valores varían de 0 a 255.

Actualmente no existe un método de segmentación que alcance resultados aceptables para todo tipo de imagen. No existen métodos que sean generales y que puedan ser aplicados a cualquier variedad de datos. Por lo tanto, la selección

de un método apropiado para un problema de segmentación puede ser muy difícil (Coto, 2003).

El análisis de conglomerados de datos ha sido usado con éxito en las más diversas áreas de investigación como mineración de datos, procesamiento de señales, biometría, con el objetivo de agrupar datos semejantes según sus características. Una técnica eficiente en el análisis de conglomerados es la utilización de la función de densidad de probabilidad (estimador kernel) que muestra el número de grupos gráficamente, pero esta técnica presenta algunos inconvenientes cuando es usado en altas dimensiones, este problema es conocido como la maldición de la dimensionalidad (curse of dimensionality) (Kriegel, 2005), (Hans-Peter Kriegel, March 2009).

Los algoritmos genéticos son usados para averiguar el número conglomerados que puede existir en un conjunto de datos, los métodos existentes para este análisis son los métodos estadísticos que necesitan de un número aproximado de conglomerados para localizarlos. El desempeño de estos métodos depende directamente del número de conglomerados. Este trabajo presenta los algoritmos genéticos con función de aptitud (función gaussiana) para descubrir el número de conglomerados en una imagen satelital.

Como alternativa de solución se plantea la metaheurística de los Algoritmos Genéticos para la búsqueda del número adecuado de conglomerados existentes en la imagen, para que este valor sea usado en la inicialización del algoritmo K-medias, a esta propuesta de algoritmo se la llamó Algoritmos Genéticos para K-medias (AGKM).

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar la aplicación de los Algoritmog Genéticos K-medias (AGKM) como técnica para la segmentación por conglomerados en imágenes satelitales de la región Puno - 2013.

1.3.2 OBJETIVOS ESPECÍFICOS

- Especificar la teoría de Algoritmos Genéticos K-medias (AGKM) como técnica de segmentación de imágenes satelitales por conglomerados.
- Aplicar la teoría de los Algoritmos Genéticos K-medias (AGKM) para segmentar imágenes satelitales de la región Puno.
- Medir el grado de segmentación alcanzado por el AGKM y K-medias a través de la entropía calculada para cada imagen satelital segmentada por cada algoritmo.

1.4 HIPÓTESIS

Los Algoritmos Genéticos K-medias (AGKM) obtienen un mejor desempeño en la segmentación de imágenes satelitales de la región Puno – 2013, en comparación con el algoritmo K-medias tradicional.

CAPÍTULO II

MARCO TEÓRICO

Pruebas realizadas en la comparación de Algoritmos Genéticos y el algoritmo de K-medias en algunas tareas, se observa que el primero supera en rendimiento, además es capaz de optimizar el número de conglomerados bien formados y definidos (Kudova, 2007), también estos AGs permitieron encontrar una topología de red neuronal artificial (número de conexiones de unidades en las capas intermediarias) que posibilitó la extracción de un conjunto de reglas que ofrecen una buena tasa de certeza (Ebecken, 2000), otro artículo donde se propone la aplicación de los AGs es en las redes ad hoc, que se caracterizan por no tener una infraestructura de comunicación. No existe una estructura jerárquica para el ruteamiento de los datos y transmisión de mensajes entre los nodos, un conglomerado agrupa dinámicamente un conjunto de nodos en torno de un nodo central, responsable por el ruteamiento de datos, llamado clusterhead.

El problema del particionamiento K-clustering es NP-completo, haciendo con que la búsqueda de una solución óptima para una topología genérica sea un desafío, una estrategia para la resolución de este problema es la aplicación técnicas basadas en meta-heurística, presentando una solución para el particionamiento k-clustering basado en AGs (Gadhoc, 2005).

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

La revisión bibliográfica se realizó tanto para las investigaciones en el tema de segmentación de imágenes como para el análisis de conglomerados para datos unidimensionales y multidimensionales y también el uso de ambas combinaciones en la segmentación de diferentes tipos de imágenes:

1. En la utilización de la técnica de segmentación en imágenes satelitales es posible encontrar la combinación de datos del Landsat TM[1] y ERS-1 SAR3[2] potencialmente podrían permitir mejorar la cobertura de mapeamiento, este artículo describe el potencial del uso de la segmentación de las imágenes del Landsat TM para describir los limites, el cual podría servir como base para el filtro de ruido del SAR speckle (Schoenmakers R. P H M, 1993).

2. Existen varios métodos (jerárquicos y no jerárquicos) para efectuar conglomerados de los datos, pero para todo los casos existe la necesidad de saber aproximadamente el número de conglomerados existentes en el conjunto de datos analizados. Después de este número de conglomerados es que los métodos comienzan a agrupar los que son semejantes. Una de las técnicas muy eficientes en el análisis de agrupamientos es la utilización de la función de densidad de los datos y

[1] Los LandSat son una serie de satélites construidos y puestos en órbita por EE.UU. para la observación en alta resolución de la superficie terrestre, el TM (mapa temático) es uno de los sensores introducidos en el programa landsat que consigue capturar la imagen en 7 bandas de imágenes (tres ondas visibles y 4 infrarojas).

[2] El European Remote Sensing Satellite (ERS) se convirtió en el primer satélite de observación de la Tierra lanzado por la Agencia Espacial Europea (ESA), al ser lanzado el 17 de julio de 1991, por un Ariane 4 desde Kourou (Guyana Francesa), a una órbita polar síncrona con el sol a una altura de entre 782 y 785 km.

es posible presentarlos gráficamente. Los picos de esta función representan el número de conglomerados (Pinto, 1998).

3. El desarrollo de algoritmos de segmentación es un área en el que se lleva empleando mucho esfuerzo, dentro de estos trabajos se puede encontrar el artículo que presenta una propuesta de una base de datos de imágenes segmentadas por el ser humano y aplicaciones para evaluar el algoritmo de segmentación (Martin D., July 2001).

4. Los algoritmos genéticos son algoritmos de búsqueda basados en mecanismos de selección natural y genética, sin las limitaciones encontradas en los métodos tradicionales, estos algoritmos se muestran muy eficientes en la búsqueda de soluciones óptimas (Kudova, 2007).

5. Por otro lado, el trabajo con una técnica similar nos dice que, los algoritmos genéticos son altamente efectivos al segmentar imágenes médicas, el principal desafío de los algoritmos de segmentación en imágenes médicas es el poco contraste y las imágenes difusas al momento de identificar órganos, no obstante los algoritmos genéticos también proveen una flexibilidad en el procedimiento de segmentación de imágenes (Maulik., 2009).

6. Una de las investigaciones que más se asemeja al trabajo presentado, un nuevo contexto para mejorar la calidad de los conglomerados de K-medias usando algoritmos genéticos (GA), y mide la calidad de los conglomerados por medio de la entropía (Prabha, 2011).

7. Finalmente uno de los artículos revisados titulado segmentación de imágenes satelitales usando lógica fuzzy y transformada de Hilbert Huang, propone la lógica fuzzy para la extracción de características de la imagen satelital como número de conglomerados para posteriormente usar K-medias, que es parte de la transformada de Hilbert Huang (Purushothaman S., 2012).

2.2 BASE TEÓRICA

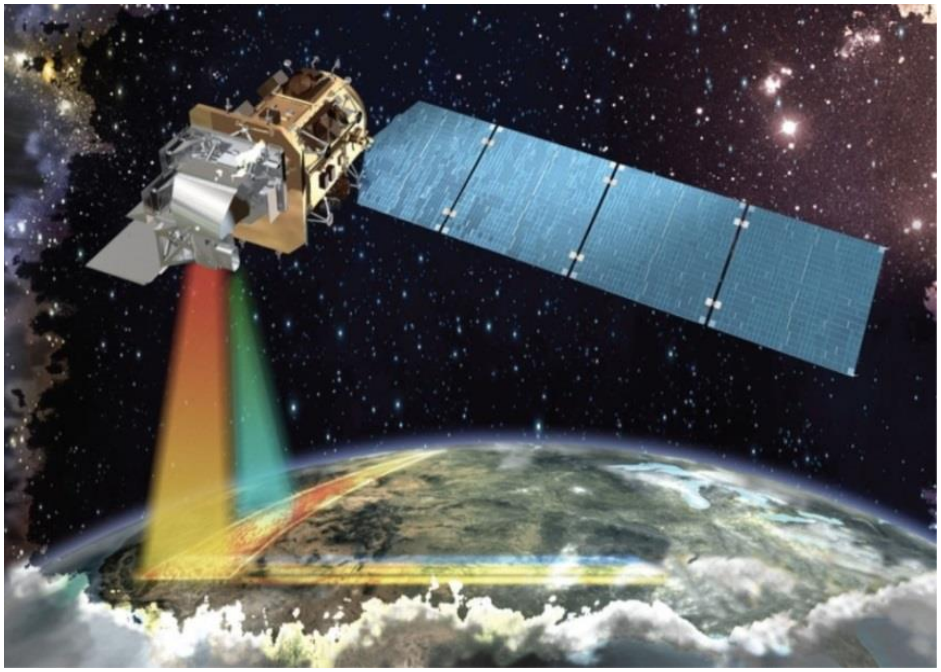
En esta sección se presenta las definiciones necesarias para comprender el proceso de la segmentación de imágenes satelitales, como saber qué tipo de imágenes se están tratando y de donde provienen los algoritmos utilizados.

2.2.1 LANDSAT 7

Las imágenes utilizadas en este trabajo fueron descargadas del INPE (Instituto Nacional de Investigaciones Espaciales) – Brasil de libre acceso, estas imágenes son procedentes de la tecnología Landsat 7, este satélite fue lanzado el 15 de abril de 1999, es el séptimo satélite del programa Landsat. El objetivo principal de Landsat 7 es actualizar el archivo mundial de fotos por satélite, que proporciona hasta la fecha imágenes libres de nubes. El programa Landsat es administrado y operado por el USGS (United States Geological Survey), y los datos de Landsat 7 se recogen y se distribuyen por el USGS. El proyecto de la NASA “World Wind” permite que las imágenes 3D a partir de imágenes Landsat 7 y otras fuentes puedan ser navegados y vistos desde cualquier ángulo libremente. El compañero del satélite Earth Observing-1, genera senderos por un minuto y sigue las mismas características orbitales. Landsat 7 fue construido por Lockheed Martin Space Systems Company.

Landsat 7 fue diseñado para una duración de cinco años, y tiene la capacidad de recoger y transmitir hasta 532 imágenes al día. Se encuentra en una órbita polar, sincronizada con el sol, lo que significa que escanea toda la superficie de toda la tierra como muestra la Figura 1. Con una altitud de 705 kilómetros, se tarda 232 órbitas, o 16 días, para hacerlo. El satélite pesa 1.973 kg, es de 4,04 m. de largo, y 2,74 m. de diámetro. A diferencia de sus predecesores, Landsat 7 tiene una memoria de estado sólido de 378 gigabits. El principal instrumento a bordo del Landsat 7 es el Enhanced Thematic Mapper Plus (ETM+) (Landsat, 2003).

FIGURA 1: Satelite Landsat 7 en órbita



FUENTE: http://www.nasa.gov/

PRINCIPALES CARÁCTERÍSTICAS

- Una banda pancromática con 15 m. de resolución espacial.
- Bandas visibles en el espectro del azul, verde, rojo, infrarrojo cercano, y en el infrarrojo medio con 30 m. de resolución espacial.

- Un canal infrarrojo térmico con resolución espacial de 60 m.
- Apertura completa, el 5% de calibración radiométrica absoluta.
- Las longitudes de onda por banda nos representa diferentes tipos de superficie como se describe en la Tabla 1.

TABLA 1: Bandas espectrales del sensor TM para Landsat7

Banda	Longitud de Onda (μm)	Características
1	0.45 a 0.52	Azul-verde. Máxima penetración en el agua (Útil para hacer cartografía batimétrica en aguas poco profundas). Útil para distinguir suelo de vegetación y coníferas de árboles de hoja caduca
2	0.52 a 0.6	Verde, ajustado al pico de reflectancia de vegetación en el verde, útil para evaluar el vigor de las plantas
3	0.63 a 0.69	Rojo. Coincide con una banda de absorción de la clorofila, importante para la discriminación de vegetación
4	0.76 a 0.9	IR reflejado. Útil para determinar contenido de biomasa y para cartografía de costas
5	1.55 a 1.75	IR reflejado. Indica contenido de humedad de suelo y vegetación. Penetra en nubes finas. Da buen contraste entre tipos de vegetación.
6	10.4 a 12.5	IR térmico. Las imágenes nocturnas son útiles para cartografía térmica y estimación de la humedad de suelos
7	2.08 a 2.35	IR reflejado. Coincide con la banda de absorción causada por iones hidroxilo en minerales. El cociente de las bandas 5 y 7 se usa para destacar rocas alteradas hidrotérmicamente, asociadas con depósitos minerales.

FUENTE: http://www.nasa.gov/

El conjunto de imágenes satelitales descargadas del servidor están compuestas de 7 imágenes las que pueden ser recombinadas como falso RGB (3 bandas), para nuestro caso se utilizó la combinación en falso color 432, la interpretación de colores para esta combinación es (Llorente, 2010):

Rojo - magenta, vegetación vigorosa, cultivos regados, prados de montaña o bosques caducifolias en imágenes de verano y cultivos herbáceos de secano en imágenes de primavera.

Rosa, áreas vegetales menos densas y/o vegetación en temprano estado de crecimiento. Las áreas residenciales suburbanas en torno a las grandes ciudades, con sus pequeños jardines y arboles diseminados, aparecen también en este color las praderas.

Blanco, áreas de escasa o nula vegetación pero de máxima reflectividad, nubes arenas, depósitos salino, canteras y suelos desnudos.

Azul oscuro o negro, superficies cubiertas total o parcialmente por el agua, ríos canales lagos y embalses, en zonas volcánicas los tonos negros pueden asimismo identificar flujos de lava.

Gris azul metálico, ciudades y áreas pobladas, si bien puede tratarse de roquedo desnudo.

Marrón, vegetación arbustiva muy variable en función del tono del sustrato, los tonos más oscuros indican la presencia de materiales paleozoicos (pizarras), mientras los materiales calcícolas, menos densos normalmente, ofrecen una coloración más clara.

Beige – dorado, identifica zonas de transición, prados secos frecuentemente asociados con el matorral ralo.

2.2.2 SEGMENTACIÓN DE IMÁGENES

La segmentación de imágenes satelitales es parte fundamental del sensoramiento remoto, este comenzó hace unos treinta años, utilizando datos e imágenes aéreas digitalizadas usando scaners multiespectrales (Kiefer, 2000) (Schowengerdt, 1997). En la actualidad, una PC estándar con suficiente velocidad de procesamiento, cantidad de memoria y capacidad de disco puede procesar fácilmente imágenes satelitales de muchos de los satélites orientados al estudio civil de recursos de tierra (Claudio Delrieux, 2001).

En este proyecto se busca implementar el algoritmo AGKM como un método de apoyo en la inicialización del algoritmo K-medias, la segmentación de imágenes consiste en dividir la imagen en estructuras con significado, por ejemplo los objetos contenidos en una imagen, y de asociar cada píxel de la imagen como perteneciente a un sólo objeto de la imagen (Coto, 2003).

Después de realizada una segmentación, se conocen las regiones y las discontinuidades entre regiones. Luego esas regiones son empleadas para extraer información relevante sobre los objetos contenidos en la imagen.

Existen diferentes enfoques para segmentar como los métodos probabilísticos, variaciones y de minimización de energía, los enfoques probabilísticos consideran asignar una distribución de probabilidad a los segmentos para cada píxel, aunque estas hacen más complejo el proceso de

segmentar y la posterior implementación. Considerando enfoques clásicos de segmentación, la clasificación seria de la siguiente forma:

SEGMENTACIÓN BASADA EN EL ANÁLISIS DE HISTOGRAMA

Los métodos basados en el histograma son muy eficientes en comparación con otros métodos de segmentación de la imagen, ya que normalmente requieren sólo una pasada por los pixeles. En esta técnica, un histograma se calcula a partir de todos los píxeles de la imagen, y los picos y valles en el histograma se utilizan para localizar los grupos en la imagen (el color o la intensidad pueden ser usados como medida).

Un refinamiento de esta técnica consiste en aplicar de forma recursiva el método de búsqueda de histograma a los clusters de la imagen con el fin de dividirlos en grupos más pequeños, esto se repite con las agrupaciones cada vez más pequeñas hasta que no se puedan formar más agrupaciones, una desventaja del método de búsqueda de histograma es que puede ser difícil de identificar los picos y valles importantes en la imagen (Tung, 1995).

Esta técnica está basada en un concepto muy simple: Se elige un parámetro "a" denominado umbral de brillo (intensidad) y entonces:

$$Si\ a[m,n] \geq \alpha \qquad entonces\ b[m,n] = 1$$

$$en\ caso\ \ contrario \qquad\qquad b[m,n] = 0$$

Suponiendo "a" la imagen original y "b" la imagen resultante o imagen segmentada. Evidentemente el umbral no es único para todas las imágenes, depende del dominio y de los objetos que se quieran detectar. Existen diferentes

procesos o métodos que se utilizan para obtener el umbral adecuado para una correcta segmentación (Tung, 1995):

P-Tile Method, Este método utiliza conocimiento acerca del área del histograma que ocupan los objetos que se quieren detectar, suponiendo que para una aplicación dada, los objetos ocupan sobre un porcentaje del área de la imagen. Utilizando el conocimiento de esta partición (en la imagen), uno o más umbrales pueden ser elegidos asignando un porcentaje de pixeles a los objetos, evidentemente, este método tiene un uso muy limitado. Solamente unas pocas aplicaciones permiten estimar el área de forma general (Taghizadeh, 2011).

Isodata Algorithm, Es una técnica iterativa que se utiliza para la obtención del umbral correcto. El histograma es inicialmente segmentado en dos partes utilizando un umbral de comienzo tal como la mitad del máximo valor del rango dinámico (Merzougui M, 2013).

A continuación se computa la media de los valores asociados con cada una de las partes en que ha quedado segmentado el histograma m1, m2. Utilizando esos valores se calcula un nuevo valor umbral mediante la fórmula:

$$\alpha = (m1 + m2)/2 \quad \textbf{(1)}$$

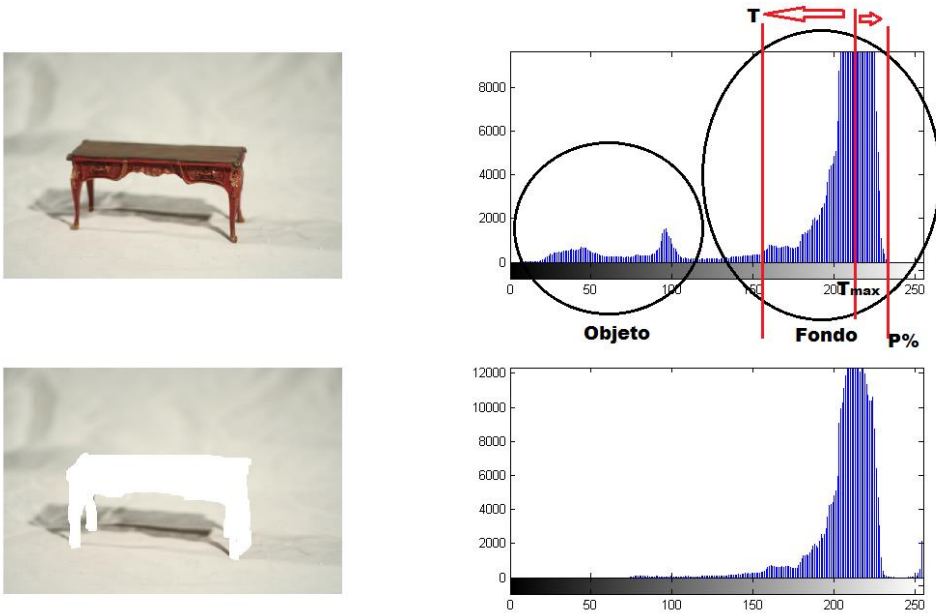
El proceso continua hasta que en dos pasos consecutivos el valor umbral calculado no cambia.

Background-simmetry algorithm, Esta técnica asume que hay un pico simétrico y dominante en el histograma, el pico puede ser del objeto o del fondo, aunque es mucho más común que sea del fondo de la imagen, dando nombre al algoritmo mostrado en la Figura 2 (Young I.T., 2005):

- Inicialmente se suaviza el histograma.
- Se obtiene el máximo global del histograma Tmax (fondo).
- Se busca a la derecha (en el lado opuesto al objeto) el punto de intensidad que corresponde al p% del histograma (por ejemplo el 95 % de puntos del histograma).
- Puesto que se supone que el lóbulo del fondo es simétrico, se toma como umbral T máximo menos un desplazamiento igual al del punto del p%.

$$T = Tmax - (P\% - Tmax) \quad \textbf{(2)}$$

FIGURA 2: Histograma del objeto usando el algoritmo background-simmetry



FUENTE: Elaboración propia

Por ejemplo en una imagen, los puntos asociados a los objetos a localizar se encuentran a la izquierda del pico del background (nivel de intensidad:183). Si tomásemos como porcentaje el 95%, el valor de luminosidad se encuentra a

la derecha del pico. Para el valor de brillo (216), el 5% de los puntos se encuentran a la derecha del pico. Como suponemos que existe una simetría entorno a dicho pico se utiliza como umbral un desplazamiento a la izquierda del máximo igual al desplazamiento que existe desde el máximo hasta la localización del porcentaje marcado (p%).

Para este caso: 183 - (216-183) = 150 (umbral utilizado).

Se ha supuesto que los objetos son oscuros sobre un fondo claro, pero también se pueden suponer objetos claros sobre fondo oscuro.

Una variación de la técnica anterior consiste en utilizar la desviación típica de la parte del fondo (a la derecha del máximo) y utilizar ese valor como desplazamiento, en lugar de p%. Tomando como umbral T = Tmax - 2,57 σ si el pico del histograma correspondiera a una distribución Gaussiana, el valor de 1.96, significaría que por encima queda un 5% de los pixeles. Para el coeficiente 2.57 quedaría un 1% de los pixeles.

Triangle algorithm, Esta técnica se basa en la detección del umbral correcto, es muy eficiente cuando los pixeles de los objetos producen un pico suave en el histograma (Young I.T., 2005):

- Se traza una línea entre el valor máximo del histograma (al nivel de gris bmax) y el mínimo gris en la imagen bmin = (p=0%).
- Se calcula la distancia desde dicha línea al histograma, para todos los valores de b, es decir desde bmin…a bmax.

- Finalmente, se escoge como umbral el valor de gris b0, debido a que la distancia entre el histograma y la línea hallada, anteriormente, es máxima.

Limitaciones de los métodos basados en histogramas, Son válidos si los objetos tienen valores de intensidad constantes sobre toda la imagen pero, si la iluminación no es uniforme sobre toda la escena, puede suceder que un único umbral no sea suficiente para poder segmentar la imagen.

Otra de las limitaciones de estos métodos consiste en que el histograma nos da información de la distribución global de la intensidad de una imagen. Imágenes muy diferentes pueden tener diferentes distribuciones espaciales de niveles de gris, pero tener histogramas muy similares.

SEGMENTACIÓN BASADA EN CRECIMIENTO DE REGIONES

Esta técnica segmenta una imagen partiendo desde el centro de un objeto y creciendo hacia el exterior del mismo hasta encontrar los bordes que lo limitan, este proceso es repetitivo para cada objeto dentro de la imagen (Macq, 2005). La técnicas basadas en píxeles aseguran la homogeneidad de las regiones pero no garantizan que sean conexas. En la segmentación de imágenes se debe considerar también la información suministrada por los píxeles del entorno para conseguir así regiones homogéneas y conexas, pues los píxeles de un mismo entorno suelen tener propiedades estadísticas similares y pertenecer a una misma región (Macq, 2005).

Una segmentación basada en regiones de una imagen digital "I" consiste en realizar una partición de la imagen en k regiones con las siguientes propiedades:

- Las regiones obtenidas en la partición, R_1, R_2, …, R_k, deben ser disjuntas.

$$R_i \cap R_j = \emptyset, \qquad i \neq j \tag{3}$$

- Su unión debe ser la imagen completa.

$$I = \bigcup_{i=1}^{k} R_i \tag{4}$$

- Cada región R_i tiene que ser conexa, es decir, todos sus píxeles conectados

- Se debe verificar que: $P(R_i) = VERDADERO\ y\ P(R_i \cup R_j) = FALSO$.

Para regiones adyacentes cualesquiera, R_i y R_j, siendo P el predicado que nos proporciona el test de homogeneidad de la región.

El crecimiento de las regiones es un procedimiento que consiste en ir formando grupos (regiones) de píxeles por incorporación sucesiva de píxeles de la imagen a los grupos según algún criterio predefinido. El procedimiento más sencillo de crecimiento de regiones comienza con un conjunto de píxeles, llamados semillas, que representan las distintas regiones de la imagen (al menos una semilla por región).

A partir de las semillas se van incorporando nuevos píxeles a las regiones utilizando un mecanismo de crecimiento que detecta en cada etapa k y para cada región $R_i(k)$, los píxeles aun no clasificados que pertenecen a un entorno predefinido de algún píxel del contorno de la región R_i(k). Para cada píxel (x,y),

detectado de esta forma, se comprueba si cumple la regla de homogeneidad, es decir, si la nueva región $R_i(k) \cup R_j(x,y)$ sigue siendo homogénea, en cuyo caso se consigue ampliar la región añadiéndole dicho píxel. El píxel se elige del entorno de un píxel del contorno de la región para garantizar de alguna manera la conectividad de la región (Macq, 2005).

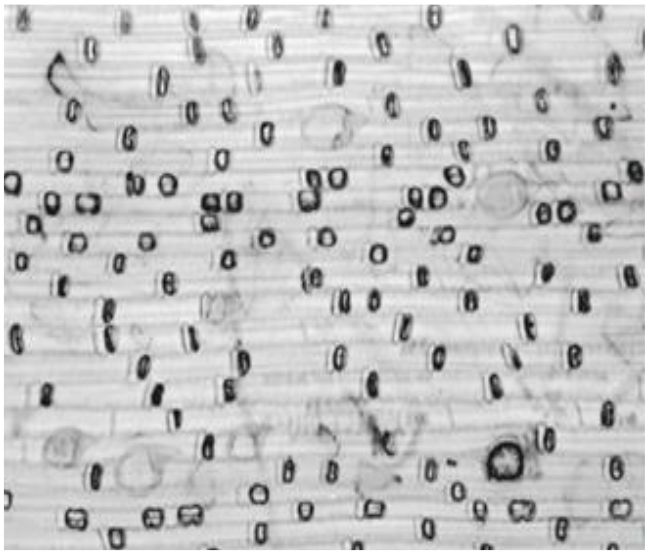
SEGMENTACIÓN BASADA EN BORDES (DISCONTINUIDAD)

Este método permite encontrar los bordes en una imagen, los cuales en realidad determinan los límites de cada segmento en la imagen y así identificar un objeto (Wang, 1999).

Un problema muy importante en la segmentación de los objetos es el problema de *Shading* o iluminación no uniforme como se presenta en la Figura 3, a pesar de la iluminación perfecta, si tenemos 2 clases de objetos: unos obscuros y otros claros, al momento de utilizar un umbral, los objetos claros tienden a engrandecerse y los objetos oscuros tienden a empequeñecerse. Este problema puede solucionarse si se utiliza un valor umbral para cada objeto, lo cual en la práctica es difícil de determinar.

La segmentación basada en bordes está fundamentada en el hecho de que la posición de un borde es dada por un máximo en las derivadas de primer orden o un cruce por cero en las derivadas de segundo orden. Por lo tanto, lo que se debe de hacer es buscar los máximos locales en la intensidad del borde. Se puede probar que esta estrategia no está influenciada por las variaciones en la iluminación (Wang, 1999).

FIGURA 3: Imagen con iluminación no uniforme



FUENTE: Elaboración propia

La estrategia típica es:

- Realizar un barrido de la imagen línea por línea en busca de un máximo local en la magnitud del gradiente.
- Cuando un máximo es encontrado, se ejecuta un algoritmo de rastreo (Tracing algorithm) el cual intenta seguir el máximo del gradiente alrededor del objeto hasta encontrar nuevamente el punto de inicio.
- Se busca nuevamente un punto de inicio.

El algoritmo más utilizado es el Canny, es un operador desarrollado por John F. Canny en 1986 que utiliza un algoritmo de múltiples etapas para detectar una amplia gama de bordes en imágenes. Lo más importante es que Canny también desarrolló una teoría computacional acerca de la detección de bordes que explica por qué la técnica funciona (Canny, 1986).

El propósito de Canny fue descubrir el algoritmo óptimo de detección de bordes. Para que un detector de bordes pueda ser considerado óptimo debe cumplir los siguientes puntos:

- Buena detección - el algoritmo debe marcar el mayor número real en los bordes de la imagen como sea posible.
- Buena localización - los bordes de marca deben estar lo más cerca posible del borde de la imagen real.
- Respuesta mínima - El borde de una imagen sólo debe ser marcado una vez, y siempre que sea posible, el ruido de la imagen no debe crear falsos bordes.

Para satisfacer estos requisitos Canny utiliza el cálculo de variaciones una técnica que encuentra la función que optimiza una función indicada. La función óptima en el algoritmo de Canny es descrito por la suma de cuatro términos exponenciales, pero se puede aproximar por la primera derivada de una gaussiana.

Etapas del algoritmo de Canny

Reducción de ruido, en esta etapa el algoritmo de detección de bordes de Canny utiliza un filtro basado en la primera derivada de una gaussiana. Debido a que es susceptible al ruido presente en datos de imagen sin procesar, la imagen original es transformada con un filtro gaussiano. El resultado es una imagen un poco borrosa respecto a la versión original. Esta nueva imagen no se ve afectada por un píxel único de ruido en un grado significativo.

Encontrar la intensidad del gradiente de la imagen, el borde de una imagen puede apuntar en diferentes direcciones, por lo que el algoritmo de Canny utiliza cuatro filtros para detectar horizontal, vertical y diagonal en los bordes de la imagen borrosa. El operador de detección de bordes (Roberts, Prewitt, Sobel, por ejemplo) devuelve un valor para la primera derivada en la dirección horizontal (G_y) y la dirección vertical (G_x) . A partir de éste, se pueden determinar el gradiente de borde y la dirección (Canny, 1986).

$$G = \sqrt{G_x^2 + G_y^2} \quad \textbf{(5)}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad \textbf{(6)}$$

SEGMENTACIÓN BASADO EN UMBRALES

Se caracterizan por trabajar con umbrales para segmentar la imagen, los umbrales actúan como separadores que permitirán decidir qué conjunto de tonos de gris pertenece a una determinada región. Estas técnicas son aplicadas sobre una imagen completa, y también pueden combinarse con otras durante el pre-procesamiento o post-procesamiento de la imagen, de manera que se obtengan mejores resultados (Shapiro, 2002).

SEGMENTACIÓN BASADO EN MATCHING

El algoritmo de segmentación basado en matching trata de identificar determinados objetos en una imagen, entonces a partir de este conocimiento es posible ubicarlos en la imagen (Yamaoka, 2006).

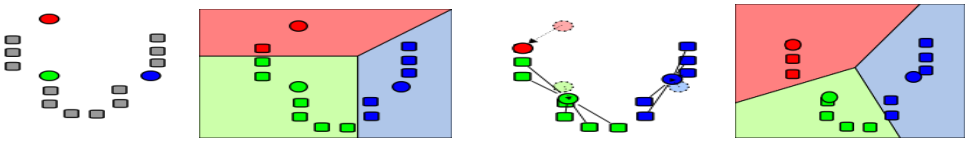
SEGMENTACIÓN POR CONGLOMERADOS USANDO K-MEDIAS

Como su nombre lo indica estas técnicas tratan de agrupar un conjunto de píxeles que son similares bajo algún criterio (Keren, 2004), el análisis de conglomerados es utilizado en la mineracion de datos no supervisada, teniendo como entrada un conjunto de datos y obteniéndose como salida un número de agrupaciones (Dash, 2012).

Útil cuando no se conocen las características de las regiones que buscamos o, incluso, si no se sabe cuántas categorías existen, el algoritmo de K-medias es una técnica iterativa que se utiliza para dividir una imagen en K conglomerados (Hartigan & Wong, 1979), El procedimiento básico de K-medias se describe En la Figura 4:

- Escoger K centros de conglomerados, ya sea de forma aleatoria o basándose en algún método heurístico.
- Asignar a cada píxel de la imagen el clúster que minimiza la varianza entre el pixel y el centro de conglomerados.
- Recalcular los centros de los conglomerados haciendo la media de todos los pixeles del conglomerados.
- Repetir los pasos 2 y 3 hasta que se consigue la convergencia (por ejemplo, los pixeles no cambian de conglomerados).

FIGURA 4: Interpretación gráfica del algoritmo estándar K-medias



(a) Se selecionan k puntos de la imagen, 3 en este caso (los mostrados en color).

(b) Se crean k clusters a partir de los puntos anteriores.

(c) El centro de gravedad de cada uno de los grupos k se convierte en el nuevo medio.

(d) Los pasos 2 y 3 se repiten hasta que se alcance la convergencia.

FUENTE: http://es.wikipedia.org/wiki/K-means

En este caso, la varianza es la diferencia absoluta entre un píxel y el centro del conglomerado. La diferencia se basa típicamente en color, la intensidad, la textura, y la localización del pixel, o una combinación ponderada de estos factores. El número K se puede seleccionar manualmente, aleatoriamente, o por una heurística. Este algoritmo garantiza la convergencia, pero puede devolver una solución que no sea óptima. La calidad de la solución depende de la serie inicial de conglomerados y del valor de K.

En estadística y aprendizaje automático, el algoritmo de las K-medias es un algoritmo de agrupamiento para dividir objetos en K grupos, donde K<n. Es similar al algoritmo de maximización de expectativas para las mezclas de gaussianas ya que ambos pretenden encontrar los centros de agrupaciones naturales de los datos. El modelo requiere que los atributos del objeto correspondan a los elementos de un espacio vectorial. El objetivo es intentar alcanzar la mínima varianza total entre conglomerados, o minimizar la función de error cuadrático.

El algoritmo de las K-medias fue propuesto en 1956. La forma más común del algoritmo usa una heurística de refinamiento conocido como el algoritmo de Lloyd. El algoritmo de Lloyd comienza dividiendo los puntos de entrada en K

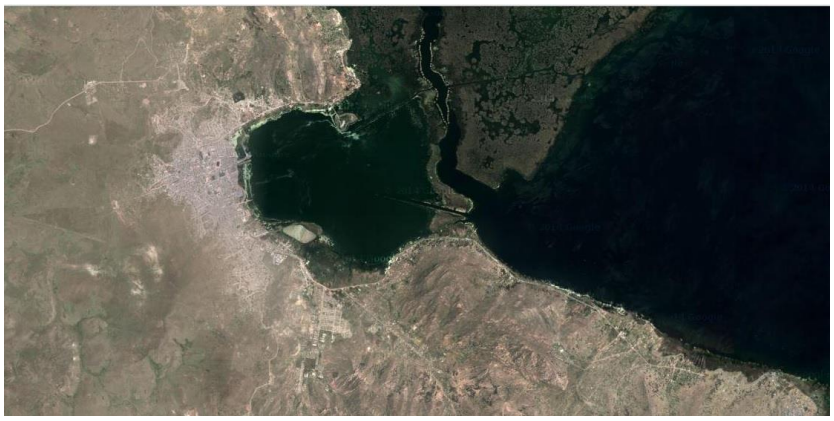
conjuntos iníciales, ya sea al azar o usando algunos datos heurísticos y a continuación, calcula el punto medio o centro de gravedad de cada conjunto. Se construye una nueva partición, asociando cada punto con el centro de gravedad más cercano. Luego se recalculan los baricentros que es para los nuevos conglomerados, y el algoritmo se repite alternando la aplicación de estos dos pasos hasta que converja, que se obtiene cuando los puntos ya no cambiar de conglomerados (o los centros de gravedad ya no se modifican). Los algoritmos de Lloyd y de las K-medias a menudo se utilizan como sinónimos, pero en realidad el algoritmo de Lloyd es una heurística para resolver el problema de las K-medias, como ocurre con ciertas combinaciones de puntos de partida y baricentros, el algoritmo de Lloyd puede converger a una solución incorrecta. Existen otras variantes, pero el algoritmo de Lloyd es el más popular, porque converge muy rápidamente. En cuanto al rendimiento, el algoritmo no garantiza que se devuelva un óptimo global.

La calidad de la solución final depende en gran medida del conjunto inicial de conglomerados, y puede, en la práctica, ser mucho más pobre que el óptimo global. Dado que el algoritmo es extremadamente rápido, es un método común ejecutar el algoritmo varias veces y devolver las mejores agrupaciones obtenidas. Un inconveniente del algoritmo de las K-medias es que el número de conglomerados K es un parámetro de entrada. Una elección inadecuada de K puede dar malos resultados. El algoritmo también asume que la varianza es una medida adecuada de la dispersión del conglomerados (Hartigan & Wong, 1979).

La Figura 5 muestra la imagen satelital de la ciudad de Puno con un acercamiento de 3125 Km. de altura. Esta imagen, al ser tratada con el algoritmo de K-medias queda transformada en 7 segmentos como muestra la Figura 6,

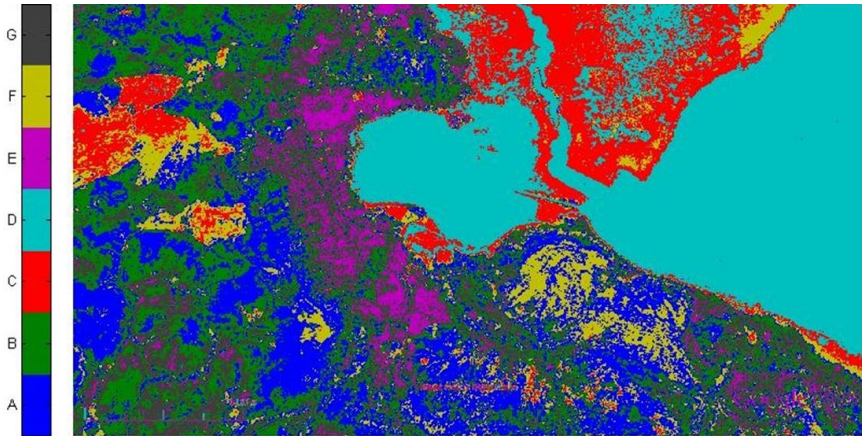
como ejemplo esta figura nos permite distinguir diferentes tipos de terrenos como regiones pobladas que es representada con la letra E, el lago está en el segmen

FIGURA 5: Fotografía Satelital de la ciudad de Puno



FUENTE: INPE

FIGURA 6: Fotografía segmentada en conglomerados (A, B, C, D, E, F, G)



FUENTE: Elaboración propia

2.2.3 ANÁLISIS DEL HISTOGRAMA

El histograma es el estimador de densidad más sencillo y mejor conocido de los estimadores no paramétricos. Algunos autores distinguen la utilización del histograma como técnica de representación de datos o como estimador de

densidad como es nuestro caso, la diferencia básica es que en este último caso debe estar normalizado para que cuando se integre nos resulte igual a 1.

La distribución de densidad de probabilidad (frecuencia relativa) es constituida a partir de las frecuencias absolutas dividida entre el número total del experimento, y h, es el ancho de las barras distribuidas a lo largo del intervalo donde se encuentran los datos. Este histograma es definido por la siguiente ecuación:

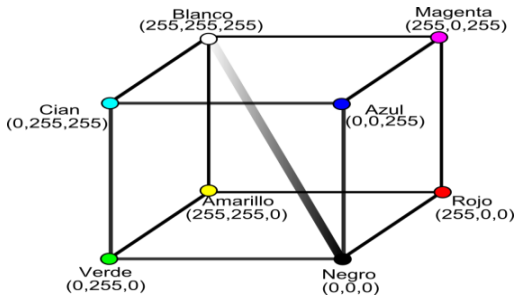
$$\hat{f} = \frac{1}{nh}(\,número\ de\ Xi\ en\ la\ misma\ barra\ que\ x) \qquad \textbf{(7)}$$

Dónde:

- n es el total del experimento.
- h ancho del intervalo.

Para presentar el siguiente ejemplo es necesario definir que el color y brillo de las imágenes que son mapeadas según el cubo de la Figura 7, esto permite generar una enorme cantidad de colores, así como intensidad que se representa en la diagonal, recorriendo desde el color negro representado por (0, 0, 0) hasta la tonalidad más clara o blanca (255, 255, 255), esta diagonal es la que usaremos para el ejemplo que sigue.

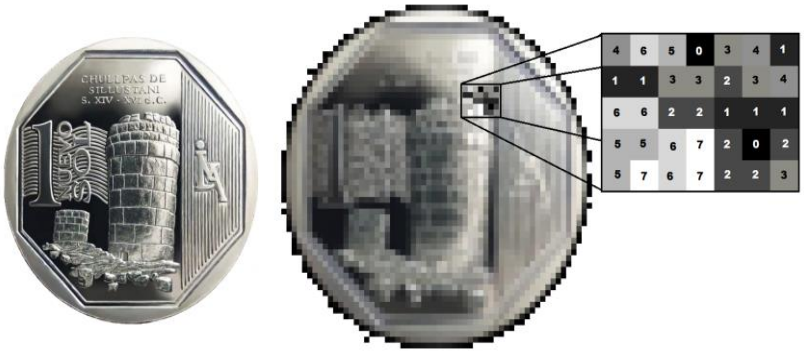
FIGURA 7: Mapa de colores e intensidad de brillo



FUENTE: Elaboración propia

Dada la Figura 8, que muestra la imagen de una moneda representada en una escala de grises con 8 tonalidades diferentes, que serían representados en la diagonal principal. Esta imagen es ampliada de manera que los pixeles individuales sean visibles, el pequeño rectángulo negro nos muestra una sección de la imagen ampliada que representaremos en histogramas.

FIGURA 8: Pixeles etiquetados según el brillo (0-7, 8 tonos de gris)



FUENTE: Elaboración propia

Para nuestro experimento en cuestión se tiene que el rectángulo tiene $n = 35$ píxeles y como se tiene 8 diferentes tonos de gris, del negro al blanco, estas tonalidades son etiquetadas con los valores 0, 1, 2, 3, 4, 5, 6, 7. El número (0) representará el negro y el (7) representa el color blanco, con esta información planteamos la siguiente pregunta: ¿Cuántos píxeles de cada tono tenemos?.

Para responder a esta pregunta solo necesitamos elaborar nuestra tabla de distribución de frecuencias realizando una inspección vertical u horizontal de la matriz de pixeles del rectángulo mostrado anteriormente, el resultado se muestra en la Tabla 2, calculando sus respectivas frecuencias para el análisis correspondiente.

TABLA 2: Niveles de brillo de la moneda, con sus respectivas frecuencias

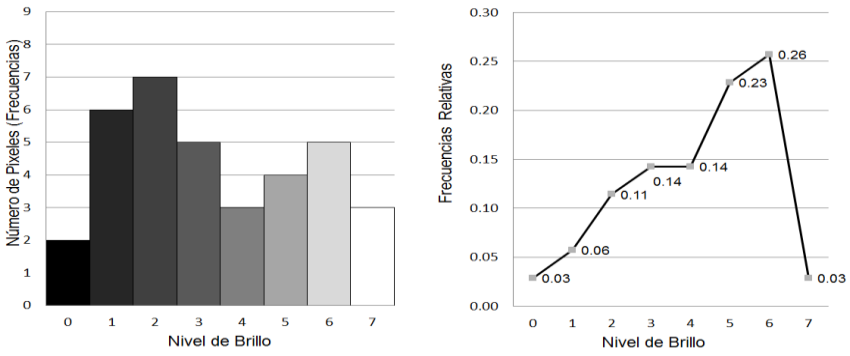
Valores de la variable (Y_i)	Frecuencia Absoluta (n_i)	Frecuencia Absoluta acumulada (N_i)	Frecuencia Relativa (f_i)	Frecuencia Relativa Porcentual ($100 * f_i$)
0	2	1	0,06	5,7
1	6	7	0,17	17,1
2	7	14	0,20	20,0
3	5	19	0,14	14,3
4	3	22	0,09	8,6
5	4	26	0,11	11,4
6	5	31	0,14	14,3
7	3	34	0,09	8,6
Totales	35		1	100

FUENTE: Elaboración propia

A partir de la tabla de frecuencias podemos construir su histograma Figura 9, siendo el parámetro h que definirá el ancho del histograma y podría ser definido arbitrariamente. Para este ejemplo se definió el parámetro $h = 1$.

Cuando se trabaja con histogramas, es posible hacer un histograma de una imagen completa que incluye todos los píxeles, o solo construir un histograma de una parte de la imagen. En este ejemplo, vamos a construir el histograma para solo los píxeles en el cuadro negro de la Figura 8, esta representación es mostrada en la Figura 9, como frecuencia absoluta y relativa de izquierda a derecha.

FIGURA 9: Frecuencia absoluta y relativa de la moneda



FUENTE: Elaboración propia

2.2.4 ESTIMACIÓN DE DENSIDAD

El objetivo del análisis de conglomerados es dividir un determinado conjunto de datos en número de grupos *(Clusters)* o clases, sin que exista una información previa sobre el conjunto de datos. Los datos, sin ningún tipo de ayuda definirán la cantidad de conglomerados existentes y la regla a la que estarán sometidas los grupos.

Muchas de las técnicas del análisis de conglomerados son basados en encontrar semejanzas entre patrones dentro de los datos. Una técnica muy eficiente es la utilización de la función de densidad de probabilidad, donde es posible estimar la densidad de los datos y presentarlos gráficamente. Los picos *(peaks)* de esta función representan conglomerados (Pinto, 1998) .

Existen dos tipos de estimación de densidad: El paramétrico y el no-paramétrico. El primer tipo considera que los datos son retirados de un conjunto conocido, por ejemplo: una distribución normal con media μ y varianza σ^2 . Por lo tanto, la estimación de densidad puede ser realizada encontrándose la estimativa de μ e σ^2. El segundo tipo considera que los datos son obtenidos de un conjunto que no se conoce.

La estimación de densidad es realizada a partir de un conjunto aleatorio X en una función de densidad de probabilidad f, la función irá a proporcionar una descripción de la distribución de ese conjunto permitiendo encontrar probabilidades asociadas con X, a partir de la ecuación:

$$P(a < X < b) = \int_{b}^{a} f(x)d(x) \qquad para\ todo\ \ a < b \tag{8}$$

El objetivo de la estimación de densidades es construir una estimativa de la función de densidad de los datos en cuestión, con frecuencia esa función es desconocida, existen dos tipos de estimación de densidad:

- Paramétrico.
- No-paramétrico.

ESTIMACIÓN DE DENSIDAD NO-PARAMÉTRICO

La estimación de funciones de densidad de probabilidad fdp[3] es necesaria en multitud de escenarios y aplicaciones reales, como son el reconocimiento de patrones, el registro de imagen y segmentación de imágenes.

Suponiendo que se dispone de un conjunto de N patrones d-dimensionales definido por $X = x_{in}$, que es una matriz de datos rectangular de dimensiones $d * N$, siendo x_{in} es el valor de la característica i-ésima para el patrón x_n. El objetivo es modelar la que generó los datos $p(x)$ sin asumir previamente ninguna forma determinada para la fdp.

[3] La fdp es la función de densidad de probabilidad.

Estas técnicas no-paramétricas se fundamentan en la probabilidad de que un vector x, obtenido a partir de una fdp desconocida $p(x)$, caiga dentro de una región R del espacio de entrada viene, por definición dada por:

$$P = \int_R p(x')\, dx' \tag{9}$$

Si se dispone de n muestras obtenidas independientemente a partir de $p(x)$, se puede obtener una buena estimación de probabilidad P a partir de la fracción media de muestras que caen en R, forma que.

$$P \approx \frac{K}{N} \tag{10}$$

Además, si se asumen que P es continua y que no varía apreciablemente sobre la región R, entonces es posible aproximar

$$P = \int_R p(x')\, dx' \tag{11}$$

Por

$$P = \int_R p(x')\, dx' = p(x) * V \tag{12}$$

Donde V es el volumen de R, y x es un patrón incluido en R de $P \approx \frac{K}{N}$, de donde se obtiene.

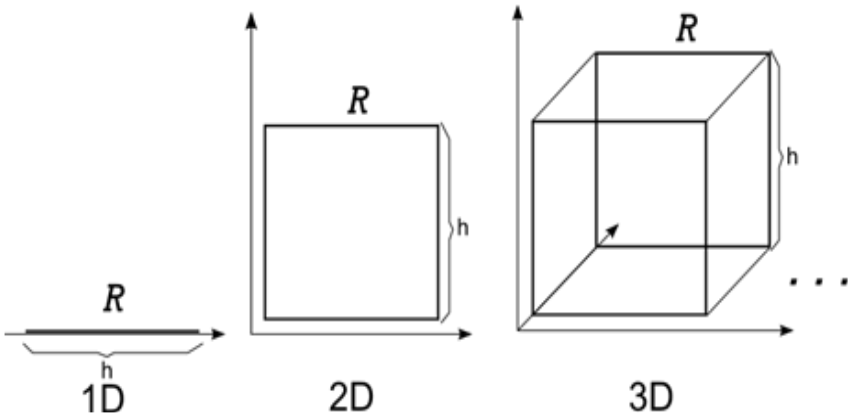
$$p(x) \approx \frac{K}{NV} \tag{13}$$

Atendiendo a la estimación de densidad dada por la Ecuación (12), se puede adoptar los métodos básicos. El primero consiste en elegir un valor fijo para K y determinar V a partir de los datos. Alternativamente se puede fijar el volumen de V y determinar K a partir de los datos. Esto nos lleva a las técnicas de estimación de densidad tipo *kernel*, que se describe a continuación

VENTANA DE PARZEN

En este tipo de abordaje fijaremos el tamaño de la región R definida por un hipercubo para estimar la densidad, también el volumen V y determinamos el correspondiente k a partir de los datos y asumiremos que la región es un hipercubo de tamañno h centrados en el punto x, cuyo volumen es h^{d} Figura 10 (Dit-Yan Yeung and Chow, 2002).

FIGURA 10: Hipercubo de R con h parzen (estimación de la densidad)



FUENTE: http://www.portalaction.com.br

Entonces su volumen viene dado por

$$V = h^d \tag{14}$$

Podemos encontrar una expresión para k, el número de muestras que caen en esta región, definiendo una función *kernel* $\theta(u)$, también conocida como *ventana básica de Parzen* (Parzen, 1962) dada por

$$\varphi(u) = \begin{cases} 1, & |u_i| < 1/2 \\ 0, & otro\ caso \end{cases} \quad \textbf{(15)}$$

De este modo $\varphi(u)$ se corresponde con un cubo unidad centrado en el origen. Por tanto, para cada x_n, la cantidad de $\varphi((x - x_n)/h)$ es igual a la unidad si x_n cae dentro del hipercubo de lado h centrado en x, y es cero en caso contrario. En la literatura, h se le conoce como *parámetro de suavidad de suavizado o ancho de kernel.* El número total de muestras que caen dentro del hipercubo es simplemente.

$$k = \sum_{n=1}^{N} \varphi\left(\frac{x - x_n}{h}\right) \quad \textbf{(16)}$$

Si se substituye la Ecuación (16) en (17) para obtener la Ecuación (18), se obtiene la siguiente estimación.

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^d} \varphi\left(\frac{x - x_n}{h}\right) \quad \textbf{(17)}$$

$$\hat{p}(x) = \frac{1}{Nh^d} \sum_{n=1}^{N} \varphi\left(\frac{x - x_n}{h}\right) \quad \textbf{(18)}$$

Donde $\hat{\mathrm{p}}(\mathrm{x})$ denota la densidad estimada mediante la ventana de Parzen (Parzen, 1962). Esta estimación de fdp puede verse como una superposición de N cubos de lado h, con un hipercubo centrado en cada una de las muestras. Este

método es similar a la estimación basada en histogramas, excepto porque en vez de intervalos se tiene hipercubos cuya posición está determinada por los datos, sin embargo sigue presente el problema de las discontinuidades en la estimación de fdp. Para solucionar este problema, una opción muy utilizada es emplear el núcleo gaussiano.

$$G(x, x_n, h) = \frac{1}{(2\pi h^2)^{d/2}} e^{-\frac{\|x-x_n\|^2}{2h^2}} \quad \textbf{(19)}$$

Donde h representa la desviación estándar en cada dimensión de entrada. De esta forma, la estimación de la fdp mediante Parzen queda.

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} G(x, x_n, h) \quad \textbf{(20)}$$

En general si las funciones de núcleo satisfacen.

$$\varphi(u) \geq 0 \quad \textbf{(21)}$$

Y

$$\int \varphi(u)\, du = 1 \quad \textbf{(22)}$$

Entonces la estimación de la Ecuación (20) satisface que $\hat{p}(x) > 0$ y $\int \hat{p}\,(x)dx = 1$.

CRITERIO DE SILVERMAN

Silverman propuso en (Silverman, 1998) la siguiente regla general de carácter práctico para calcular el valor de h.

$$h_{sil} = \frac{0{,}9A}{N^{1/5}} \tag{23}$$

Siendo $A = mín\left\{s, \frac{r}{1{,}34}\right\}$, donde s es la desviación estándar y r es el rango intercuartil, medidos en el conjunto de datos (García L. & Sancho G, 2010).

2.2.5 ESTIMADOR KERNEL UNIDIMENSIONAL

El Estimador *kernel* (núcleo) es una forma no-paramétrica de estimar las curvas de densidad de probabilidad de una variable aleatoria, donde cada observación es ponderada por la distancia en relación a un valor central, el *kernel*. La idea es centrar cada observación x donde se quiera estimar la densidad, una ventana h que defina la vecindad de x y los puntos que pertenecen a la estimación.

Dada la fórmula del estimador de densidad *Kernel* de la Ecuación (24) y reemplazando los parámetros $d = 1$ para el caso unidimensional, $\hat{p}(x) = f(x)$, $N = n$, $x_n = X_i$ y $\varphi = k$, tenemos.

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} k\left(\frac{x - X_i}{h}\right) \tag{24}$$

Donde:

- X_i, El i-ésimo punto del conjunto de datos del experimento.
- x, Punto donde será calculada la función de densidad de probabilidad.

- k, Una función escogida arbitrariamente.

- h, Coeficiente de suavidad (equivalente al ancho de los rectángulos en el histograma).

- n, Número de resultados del experimento.

La función *kernel* k tiene que satisfacer la siguiente condición:

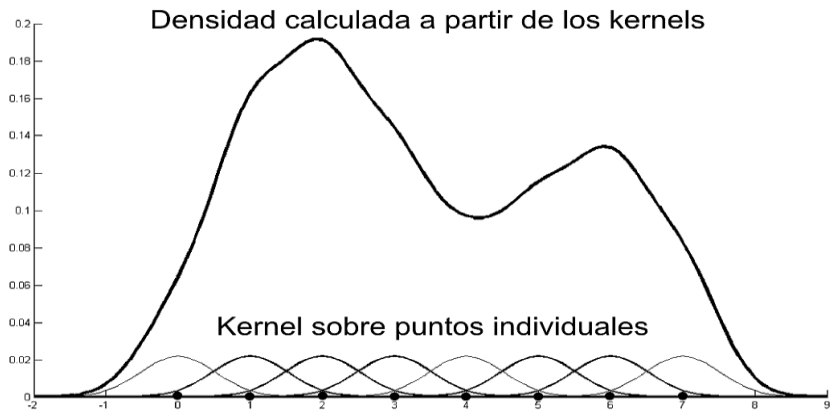
$$\int_{-\infty}^{\infty} k(x)dx = 1 \qquad \textbf{(25)}$$

En este trabajo se utilizará la función *Gaussiana* como función *kernel* k por ser la más suave y presentar los datos de forma más realista. La *Gaussiana* como función *kernel* es definida dela siguiente manera.

$$k(x) = \frac{1}{\sqrt{2 * \pi}} e^{-\frac{x^2}{2}} \qquad \textbf{(26)}$$

Cuando se utiliza una función Gaussiana como función *kernel* se está colocando una pequeña Gaussiana centrada en cada uno de los puntos del conjunto de datos analizado. Posteriormente, se suma toda las Gaussianas a fin de llegar en la función de densidad de probabilidad de todos los puntos como se muestra en la Figura 11.

FIGURA 11: Función de densidad de probabilidad utilizando *kernels*



FUENTE: Elaboración propia

Para ejemplificar el uso del *kernel* se utilizó como ejemplo la Tabla 2, este método utiliza todos los pixeles para calcular la función de densidad de probabilidad para un único punto. Sea $x = 1$, se necesita calcular la función *k* para todos los 35 pixeles, sumarlos y dividirlos por el producto nh, asi sucesivamente.

En la Figura 12, se puede observar el resultado del estimador usando una *Gaussiana* como función *kernel* y el parámetro $h = 1$, dando un agrupamiento, esto es debido al factor de suavidad $h = 1$.

Si el caso fuese, una curva de densidad con el parámetro $h = 0.5$, esta suavidad permite observar dos agrupaciones en los puntos 2 y 4, que son los niveles de intensidad como se presenta en la Figura 13.

Finalmente, la Figura 14 con factor de suavidad $h = 0.1$ presenta claramente los agrupamientos o conglomerados en cada punto de intensidad, como puede ser verificado según la Tabla 2, esto nos indica que si h es pequeño muestra una distribución por cada nivel de intensidad.

A medida que h decrece la figura se va convirtiendo en algo parecido a la frecuencia relativa de la imagen analizada, esto se podría entender como el posicionamiento de una gaussiana en cada nivel de intensidad. Como el objetivo es encontrar la cantidad de conglomerados en los que se puede dividir la imagen, para el ejemplo de la moneda, es posible apreciar claramente que existen 2 picos cuando usamos un factor de suavidad $h = 0.5$.

FIGURA 12: Estimador *kernel* con factor de suavidad (h=1)

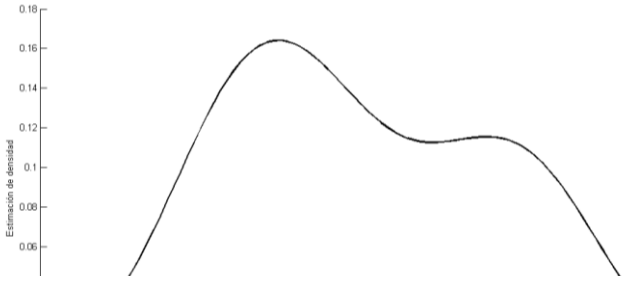
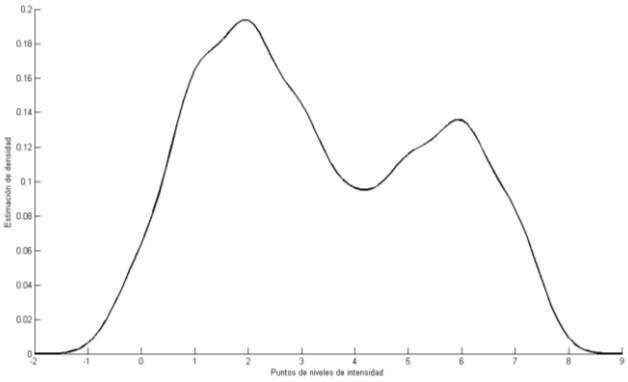
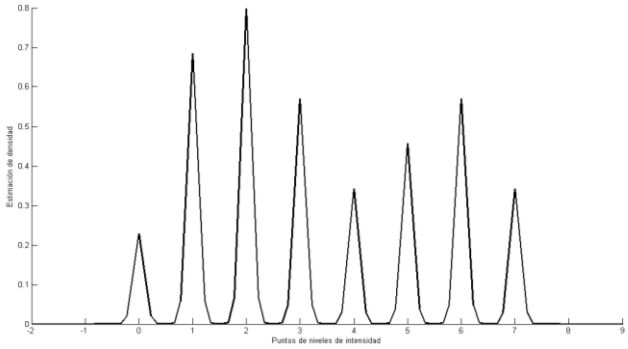


FIGURA 13: Estimador *kernel* con factor de suavidad (h=0.5)



FUENTE**:** Elaboración propia

FIGURA 14: Estimador *kernel* con factor de suavidad (h=0.1)



FUENTE: Elaboración propia

Otro ejemplo de uso del *kernel* unidimensional es presentado con una cantidad considerable de datos, para este caso usaremos 1800 datos aleatorios aglomerados intencionalmente en los puntos 2, 7 y 14 con sus respectivas desviaciones estándar como se observa en la Tabla 3.

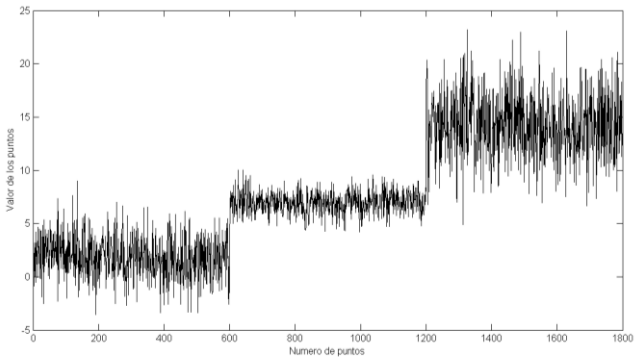
TABLA 3: Parámetros del conjunto de datos unidimensional

Subconjuntos	Media	Desviación Estándar	Número de Puntos
X1	2	2	600
X2	7	1	600
X3	14	3	600

FUENTE: Elaboración propia

La grafica de la Tabla 3 se presenta en la Figura 15, a primera vista es posible apreciar 3 grupos claramente.

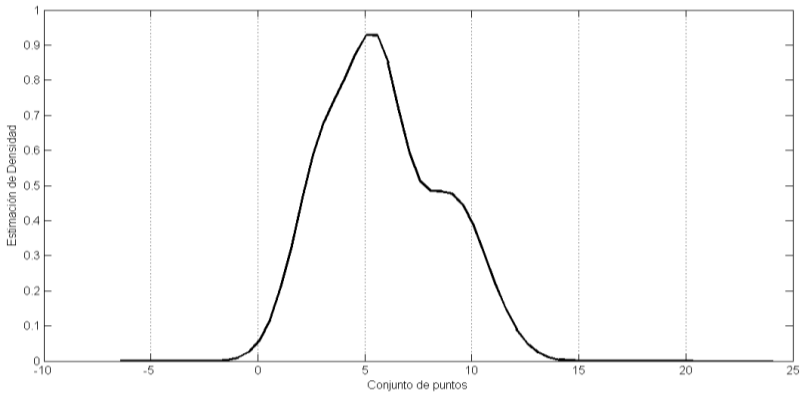
FIGURA 15: Conjunto de datos unidimensional



FUENTE: Elaboración propia

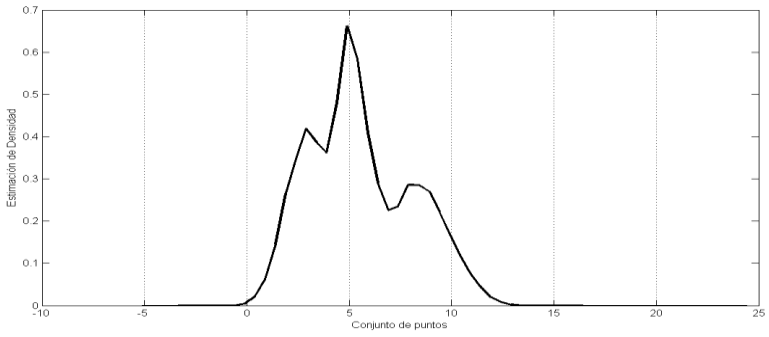
La función de densidad de los datos fue implementada usando el estimador *kernel* con parámetro de suavidad *h=1* y *h=0.5* respectivamente, con la finalidad de observar el tipo de función de densidad resultante, estos resultados son presentados en la Figura 16 y Figura 17, es posible notar la importancia de escoger correctamente el parámetro de suavidad debido a que la Figura 16 se puede identificar apenas un grupo y en la Figura 17 es posible apreciar claramente los 3 grupos.

FIGURA 16: Estimador *kernel* con factor de suavidad h=1



FUENTE: Elaboración propia

FIGURA 17: Estimador *kernel* con factor de suavidad h=0.5



FUENTE: Elaboración propia

Existen varios métodos para escoger los parámetros de suavidad y ningún consenso en cuanto a estos métodos. En este trabajo adoptaremos el método visual donde los gráficos son generados y es escogida la estimación que esta más de acuerdo con las ideas sobre la densidad de datos. Aparentemente simple, este método puede ser perfectamente satisfactorio, una vez que se analiza los gráficos con parámetros de suavidad diferentes se puede obtener más información de los datos del que se consigue con el método automático. Por el contrario, en el análisis de un caso real, cuando no se tiene ninguna información sobre los datos analizados, es necesaria la utilización de un método automático para obtener el número de conglomerados coherente con la realidad

2.2.6 ESTIMADOR KERNEL MULTIDIMENSIONAL

Hasta ahora se había concentrado en la estimación de densidad para apenas un conjunto de datos (*estimador kernel univariado*). Pero muchas aplicaciones importantes envuelven el análisis de datos multivariados. En este trabajo se utiliza un estimador *kernel* multivariado para tratar los datos bidimensionales. La definición de un estimador *kernel* como una suma de picos centrados en cada uno de los puntos del conjunto de datos analizados es fácil generalizarlo para el caso multivariado. Por lo cual, se adoptó la notación $\boldsymbol{x}$ para un conjunto de datos multivariados de *d* dimensiones. Este estimador *kernel multivariado* es definido por:

$$f(x) = \frac{1}{nh^d} k\left\{\frac{1}{h}(x - Xi)\right\} \qquad \textbf{(27)}$$

Dónde:

- Xi, i-ésimo punto de un conjunto de datos multivariado.
- x, punto donde será calculada la función de densidad de probabilidad.
- k, una función escogida arbitrariamente.
- h, coeficiente de suavidad (equivalente a la largura de los rectángulos en el histograma).
- n, número de resultados del experimento.
- d, número de dimensiones.

La función *kernel k(x)* ahora es una función definida por ***x*** de ***d*** dimensiones satisfaciendo la ecuación.

$$\int_{R^d} k(x)dx = 1 \quad \textbf{(28)}$$

La gaussiana como función *kernel* para el caso multivariado es definido por la siguiente ecuación.

$$k(x) = (2\pi)^{-d/2} \left(e^{-\frac{1}{2}X^tX}\right) \quad \textbf{(29)}$$

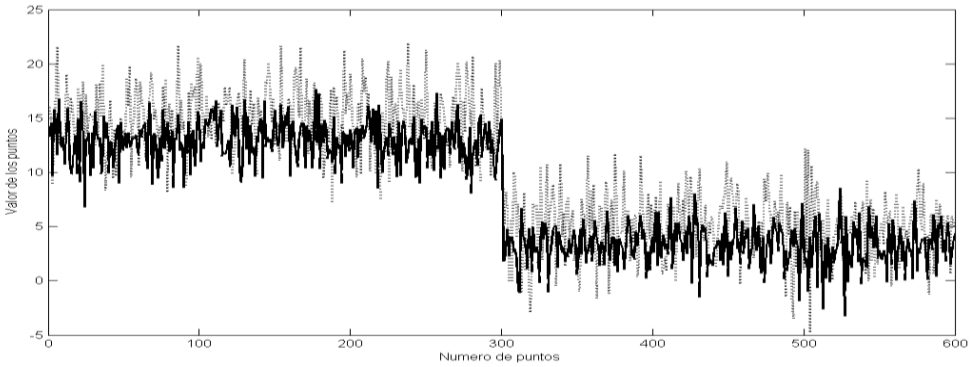
Para el caso multivariado demostrar con el caso bivariado se presenta un ejemplo con los datos generados aleatoriamente pero con parámetros conocidos como son presentados en la Tabla 4, los mismos que son presentados gráficamente en la Figura 18. Para un conjunto A los datos están agrupados alrededor de las medias 15 y 5 y para el conjunto B las medias son 13 y 3. De esta manera esperamos encontrar agrupaciones concentradas en esos puntos.

TABLA 4: Parámetros del conjunto de datos bidimensionales

Subconjuntos	Media	Desviación Estándar	Número de Puntos
A1	15	3	300
A2	5	3	300
B1	13	2	300
B2	3	2	300

FUENTE: Elaboración propia

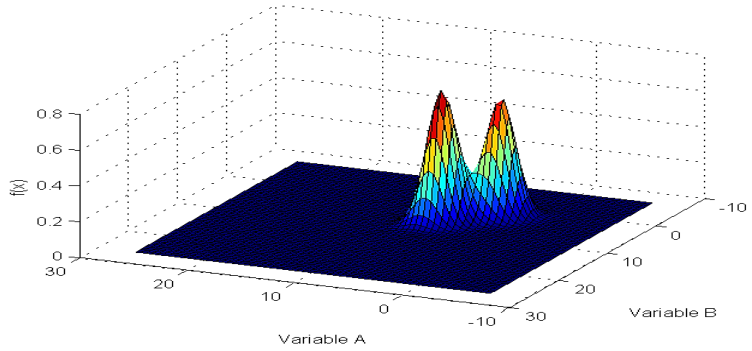
FIGURA 18: Conjunto de datos bidimensional



FUENTE: Elaboración propia

La función de densidad de los datos fue analizado utilizando el estimador *kernel* bivariado con el parámetro de suavidad *h=1*. El resultado es presentado en la Figura 19, se debe recordar que el parámetro *h* establece la definición de la curva.

FIGURA 19: Estimador *kernel* bivariado con factor de suavidad h=1



FUENTE: Elaboración propia

2.2.7 ALGORITMOS GENÉTICOS

El origen de las especies explicado por el científico inglés Charles R Darwin se une al explorador Beagle, como geólogo y naturalista, para un viaje de seis años alrededor del mundo. A lo largo de este viaje se ejecutó una extensa recolección de las muestras zoológicas y botánicas que resultan en la construcción de su teoría de selección natural, publicada en 1859 en el libro “On The Origin of Species by Means of Natural Selection”. En su teoría, el propone demostrar que los organismos tienden a producir descendientes ligeramente diferente de los progenitores y que la selección natural tiende a favorecer a aquellos que mejor se adaptan al medio ambiente. Algunos individuos tienen características que los convierten en los más aptos para sobrevivir y mayores posibilidades de reproducirse y transmitir sus características a sus descendientes y, con el tiempo especies distintas se desarrollaran. Esas ideas fueron muy criticadas por los naturalistas de aquella época que creían que las especies eran creadas separadamente por medio de la generación espontánea o por un principio divino (Whitley, 1994).

A inicios del siglo XX, comenzaron a surgir muchos trabajos sobre evolución basados en los principios de la herencia genética. Estos trabajos unían la genética y la selección natural creando el principio básico de la Genética Poblacional: Una población de organismos que tiene su reproducción realizada sexualmente producirá individuos diferentes por medio del cruce genético y mutaciones (Mendes Filho, 1998).

A partir de la década de 50, muchos biólogos comenzaron a estudiar y a desarrollar simulaciones de sistemas genéticos utilizando computadoras. En

1975, después de mucha investigación, John Holland publico el libro “Adaptation in Natural and Artificial Systems” que hoy es usado como bibliografía básica en el estudio de los algoritmos genéticos. A partir de este trabajo, surgieron varios con éxito utilizando los algoritmos genéticos en problemas de optimización y búsqueda (Pinto, 1998).

DEFINICIÓN DE LOS ALGORITMOS GENÉTICOS

Los Algoritmos genéticos (AGs) son algoritmos de búsqueda basados en mecanismos de selección natural y genética y fueron desarrollados por John Holland y su equipo de investigación en la Universidad de Michigan (Holland, 1992), (Goldberg, 1989), su investigación tenía como objetivo explicar rigurosamente los procesos adaptativos de sistemas naturales y desarrollar un software de un sistema artificial que implementase los mecanismos importantes de estos sistemas naturales. Este abordaje condujo a importantes descubrimientos para la ciencia de sistemas naturales y artificiales.

Una tarea de búsqueda y optimización abarca, varios componentes, el espacio de búsqueda y la función de evaluación. Algunas técnicas tradicionales se inician con un único candidato que, iterativamente es manipulado utilizando algunas heurísticas directamente asociadas al problema a ser solucionado. Utilizadas con éxito en varias aplicaciones, estas técnicas no son bastante robustas y su simulación en computador solían convertirse muy complejas. Los AGs. Son simples del punto de vista computacional por consiguiente son métodos de búsqueda extremadamente eficientes. Partiendo de una población de candidatos, los AGs realizan una búsqueda paralela en diferentes áreas del espacio de soluciones (Norvig, 1995).

También podemos identificar en relación a las técnicas tradicionales, que los AGs trabajan con una codificación del conjunto de parámetros y no con los mismos. Otra comparación seria que ellos utilizan informaciones de recompensa o costo y no derivadas u otro conocimiento auxiliar. Ellos son muy eficientes para búsqueda de soluciones de optimización, o aproximadamente optimas, en una grande variedad de problemas, pues no imponen muchas de las limitaciones encontradas en los métodos de búsqueda tradicional (Mendes Filho, 1998).

Los AGs son métodos de búsqueda ciega por no tener conocimiento específico del problema a ser resuelto, teniendo como guía apenas una función de evaluación. Los números aleatorios, no ejecutan búsquedas sin rumbo, pues por medio de procesos iterativos (generaciones) ellos exploran informaciones históricas de cada generación para encontrar nuevos puntos de búsqueda donde son esperados mejores resultados.

CONCEPTOS BÁSICOS

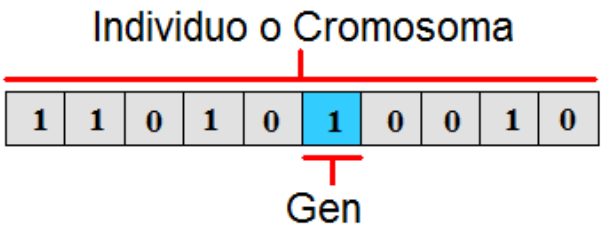
Los AGs forman una clase de procedimientos con varias etapas y cada una de estas etapas posee muchas variaciones, es por eso que se usan términos de **"algoritmos genéticos"** o **"un algoritmo genético"** y no **"el algoritmo genético"**.

El funcionamiento de un algoritmo genético clásico se explica exponiendo naturalmente algunos conceptos básicos. El primer paso es generar una población inicial donde sus individuos representan posibles soluciones para un determinado problema. Esta población inicial puede ser generada a partir de valores aleatorios o a partir de valores predefinidos (semillas). Cada individuo es evaluado de acuerdo con el problema en cuestión donde los más aptos son

mantenidos y los demás son eliminados. Por medio de operaciones genéticos (cruce y mutación) los individuos restantes genera descendientes (reproducción) los cuales tiene una grande posibilidad de ser los más del que sus progenitores. La reproducción es repetida hasta que una condición deparada sea satisfecha. Esta condición puede estar relacionada con una solución satisfactoria, el número de generaciones o hasta el número de procesamiento (Goldberg, 1989).

En un algoritmo genético clásico un individuo es representado por una cadena binaria (0,1) donde cada elemento es llamado de *Gen* Figura 20. Cada elemento de la cadena puede indicar la presencia “1” o ausencia “0” de una determinada característica que en la genética es conocida como genotipo. Los elementos combinados forman las características reales del individuo o su fenotipo.

FIGURA 20: Representación de un cromosoma de genes binarios



FUENTE: (Izidoro, 2008)

La representación de cada cromosoma es calculado teniendo en consideración los siguientes criterios:

- Si el dominio de una variable x_i, es $[a_i, b_i]$ y la precisión requerida después del punto decimal es de 5 dígitos.

- El rango del dominio debe ser dividido en al menos $(b_i - a_i) * 10^5$ rangos de igual tamaño. Los bits requeridos para m_i serán.

- Una vez calculado $(b_i - a_i) * 10^5$ se busca el valor que corresponde a m_i, que nos indica el número de bits a ser utilizado.

$$2^{m_i-1} < (b_i - a_i) * 10^5 \leq 2^{m_i} - 1 \quad \textbf{(30)}$$

OPERADORES PARÁMETROS GENÉTICOS

La función de los operadores genéticos es, por medio de un proceso recursivo, transformar la población inicial en una población que represente un resultado satisfactorio. Un algoritmo genético clásico es compuesto por tres operaciones (Goldberg, 1989) (Izidoro, 2008):

1. Reproducción o Selección
2. Cruce
3. Mutación

La idea básica de la reproducción y selección de los mejores individuos de la población corriente por medio de una función de aptitud (el más apto). Los individuos con un alto valor de aptitud tendrán una alta probabilidad de contribuir con uno o más descendientes en la próxima generación.

La operación de reproducción puede ser implementada de varias formas por lo tanto, el método más utilizado es el método de la ruleta. En este método cada individuo de la población corriente tiene su representación en la ruleta de acuerdo con su valor de aptitud. Individuos con valores altos de aptitud tendrán un segmento mayor dentro de la ruleta y los individuos con valores menores tendrán segmentos menores como se observa en la Tabla 5 y su respectiva grafica la Figura 21. Posteriormente la ruleta es girada *n* veces y de acuerdo con

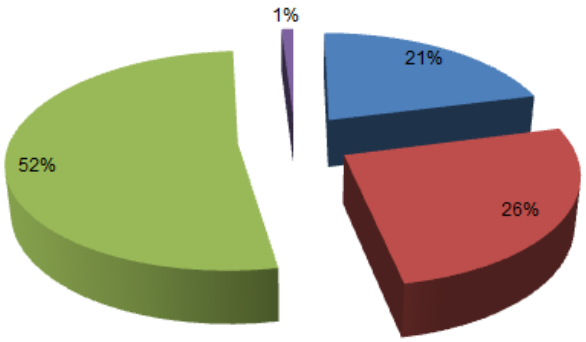
el tamaño de la población los individuos que salen sorteados formaran parte de la próxima generación.

TABLA 5: Ejemplo de una población con respectivos valores de aptitud

No	Individuos	Aptitud	% del Total
1	10011	361	21
2	10101	441	26
3	11110	900	52
4	00011	9	1
Total		1711	100

FUENTE: (Izidoro, 2008)

FIGURA 21: Ruleta de selección según valores de aptitud



FUENTE: (Izidoro, 2008)

El operador de cruce es utilizado después de la reproducción. En esta fase sucede el cambio de segmentos entre parejas de individuos dando origen a nuevos individuos. Con este cambio lo que se intenta hacer es propagar las características de los individuos más aptos de la población corriente para futuras generaciones.

Los individuos seleccionados por la ruleta serán transferidos para una piscina de apareamiento “mating pool” donde el cruce es realizado en dos pasos:

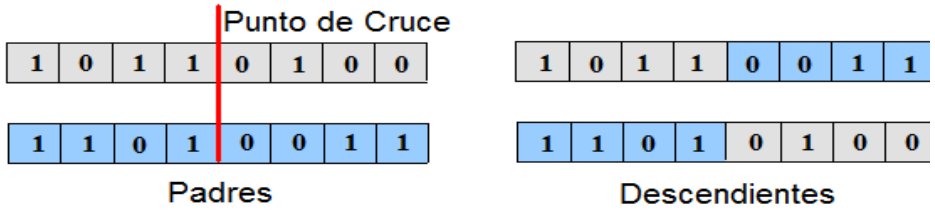
- Primero, consiste en definir las parejas de individuos de forma aleatoria.

- Segundo, fijar un punto aleatorio de desdoblamiento del individuo a lo largo de la cadena “string” que lo representa.

A partir de este punto se realiza el intercambio de genes entre el par de individuos. El operador de cruce puede ser implementado de varias formas, entre las más usadas están:

- Un punto de cruce, el punto de desdoblamiento del cromosoma es escogido de forma aleatoria y a partir de este punto las informaciones genéticas del par de individuos serán intercambiadas como se aprecia en la Figura 22.

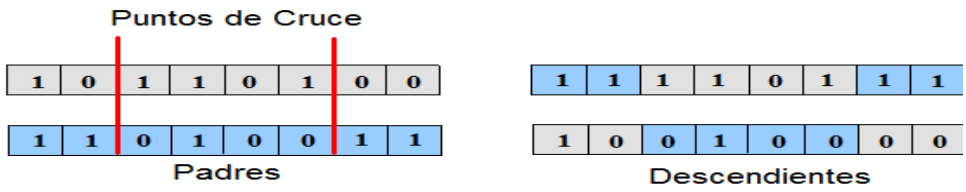
FIGURA 22: Punto de cruce



FUENTE: (Izidoro, 2008)

- Múltiples puntos, realizada de manera similar al cruce de un punto, por lo tanto el intercambio de genes es realizada en más de un punto a lo largo de la cadena que representa el individuo Figura 23.

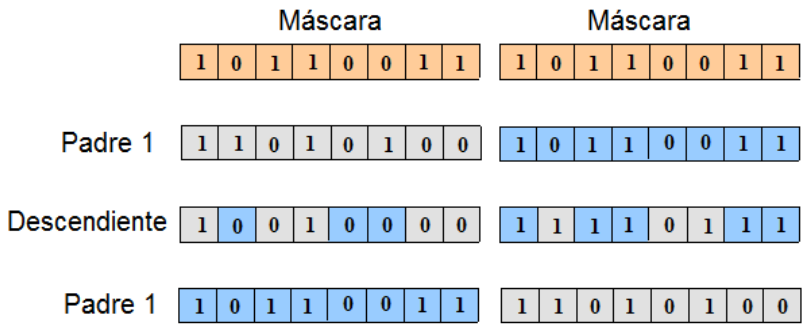
FIGURA 23: Múltiples puntos de cruce



FUENTE: (Izidoro, 2008)

- Cruce Uniforme, el cruce es realizado basado en una máscara generada de forma aleatoria con el mismo número de genes de los individuos que serán cruzados. Si existiese una máscara de cruce el gen correspondiente será copiado del primer padre y si hubiese cero será copiado del segundo padre. Una vez formado el primer descendiente el proceso se repetirá con los padres cambiando para formar el segundo descendiente Figura 24

FIGURA 24: Cruzamiento uniforme

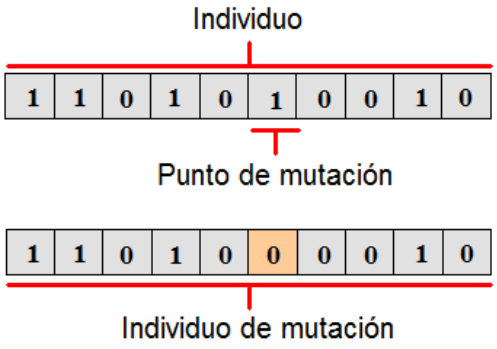


FUENTE: (Izidoro, 2008)

Después del cruzamiento la operación de mutación es aplicada para cada gen de todos los individuos de forma aleatoria. La operación consiste simplemente en alterar el valor del gen (1 para 0 y viceversa) Figura 25. Utilizada para dar una nueva información para la población, la mutación previene la saturación de la población con individuos semejantes.

El operador de mutación garantiza que la probabilidad de llegar a cualquier punto del espacio de búsqueda nunca será cero, más allá de encontrar el problema de los mínimos locales (Izidoro, 2008).

FIGURA 25: Operación de mutación



FUENTE: (Izidoro, 2008)

El funcionamiento de un algoritmo genético y sus operadores depende de algunos parámetros que pueden ser ajustados conforme las necesidades de cada problema. Los parámetros son (Izidoro, 2008):

- Tasa de Cruzamiento: valor que determina la probabilidad de cruzamiento dentro de una población. Tasas altas implican en un mayor número de cruzamientos y, consecuentemente, nuevos individuos. Por lo tanto, se debe observar que tasas muy altas pueden resultar en perdida de las características de los individuos en las generaciones futuras. Individuos con buena aptitud pueden desaparecen en una próxima generación. Valores bajos pueden convertir su funcionamiento extremadamente lento.

- Tasa de mutación: determina la probabilidad de ocurrir una mutación. Los valores bajos son usados para desplazar el objetivo de la búsqueda de los algoritmos genéticos. Tasas altas convierten la búsqueda

esencialmente aleatoria pudiendo implicar en la posibilidad de que una buena solución sea distribuida.

- Tamaño de la población: poblaciones pequeñas presentan un bajo desempeño, pues actúan en un pequeño espacio de búsqueda del problema. Poblaciones grandes efectúan una cobertura significativa del espacio de búsqueda del problema. De esta forma se evita también el problema de mínimos locales.

REPRESENTACIÓN DE LOS INDIVIDUOS

Un punto importante es la codificación de los individuos, en un problema donde los valores involucrados son números enteros la solución sería una conversión directa de un número para la base binaria. El número convertido en cadena de 0's y 1's representan el individuo (Goldberg, 1989).

Pero, se debe tener presente que la mayoría de las aplicaciones envuelven valores reales. Para trabajar con números reales no se puede ejecutar una simple conversión de la base decimal para la base binaria. La técnica más utilizada es efectuada por medio de una representación discreta de los datos dentro de un intervalo $[x_{max},\ x_{min}]$ en una cantidad de puntos 2t, tal que la distancia entre puntos consecutivos sea menor que un valor de tolerancia especificado, o sea (Izidoro, 2008).

$$\frac{x_{max} - x_{min}}{2^t - 1} < TOL \qquad \textbf{(31)}$$

Por lo tanto, cada punto del espacio de búsqueda será representado por un número binario de tamaño t, comenzando por 0…0 que representa x_{min} y terminando en 1…1 que representa x_{max}.

El punto principal de la representación esta en calcular el tamaño (t) de los individuos. Con base en el valor de tolerancia este tamaño puede ser calculado a partir de la siguiente ecuación (Izidoro, 2008).

$$t = log_2\left(1 + \frac{x_{max} - x_{min}}{TOL}\right) \quad \textbf{(32)}$$

Por ejemplo, en un intervalo $x \in [0,1]$ es una precisión de dos casas decimales ($TOL = 5 \times 10^{-3}$), entonces el tamaño de individuo seria.

$$t = log_2\left(1 + \frac{1 - 0}{0.005}\right) = 8 \quad \textbf{(33)}$$

De esta forma, se puede utilizar individuos con el tamaño de 8 bits para representar el intervalo $x \in [0,1]$ con precisión menor igual a 0.005.

Con la definición del tamaño de los individuos y del intervalo $[x_{max}, x_{min}]$ se puede realizar las modificaciones necesarias. Un valor real deberá ser convertido en un valor entero y este por lo tanto deberá ser codificado en binario para que sean realizadas las operaciones de cruce y mutación. Para el proceso inverso se debe convertir el valor binario para un valor entero y finalmente, efectuar la conversión de entero para real. La codificación para real seria (Izidoro, 2008).

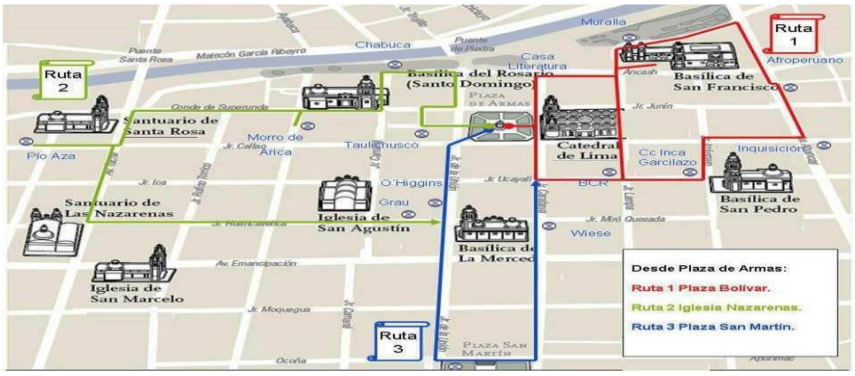
$$x_{real} = x_{bin}\left(\frac{x_{max} - x_{min}}{2^t - 1}\right) + x_{max} \quad \textbf{(34)}$$

Caso 1: Un turista pretende elaborar un itinerario para visitar las 9 iglesias del centro histórico de Lima cercado, El recorrido será a pie, comienza y finaliza en la iglesia Santa Rosa y debe ser realizado con la menor distancia en Km. y menor tiempo en minutos. Cada iglesia será visitada solamente una vez (la variable a ser optimizada es la distancia y tiempo), ¿Cual deberá ser la mejor ruta a fin de obtener un recorrido en kilometraje mínimo? .

Para este ejemplo se utilizó un mapa para ubicar geográficamente las iglesias en el centro histórico de Lima como se puede apreciar en la Figura 26, las mismas que son enumeradas como sigue:

1. Iglesia de Santa Rosa
2. Iglesia Las Nazarenas
3. Iglesia de San Marcelo
4. Iglesia de San Agustín
5. Iglesia de Santo Domingo
6. Iglesia de la Merced
7. Catedral de Lima
8. Convento San Francisco
9. Iglesia San pedro.

FIGURA 26: Plano de ubicación de las iglesias (centro histórico de Lima)



FUENTE: Municipalidad de Lima Metropolitana

A partir de este plano se comenzó a buscar y ubicar los objetivos en el Google Maps, el cual nos permite obtener información sobre distancia (mt.) entre los diferentes puntos los mismos que son ubicados en una matriz bidimensional como se muestra en la Tabla 6.

TABLA 6: Matriz de distancias entre iglesias en Lima cercado (Mt.)

	Iglesia de Santa Rosa	Iglesia Las Nazarenas	Iglesia de San Marcelo	Iglesia de San Agustín	Iglesia de Santo Domingo	Iglesia de La Merced	Catedral de Lima	Convento San Francisco	Iglesia San Pedro
Iglesia de Santa Rosa	0	450	650	750	500	1000	750	1000	1300
Iglesia Las Nazarenas	450	0	270	600	900	600	1000	1400	1000
Iglesia de San Marcelo	650	270	0	500	750	600	1000	1300	1000
Iglesia de San Agustín	750	600	500	0	350	250	250	850	600
Iglesia de Santo Domingo	500	900	750	350	0	600	400	550	850
Iglesia de La Merced	1000	600	600	250	600	0	400	800	400
Catedral de Lima	750	1000	1000	250	400	400	0	400	450
Convento San Francisco	1000	1400	1300	850	550	800	400	0	500
Iglesia San Pedro	1300	1000	1000	600	850	400	450	500	0

FUENTE: Elaboración propia

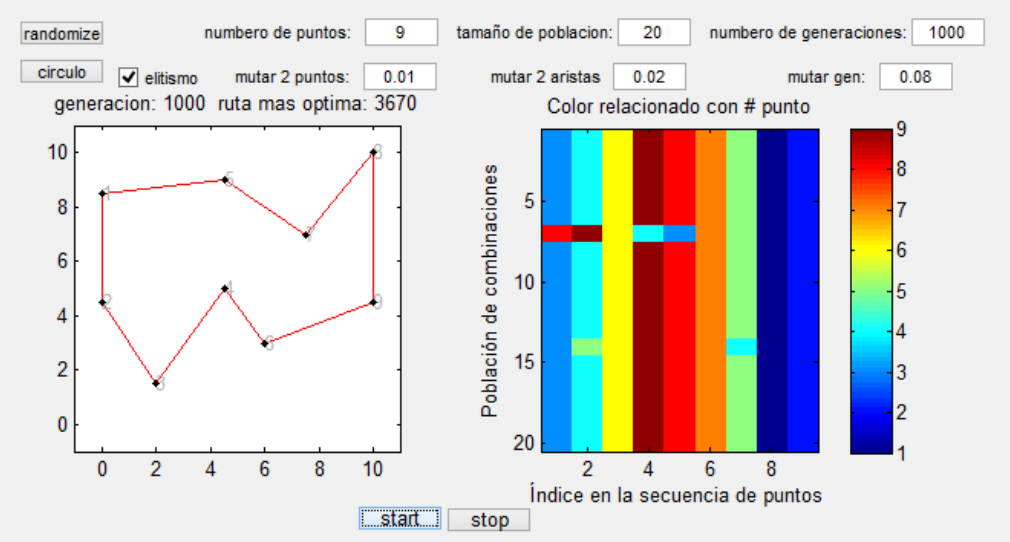
Esta matriz de distancias es adicionada en el programa para poder encontrar la ruta con la mínima distancia de recorrido como, el resultado es

presentado en la Figura 27. Donde los algoritmos genéticos fueron configurado de la siguiente forma:

- Número de Puntos: 9.
- Tamaño de la población (cromosomas): 20.
- Numero de generaciones: 1000.
- Porcentaje de mutación en 2 puntos: 0.01.
- Porcentaje de mutación en 2 aristas: 0.02.
- Porcentaje de mutación en los Gens: 0.08.

Esta configuración de los algoritmos genéticos proporciona un resultado mínimo de 3670 mt. o 3.67 km lineales aproximadamente, y el orden de los puntos a recorrer serian [1 2 3 4 6 9 8 7 5 1].

FIGURA 27: Optimización de ruta usando Algoritmos Genéticos



FUENTE: Elaboración propia

APLICACIONES DE LOS ALGORITMOS GENÉTICOS

Las aplicaciones de los AGs se extienden a diversas áreas, algunas de las cuales son:

Optimización: Se trata de un campo especialmente estudiado para el uso de los AGs, por las características intrínsecas de estos problemas. Los AGs se han utilizado en numerosas tareas de optimización, incluyendo la optimización numérica, y los problemas de optimización combinatoria (Weile, 1997).

Programación automática: Los AGs se han empleado para desarrollar programas para tareas específicas, y para diseñar otras estructuras computacionales tales como el autómata celular, y las redes de clasificación (Maragathavalli, 2012).

Aprendizaje máquina: Los algoritmos genéticos se han utilizado también en muchas de estas aplicaciones, tales como la predicción del tiempo o la estructura de una proteína. Han servido asimismo para desarrollar determinados aspectos de sistemas particulares de aprendizaje, como pueda ser el de los pesos en una red neuronal, las reglas para sistemas de clasificación de aprendizaje o sistemas de producción simbólica, y los sensores para robots (Chisholm, 1997).

Economía: En este caso, se ha hecho uso de estos Algoritmos para modelar procesos de innovación, el desarrollo estrategias de puja, y la aparición de mercados económicos (Man, Tang, & Kwong, 1996).

Sistemas inmunes: A la hora de modelar varios aspectos de los sistemas inmunes naturales, incluyendo la mutación somática durante la vida de un

individuo y el descubrimiento de familias de genes múltiples en tiempo evolutivo, ha resultado útil el empleo de esta técnica (Man, Tang, & Kwong, 1996).

Ecología: En la modelización de fenómenos ecológicos tales como las carreras de armamento biológico, la co-evolución de parásito-huésped, la simbiosis, y el flujo de recursos (Man, Tang, & Kwong, 1996).

Genética de poblaciones: En el estudio de preguntas del tipo ¿Bajo qué condiciones será viable evolutivamente un gen para la recombinación? (Man, Tang, & Kwong, 1996).

Evolución y aprendizaje: Los AGs se han utilizado en el estudio de las relaciones entre el aprendizaje individual y la evolución de la especie (Man, Tang, & Kwong, 1996).

Sistemas sociales: En el estudio de aspectos evolutivos de los sistemas sociales, tales como la evolución del comportamiento social en colonias de insectos, y la evolución de la cooperación y la comunicación en sistemas multiagentes (Man, Tang, & Kwong, 1996).

Aunque esta lista es reducida, pero si transmite la idea de variedad de aplicaciones que tienen los algoritmos genéticos.

2.2.8 ANÁLISIS DE CONGLOMERADOS UTILIZANDO AGs

La técnica del análisis de conglomerado utilizando AGs es muy simple y eficiente. La idea principal consiste en utilizar los AGs para encontrar los máximos de la función de densidad de un conjunto de datos (Pinto, 1998), los algoritmos genéticos para esta implementación utiliza el estimador *kernel* como

función de aptitud. El objetivo es encontrar los máximos locales obtenidos por la función de aptitud una vez que el agrupamiento delos datos no está apenas en el máximo global de la función de densidad. Una de las características de los AGs es que ellos pueden encontrar máximos locales de un conjunto de datos a partir de una población pequeña con un número pequeño degeneraciones (Serrada, 1996).

La ejecución de los algoritmos genéticos apenas una vez, no garantiza que todos los máximos locales serán encontrados. Se ejecuta los algoritmos genéticos *N* veces, almacenando la población final después de *M* generaciones. En seguida, se calcula la función de densidad de la población de soluciones y los picos presentados representaran los clusters. Los pasos de esta técnica son.

1. Definir el estimador *kernel* como función de aptitud.
2. Definir una pequeña población inicial.
3. Definir un valor pequeño para el número máximo de generaciones.
4. Ejecutar los algoritmos genéticos *N* veces y guardar la población final en cada ejecución.
5. Utilizar el estimador *kernel* para estimar la función de densidad de la población final obtenida después de la ejecución de los algoritmos genéticos *N* veces.
6. El número de picos será el número de clusters y las variables de cada pico será el centro de los clusters.

ALGORITMOS GENÉTICOS PARA LA OPTIMIZACIÓN

Un algoritmo genético clásico fue implementado con el propósito de ser utilizado como base para otros programas pertinentes a este trabajo. El programa fue escrito en lenguaje MATLAB debido a popularidad en el ámbito de la ingeniería.

Para demostrar la funcionalidad de los algoritmos genéticos se utilizó el ejemplo de maximizar la función mostrada en la siguiente ecuación donde *x* y *y* varía entre -1 y 1, tambien mostrada en la Figura 28.

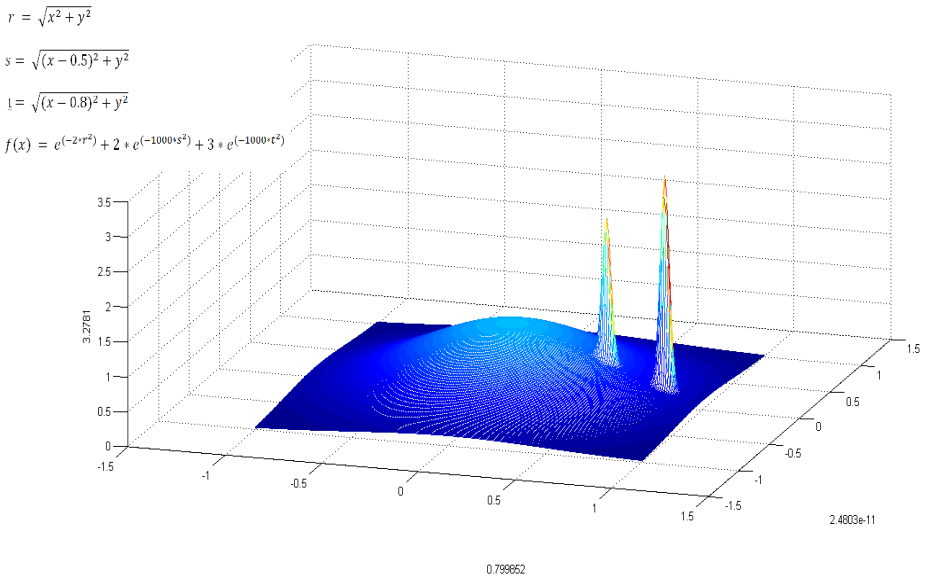
$$r = \sqrt{x^2 + y^2} \quad \textbf{(35)}$$

$$s = \sqrt{(x - 0.5)^2 + y^2} \quad \textbf{(36)}$$

$$t = \sqrt{(x - 0.8)^2 + y^2} \quad \textbf{(37)}$$

$$f(x) = e^{(-2*r^2)} + 2 * e^{(-1000*s^2)} + 3 * e^{(-1000*t^2)} \quad \textbf{(38)}$$

FIGURA 28: Gráfico generado para la función f(x)



FUENTE: Elaboración propia

Inicialmente las variables del problema fueron definidas como cadenas de tamaño finito. En el problema abordado, la codificación de los individuos fue realizada con los valores binarios. El próximo paso esta en seleccionar una población inicial aleatoriamente o predefinida (semilla).

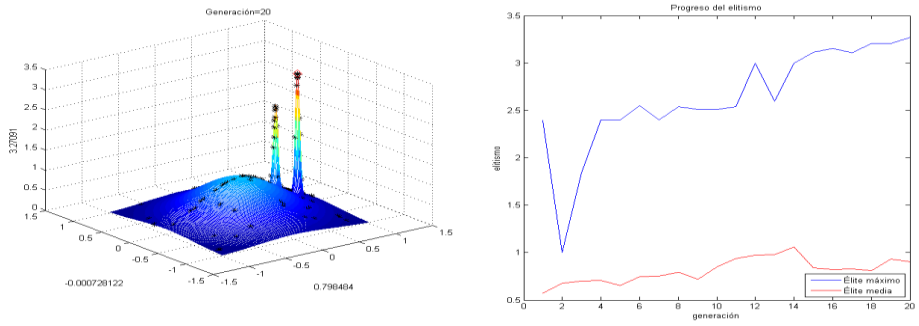
Aun en el proceso de configuración se debe definir las tasas de mutación y cruce. Los valores para la tasa de cruzamiento son típicamente definidos entre 0.6 y 1.0, para la tasa de mutación los valores se encuentran en torno a 0.001 (Beasley, 1993). En el programa fueron configurados valores estándares *(default)* que pueden ser alterados.

Estos parámetros son fijados en:

- Tasa de cruzamiento: 0.8
- Tasa de mutación: 0.5
- Población: 100
- Generaciones: 20

Esta configuración nos permite encontrar óptimos locales, que es lo que se busca para el presente proyecto de segmentación de imágenes, esta situación se encuentra cuando la configuración presenta menor cantidad de generaciones y una población relativamente grande como se puede apreciar en la Figura 29 del lado izquierdo y a la derecha está la curva donde se puede apreciar el progreso del valor máximo alcanzado y el medio.

FIGURA 29: Configuración de AGs para la búsqueda de óptimos locales



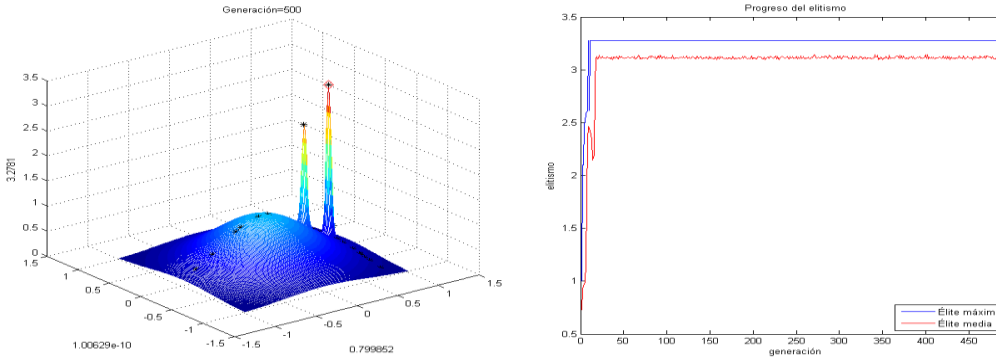
FUENTE: Elaboración propia

Estos parámetros fueron fijados en:

- Tasa de cruzamiento: 0.8
- Tasa de mutación: 0.5
- Población: 100
- Generaciones: 20

Esta otra configuración permite encontrar óptimos global, esta situación se encuentra cuando la configuración presenta mayor cantidad de generaciones y una población relativamente grande y para asegurar una convergencia rápida hacia el óptimo global, se consideró la tasa de mutación 0.03, el resultado se puede verificar en la Figura 30 de izquierda a derecha está la curva donde se puede apreciar el progreso del valor máximo alcanzado por el algoritmo.

FIGURA 30: Configuración de AGs para la búsqueda de óptimo global



FUENTE: Elaboración propia

ALGORITMO GENÉTICO PARA ANÁLISIS DE CONGLOMERADOS

El problema de conglomerados consiste en que, dado un conjunto de datos, descubrir la estructura de grupos que se encuentra en dicho conjunto, si es que existe. Las técnicas de conglomerados más utilizadas son dos: el conglomerado particional y jerárquico (Sheikh, 2008).

El conglomerado particional consiste en dividir el conjunto de datos en grupos, de manera que los datos que se encuentren en un grupo sean lo más parecidos entre sí, a la vez que lo más diferentes posible a los datos que se encuentren en los demás grupos. Es decir, se trata de dar una partición del conjunto de datos. El conglomerado jerárquico es otra técnica de conglomerado que en vez de crear una única partición, crea una sucesión encajada de particiones cuya estructura puede ser representada por medio de un árbol un ejemplo de conglomerado jerárquico se puede encontrar en (Sheikh, 2008).

El pseudocódigo de los AGs muestra el funcionamiento del programa.

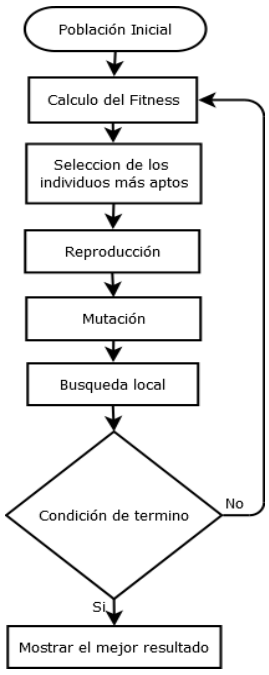
AG { | *i = 1;*

```
PI = poblacion_inicial( );
Repita de i hasta N{
P = PI;  j = 1;
Repita de  j hasta M{
        Fitnes (P);
        Real_to_Int (P);
        Decimal_Binário (P);
        Cruce (P);
        Mutación (P);
        Binário_Decimal (P);
        Int_to_Real (P);
        j = j + 1;
}
Selección_Población (P);
i = i + 1;
}
```

Dónde:

- PI, población inicial.
- P, población.
- N, número de repeticiones del procedimiento AG.
- M, número de generaciones.
- i, contador do número de repeticiones del procedimiento AG.
- j, contador del número de generaciones.

FIGURA 31: Diagrama del AG para la búsqueda local de picos máximos



FUENTE: Elaboración propia

En el presente trabajo se trata el conglomerado particional, el mismo que se escribirá mediante el siguiente ejemplo. La utilización de los algoritmos genéticos en el análisis de conglomerado se implementó en Matlab, para esto generando tres subconjuntos de datos con una distribución normal basados en los parámetros presentados en la Tabla 7 y la configuración de los AGs en la Tabla 8, finalmente la representación gráfica de estos datos son presentados en la Figura 32.

TABLA 7: Parámetros del conjunto de datos unidimensionales

Subconjuntos	Media	Varianza	Número de Puntos
1	28	2	500
2	20	1	400
3	35	1	500
4	15	1	600
Total			**2000**

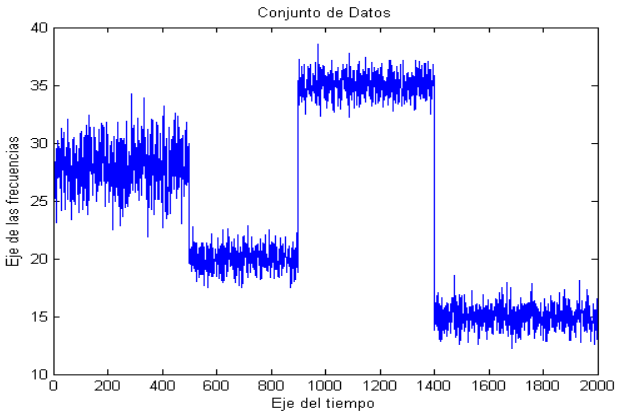
FUENTE: Elaboración propia

TABLA 8: Configuración del programa AG para el caso unidimensional

Parámetro	Característica
Población inicial	Aleatoria
Tamaño de la Población	40
Número de Generaciones	500
Número de Ejecuciones del AG	1
Número de Puntos	2000
Tasa de Cruzamiento	0.4
Tasa de Mutación	0.08
Función de Aptitud	Estimador Kernel (Gaussiana)
Precisión	0.1
Tamaño de Individuo	6

FUENTE: Elaboración propia

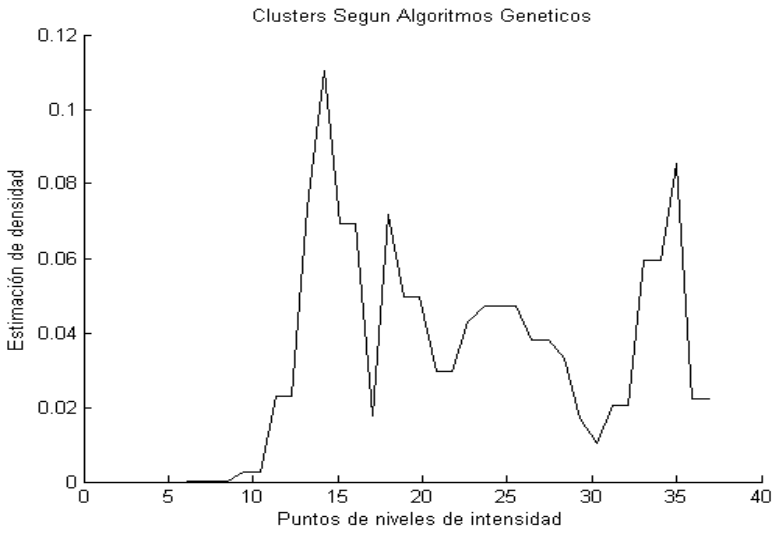
FIGURA 32: Conjunto de datos generados a partir de la TABLA 7



FUENTE: Elaboración propia

Los datos generados según los parámetros de la Tabla 7 son procesados con los AGs, esto permitió encontrar el número de conglomerados, para esto los AGs fueron configurados previamente según la Tabla 8, el resultado del procesamiento es presentado en la Figura 33.

FIGURA 33: Conjunto de datos con cuatro tipos de frecuencias



FUENTE: Elaboración propia

2.2.9 ANÁLISIS DE CONGLOMERADOS UTILIZANDO K-MEDIAS

El análisis de conglomerados de K-medias es un método de agrupación de casos que se basa en las distancias existentes entre ellos, ahora en un conjunto de variables (este método de aglomeración nos permite agrupar variables). En la versión implementada del algoritmo K-medias el usuario determina inicialmente el numero k de conglomerados que desea obtener, a continuación se inicia la lectura secuencial del archivo de datos asignando cada caso al centro más próximo y actualizando el valor de los centros a medida que se van incorporando nuevos casos. Una vez que todos los casos a uno de los k conglomerados, se inicia un proceso iterativo para calcular los centroides finales de esos k conglomerados (Bow, 1984).

El análisis de conglomerados de K-medias es especialmente útil cuando se dispone de un gran número de casos. Existe la posibilidad de utilizar la técnica de manera exploratoria, clasificando los casos e iterando para encontrar la ubicación de los centroides, o solo como técnica de clasificación, clasificando los casos a partir de estos centroides conocidos, suministrados por el usuario. Cuando se utiliza como técnica exploratoria, es habitual que el usuario desconozca el número idóneo de conglomerados, por lo que es conveniente repetir el análisis con distinto número de conglomerados y comparar las soluciones obtenidas; en estos casos también puede utilizarse el método análisis de conglomerados jerárquicos con una sub-muestra de casos.

En la mineración de datos, el conglomerado por K-medias es un método de conglomeración que tiene como objetivo particionar n observaciones dentro de k conglomerados donde cada observación pertenece al conglomerado más

próximo de la media. Esto resulta en una división del espacio de los datos en un diagrama de varonoi.

El problema es NP complejo (tiempo polinomial no determinista), pero existen algoritmos heurísticos eficientes que son comúnmente empleados y convergen rápidamente hacia un óptimo local. Estos son generalmente semejantes al algoritmo de maximización de expectativa para mesclas de distribuciones gaussianas por medio de un abordaje de refinamiento iterativo por ambos algoritmos. Ambos usan los centros de conglomerados para modelas datos, pero la conglomeración K-medias tiende a encontrar conglomerados de extensión espacial comparables mientras el mecanismo de maximización de la expectativa permita tener diferentes formas.

El Algoritmo K-Medias

Sea Del conjunto de datos con n instancias y sea $C_1,\ C_2, \dots ,\ C_k$ los k grupos disjuntos de D. entonces la función error es definida como:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, u(C_i)) \qquad \textbf{(39)}$$

Donde $\mu(C_i)$ es el centroide del conglomerado C_i y $d(x, \mu(C_i))$ denota la distancia entre x y $\mu(C_i)$. Aquí definimos la distancia euclidiana $d(x, y_i)$ como medida de distancia estándar entre dos puntos x, y.

$$x = (x_1, x_2, \dots , x_d) \qquad \textbf{(40)}$$

$$y = (y_1, y_2, \dots , y_d) \qquad \textbf{(41)}$$

$$d(x,y) = \sqrt{\sum_{j=1}^{d} (x_j - y_j)^2} \quad \textbf{(42)}$$

El algoritmo tiene un procedimiento iterativo. Por lo general, primero se atribuye un vector de centroides inicial arbitrario, en la segunda etapa clasifica cada punto para el siguiente conglomerado, en la tercera etapa nuevos centroides son calculados con base en todos los puntos en un conglomerado, finalmente la segunda y tercera etapa son repetidas hasta que el cambio en la iteración es pequeño. El cambio puede ser definido de varias maneras, sea por medio de la medición de distancias entre el centroide y los otros puntos del grupo, o por el porcentual de puntos que cambiaron de grupos entre las iteraciones (Lucchese & Mitra, 1999).

El seudocódigo del algoritmo K-medias convencional es presentado a continuación.

Algoritmo: Algoritmo K-medias convencional

Entrada: Distribución de los n puntos de D entre los k conglomerados

Salida: Distribución de los n puntos de D entre los k conglomerados

Sea C_i es el i-ésimo conglomerado.

$C_1, C_2, \dots, C_k$ = partición inicial de D

Repita

$d_{i,j}$ =distancia entre el caso i y el conglomerado j

Para todo $1 \leq j \leq k$ hacer

$n_i = \arg\min\{d_i : \forall_{i,j}\}$

Atribuir el caso i al conglomerado n_i

Recalcule el centroide de cualquier

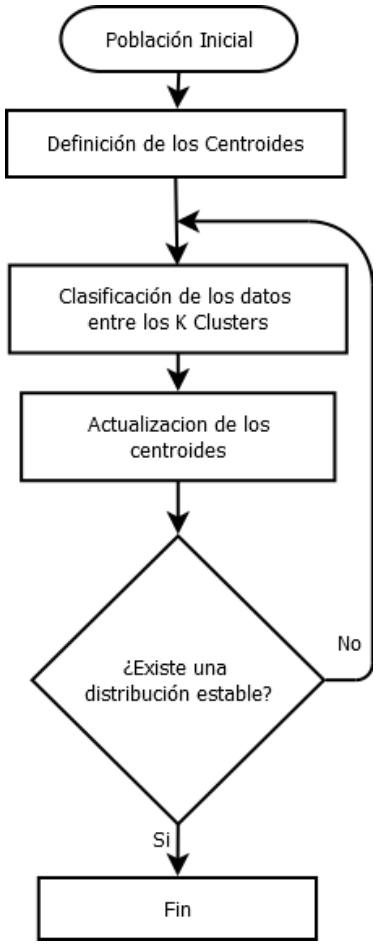
Fin

Hasta ningún centroide cambie de lugar

Retornar salida

El diagrama de flujo de la Figura 34 presenta la secuencia del algoritmo K-medias.

FIGURA 34: Diagrama del K-medias para encontrar conglomerados



FUENTE: Elaboración propia

Las características claves del K-medias, las que lo hacen eficiente vienen a convertirse en su principal problema:

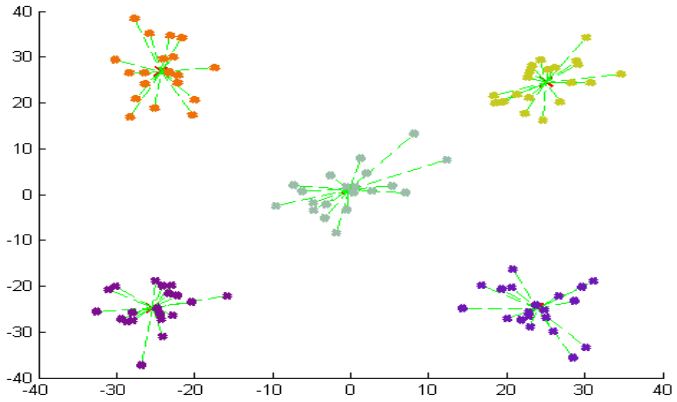
- La distancia euclidiana se usa como una métrica y la varianza es usada como una medida de la dispersión de los grupos.
- El número de grupos k es un parámetro de entrada: una elección inapropiada puede acarrear malos resultados. Por eso es muy importante cuando corremos el K-medias tener en cuenta la importancia de determinar el número de grupos para un conjunto de datos.
- La convergencia a óptimos locales puede traer malos resultados

Una limitación clave del K-medias es su modelo de agrupamiento, el concepto se basa en grupos esféricos que son separables de una forma en que el valor de la media converge hacia el centro del grupo. Se espera que los grupos tengan igual tamaño, por lo que la asignación al grupo más cercano es la asignación correcta.

Por ejemplo si se tiene un conjunto de datos visiblemente agrupables en 5 conglomerados como se puede observar en la Figura 35,

Caso 1: utilizando el algoritmo K-medias con un valor de inicialización de k=5, el resultado es 5 conglomerados como el esperando, este sería el mejor caso donde el algoritmo tiene un buen desempeño. Pero no siempre es así, como se muestra a seguir.

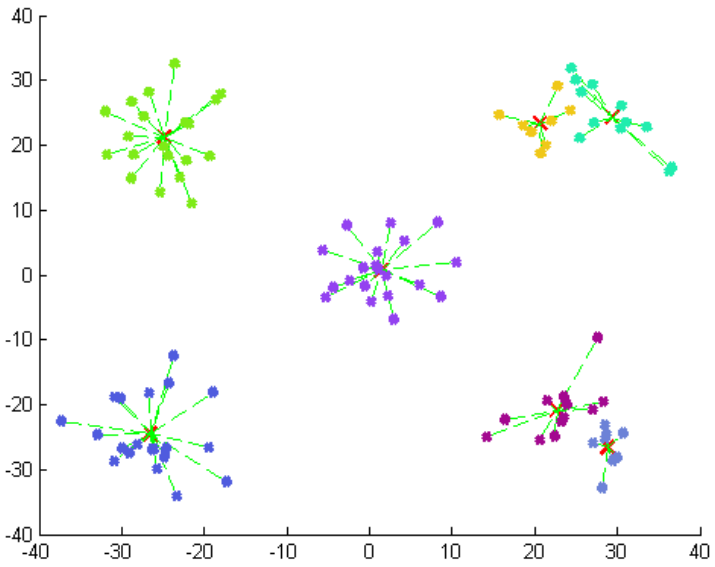
FIGURA 35: Convergencia adecuada del algoritmo K-medias



FUENTE: Elaboración propia

Caso 2: Dado el conjunto de datos de la figura anterior, donde a simple vista es agrupable en 5 grupos, ejecutando el algoritmo con k=7, los cinco grupos visibles no fueron agrupados adecuadamente, esto podría ser debido a que el número de conglomerados iniciales fue k=7, esto es superior al número de conglomerados existentes en la figura, el algoritmo genera grupos donde no era posible observar.

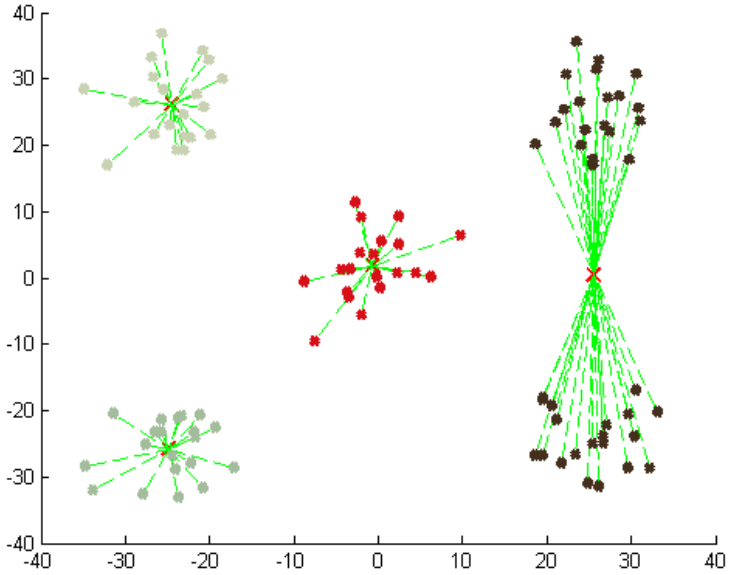
FIGURA 36: Convergencia inadecuada del algoritmo K-mean (7 grupos)



FUENTE: Elaboración propia

Caso 3: Para el mismo ejemplo de 5 grupos visibles, esta vez el algoritmo se encuentra ante una situación donde podemos observar un comportamiento inadecuado debido a que se inicializó con k=4, que es un número inferior a la cantidad de conglomerados existentes como se observa en la Figura 37.

FIGURA 37: Convergencia inadecuada del algoritmo K-medias (4 grupos)



FUENTE: Elaboración propia

ENTROPÍA

El concepto de entropía o incerteza fue introducido por Shannon (1940) para medir la cantidad de información transferida por un canal o generada por una fuente, mientras mayor sea el valor de la entropía mayor será la incerteza, por lo tanto, más información estará asociada al canal (Shah, 1992).

El principio fundamental de la teoría de la información establece que la generación de información puede ser modelada como un modelo probabilístico, una imagen puede ser considerada como el resultado de un proceso aleatorio en el cual la probabilidad de p_i , corresponde a la probabilidad de un pixel en una imagen digital asumir el valor de intensidad i, $i = 0, 1, \dots, L_{max}$.

La distribución de los niveles de intensidad de la imagen puede ser transformada en una función de densidad de probabilidad, dividiéndose el número de píxeles de intensidad i, denotado n_i, por el número total n de pixeles en la imagen, es decir.

$$p_i = \frac{n_i}{n} \tag{43}$$

Dónde:

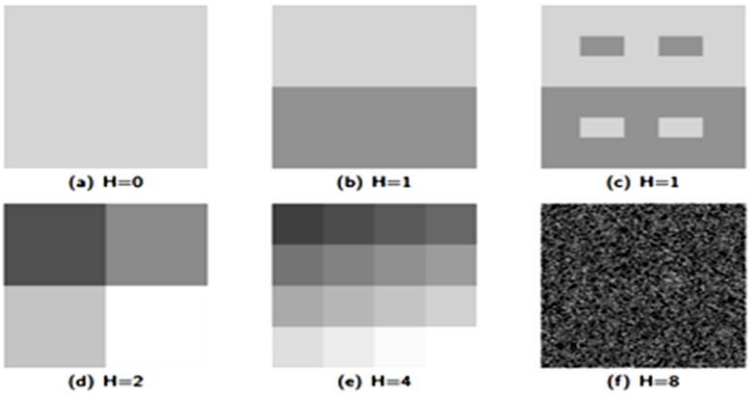
$$\sum_{i=0}^{L_{max}} p_i \log p_i = 1 \tag{44}$$

La entropía H de una imagen puede ser calculada como.

$$H = -\sum_{i=0}^{L_{max}} p_i \log p_i \tag{45}$$

La entropía de una imagen es una medida positiva y cuando la base del logaritmo es dos, la unidad resultante es dada en bits. El menor valor para la entropía es cero, ocurre cuando una imagen contiene la misma cantidad de píxeles para todas las intensidades como se observa en la Figura 38.

FIGURA 38: Imágenes con 256 X 256 píxeles y diferentes entropías



FUENTE: Elaboración propia

Observaciones:

- El valor mínimo y máximo de entropía son observados en las figuras (a) y (f), respectivamente.
- La imagen de la figura (f) posee todo los 256 niveles de gris posibles distribuido con la misma cantidad de píxeles..
- Los valores intermediarios de entropía son presentados en las figuras (b) y (e).
- La entropía no está relacionada con la disposición espacial de la información, como se puede evidenciar en las figuras (b) y (c), donde las dos imágenes poseen la misma cantidad de píxeles con las mismas intensidades, pero distribuidos espacialmente de forma diferente.

2.2.10 OPERACIONALIZACIÓN DE VARIABLES

La tarea de diseñar o elegir una medida apropiada de la efectividad de la segmentación de imágenes satelitales es una tarea en sí compleja. La métrica

de evaluación debería proporcionar información relevante sobre la imagen tratada, como por ejemplo si es diagnóstica o intervencionista. Sin embargo, en todo tipo de tareas en las cuales esté presente la segmentación de imágenes, hay consenso sobre la existencia de tres tipos de métricas que deben evaluarse, y estas son las que detallaremos en esta sección, a saber: exactitud, estabilidad y eficiencia (Chiu., 2005)

Lo característico de una imagen homogénea es que es prácticamente constante, sin cambio sustancial de colores ni de formas. Tanto en (Xian-Sheng Hua L. L.-J., 2004) como en (Xian-Sheng Hua L. L.-J., 2003) y (Xian-Sheng Hua L. L.-J., 2006) se usa la entropía (medida del desorden) para determinar hasta qué punto es invariante la imagen, la operacionalizacion de variable para el presente trabajo se muestra en la Tabla 9.

TABLA 9: Operacionalización de variables

Variable	Dimensión	Indicador	Valores	Naturaleza y escala	Instrumento
Var Ind. Algoritmo de Segmentación de Imágenes	Algoritmo de segmentación en imágenes satelitales	**Tipo de técnica** Algoritmo estándar K-Medias Algoritmo Genético K-medias (AGKM)	0 y 1	cualitativa nominal	hoja de registro
Var Dep. Eficacia en la segmentación de Imágenes	Calidad de segmentación de imágenes satelitales	Entropía (homogeneidad)	Media de bits por pixel	Cuantitativa	hoja de registro

FUENTE: Elaboración propia

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 LUGAR DE ESTUDIO

La región Puno está ubicada al extremo sur este del Perú. Su superficie es de 71 999 km2 (6 por ciento del territorio nacional), siendo el quinto departamento más grande en el ámbito nacional. Limita por el norte con el departamento de Madre de Dios, por el este con la República de Bolivia, por el sur con el departamento de Tacna y la República de Bolivia y por el oeste con los departamentos de Moquegua, Arequipa y Cusco. El 61% del territorio puneño corresponde a la zona sierra, el 32% a selva y el resto es la parte peruana del Lago Titicaca.

Políticamente, está dividido en 13 provincias y 109 distritos. Las provincias son Puno, Azángaro, Carabaya, Chucuito, El Collao, Huancané, Lampa, Melgar, Moho, San Antonio de Putina, San Román, Sandia y Yunguyo.

Según el último Censo del año 2007, Puno se ubica en quinto lugar en cuanto a tamaño de población en el país, con 1 millón 268 mil 441 habitantes, lo que representa el 4,6% de la población nacional. La densidad poblacional es de 18,9 habitantes por km2. Al interior, la provincia de Yunguyo es la más densamente poblada con 164,4 habitantes por km2. La región, tiene un ligero

predominio de población rural, llegando a representar el 50,3% del total. Sin embargo, se mantiene un dinámico proceso de urbanización.

3.2 POBLACIÓN

La población en estudio está definida por todas las imágenes satelitales tomadas con el satélite landsat-7, estas imágenes son descargadas del catálogo de imágenes INPE - Instituto Nacional de Investigaciones Espaciales (Brasil). Para el presente trabajo se utilizará imágenes que corresponden a la región de Puno, estas imágenes ya vienen catalogadas en cuadrantes.

3.3 MUESTRA

El tipo de muestreo utilizado para el presente trabajo es el comparación para dos medias, se consideran las hipótesis nula y alternativa a ser verificadas, siendo $H_0: \mu_1 = \mu_2$, y $H_1: \mu_1 \neq \mu_2$. Si se quisiera calcular el tamaño de la muestra, es necesario conocer:

- Magnitud de la diferencia a detectar que tenga interés relevante.
- Tener una idea aproximada de los parámetros de la variable que se estudia (a través de la bibliografía de estudios previos o realizar una prueba piloto), para el presente trabajo se optó por la segunda opción, realizar una prueba piloto.
- Seguridad del estudio (riesgo de cometer un error α);
- Potencia estadística 1−β (riesgo de cometer un error β);

- Definición de si la hipótesis va a ser unilateral o bilateral, para nuestro caso es bilateral por ser la más conservadora para dar una conclusión.

3.4 MUESTRA PILOTO

Para el presente trabajo no se encontró estudios previos sobre el tipo de prueba estadística para verificar la eficiencia de los algoritmos estudiados, es por este motivo que se decidió realizar una prueba piloto, la muestra piloto está conformada por las imágenes y sus respectivas entropías para cada una de 9 muestras por cada algoritmo, apartir de la Tabla 10 se realizó los cálculos estadísticos necesarios con el nivel de significancia $\alpha = 0.05$.

TABLA 10: Datos de la muestra piloto

N	Algoritmo	Entropía en imagen original	Entropía en imagen tratada
1	K-media	7.42	3.12
2	K-media	7.26	4.78
3	K-media	7.45	5.56
4	K-media	6.55	5.89
5	K-media	7.35	6.33
6	K-media	7.00	3.01
7	K-media	6.93	5.30
8	K-media	6.78	5.70
9	K-media	7.23	4.03
10	AGKM	7.35	2.50
11	AGKM	7.00	3.79
12	AGKM	6.93	3.87
13	AGKM	6.78	3.00
14	AGKM	7.23	3.55
15	AGKM	6.57	2.65
16	AGKM	6.34	3.46
17	AGKM	6.55	4.34
18	AGKM	5.99	2.33

FUENTE: Elaboración propia

Elección de la prueba estadística

Como el estudio es del tipo transversal para muestras independientes y con una variable aleatoria del tipo numérica, implica que se utilizará una prueba paramétrica para dos grupos que es la prueba de T-Student de acuerdo al ANEXO 03.

Cálculo del P-valor para la muestra piloto.

Para calcular el P-valor primero se debe corroborar los dos supuestos con los que trabaja la prueba T de Student:

Normalidad, se corroboró que la variable aleatoria en ambos grupos se distribuye normalmente. Para ello se utilizó la prueba de Kolmogorov-Smirnov para las muestras (n>30), o la prueba de Shapiro-Wilk cuando el tamaño de la muestra es (n≤30). La prueba de hipótesis planteada para determinar la normalidad es:

H_0: *P-valor ≥ α entonces Aceptar* H_0 *= Los datos provienen de una distribución normal.*

H_1: *P-valor < α entonces Aceptar* H_1 *= Los datos no provienen de una distribución normal.*

Para calcular la normalidad se utilizó el software SPSS V21, procesando los datos se obtuvo como resultado la Tabla 11, como nuestra muestra piloto es n ≤ 30 entonces utilzaremos el resultado para la prueba de Shapiro-Wilk que es 0.266, este resultado se contrasta con α y su interpretacón como se presenta en la Tabla 12.

TABLA 11: Prueba de normalidad para el muestreo piloto

Algoritmo		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
		Estadístico	gl	Sig.	Estadístico	gl	Sig.
Entropia	Segmentación Kmedia	.198	9	,200*	.902	9	**.266**
	Segmentación AGKmedia	.159	9	,200*	.949	9	**.682**

*. Este es un límite inferior de la significación verdadera.

a. Corrección de la significación de Lilliefors

FUENTE: Elaboración propia

TABLA 12: Resultado de la prueba de normalidad para la muestra piloto

NORMALIDAD		
P-valor para K-Medias = 0.266	>	$\alpha = 0.05$
P-valor para AGKM = 0.682	>	$\alpha = 0.05$
CONCLUSION: Entonces dado que el P-valor es mayor que el nivel de significancia α, entonces aceptamos la H_0, por lo tanto podemos afirmar que los datos provienen de una distribución normal.		

FUENTE: Elaboración propia

Igualdad de varianza (prueba de Levene), también se corroboró la igualdad de varianza entre los grupos, el resultado del cálculo se presenta en la Tabla 13 y se interpretó de acuerdo al siguiente criterio.

Si P-valor ≥ α entonces Aceptar H_0 *= Las varianzas son iguales.*

Si P-valor < α entonces Aceptar H_1 *= Existe diferencia significativa entre las varianzas.*

TABLA 13: Prueba para la varianza del muestreo piloto

Prueba de muestras independientes

		Prueba de Levene para la igualdad de varianzas		Prueba T para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Error típ. de la diferencia
Entropia	Se han asumido varianzas iguales	3.521	.079	3.401	16	.004	1.583	0.465
	No se han asumido varianzas iguales			3.401	12.678	.005	1.583	0.465

FUENTE: Elaboración propia

TABLA 14: Resultado sobre la igualdad de varianzas para la muestra piloto

IGUALDAD DE VARIANZA		
P-valor = 0.079	>	$\alpha = 0.05$
CONCLUSION: Por lo tanto podemos asumir que las varianzas son iguales		

FUENTE: Elaboración propia

Luego de realizar los cálculos correspondientes con el software SPSS V21, se obtuvo las medias y varianzas correspondientes para cada algoritmo; $\bar{x}_{K-medias} = 4.858$ $s^2_{k\_medias} = 1.473$ y $\bar{x}_{AGKM} = 3.275$ $s^2_{k\_medias} = 0.476$, dando una diferencia de medias $d = 1.583$.

3.5 CÁLCULO DE LA MUESTRA

Para estimar el tamaño de muestra se utilizó la Ecuación (46), esto es aplicado cuando se desea estimar el tamaño de la muestra con base en los errores tipo I y del tipo II, los valores para estos errores fueron obtenidos del ANEXO 02, y reemplazada en la Ecuación (47).

$$n = \frac{2 * (Z_{1-\frac{\alpha}{2}} + Z_{1-\beta})^2 * S^2}{d^2} \quad \textbf{(47)}$$

- n, Número de individuos necesarios en cada una de las muestras n=?.
- $Z_{1-\frac{\alpha}{2}}$, Valor Z del nivel de confianza (para dos colas) α = 0.05.
- $Z_{1-\beta}$, Valor Z para un valor de poder (β = 20) 1- β = 80.
- S^2, Varianza del grupo control o referencia = 1.213624^2 =1.473.
- d, Valor mínimo de la diferencia que se desea detectar (diferencia entre las medias). 1.58.

Substituyendo los valores en la Ecuación (47) de la muestra se obtiene.

$$n = \frac{2 * (1.96 + 0.842)^2 * 1.213624^2}{1.3^2} = 13.7$$

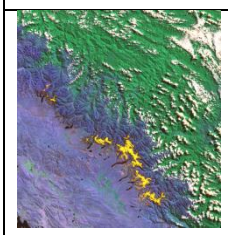
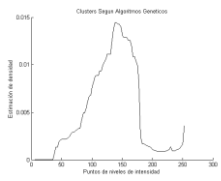
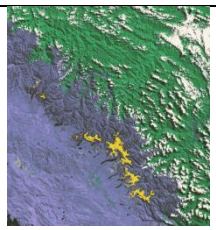
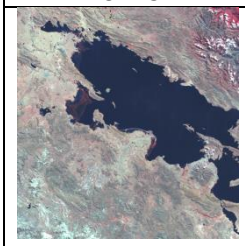
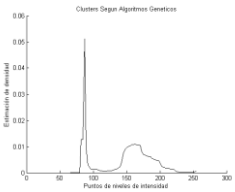
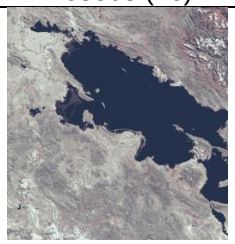
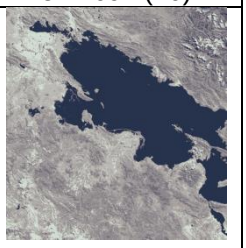
Por lo tanto el tamaño de muestra necesario para el estudio fue de $n = 13.7 \approx 14$ imágenes por cada algoritmo.

Como la muestra fue n = 14, implica que debe muestrearse 14 imágenes por cada algoritmo, estas imágenes son procesadas y ordenadas como se observa en la Figura 39. De izquierda a derecha, la primera columna presenta la imagen original y debajo de la imagen en valor de la entropía encontrada, la segunda columna es el resultado de los Algoritmos Genéticos que muestra una curva que permite observar la cantidad de conglomerados en los que se puede dividir la imagen que a su vez es usado como valor de inicialización del algoritmo

K-medias, el resultado final del procesamiento fue obtenido y ubicado en la última columna que presenta la imagen segmentada y su respectivo valor de entropía.

Finalmente la tercera columna presenta la imagen procesada con el algoritmo K-medias tradicional, con valor de inicialización obtenida previa inspección visual de la imagen satelital, presenta como resultado la imagen segmentada y su respectivo valor de entropía.

FIGURA 39: Imágenes tratadas con AGKM y K-medias

Imagen original	Conglomerados por Algoritmos Genéticos	Segmentación K-Medias	Segmentación AGKM
			
7.54134		4.63503 (16)	5.17694 (10)
			
7.10686		4.39168 (10)	3.7476 (6)

FUENTE: Elaboración propia

A partir de la Figura 39 se genera la Tabla 15 que permite observar las 14 muestras por algoritmo con sus respectivas entropías.

TABLA 15: Entropía por cada imagen satelital según el tipo de algoritmo

N	Algoritmo	Entropía en imagen original	Entropía en imagen tratada	N	Algoritmo	Entropía en imagen original	Entropía en imagen tratada
1	Seg. Kmedia	7.42	4.12	*15*	Seg. AGKM	7.35	4.00
2	Seg. Kmedia	7.26	3.78	*16*	Seg. AGKM	7.00	3.79
3	Seg. Kmedia	7.45	5.56	*17*	Seg. AGKM	6.93	3.87
4	Seg. Kmedia	6.55	5.89	*18*	Seg. AGKM	6.78	5.00
5	Seg. Kmedia	7.35	5.33	*19*	Seg. AGKM	7.00	2.33
6	Seg. Kmedia	7.00	3.01	*20*	Seg. AGKM	6.50	3.88
7	Seg. Kmedia	6.93	4.30	*21*	Seg. AGKM	6.88	4.88
8	Seg. Kmedia	6.78	5.00	*22*	Seg. AGKM	7.09	4.56
9	Seg. Kmedia	7.00	4.38	*23*	Seg. AGKM	6.80	3.45
10	Seg. Kmedia	6.50	3.98	*24*	Seg. AGKM	6.54	4.65
11	Seg. Kmedia	6.88	4.99	*25*	Seg. AGKM	6.43	3.24
12	Seg. Kmedia	7.09	4.78	*26*	Seg. AGKM	7.23	3.55
13	Seg. Kmedia	6.80	5.67	*27*	Seg. AGKM	6.57	3.45
14	Seg. Kmedia	7.34	4.84	*28*	Seg. AGKM	6.34	3.46
	...	...	...				

FUENTE: Elaboración propia

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

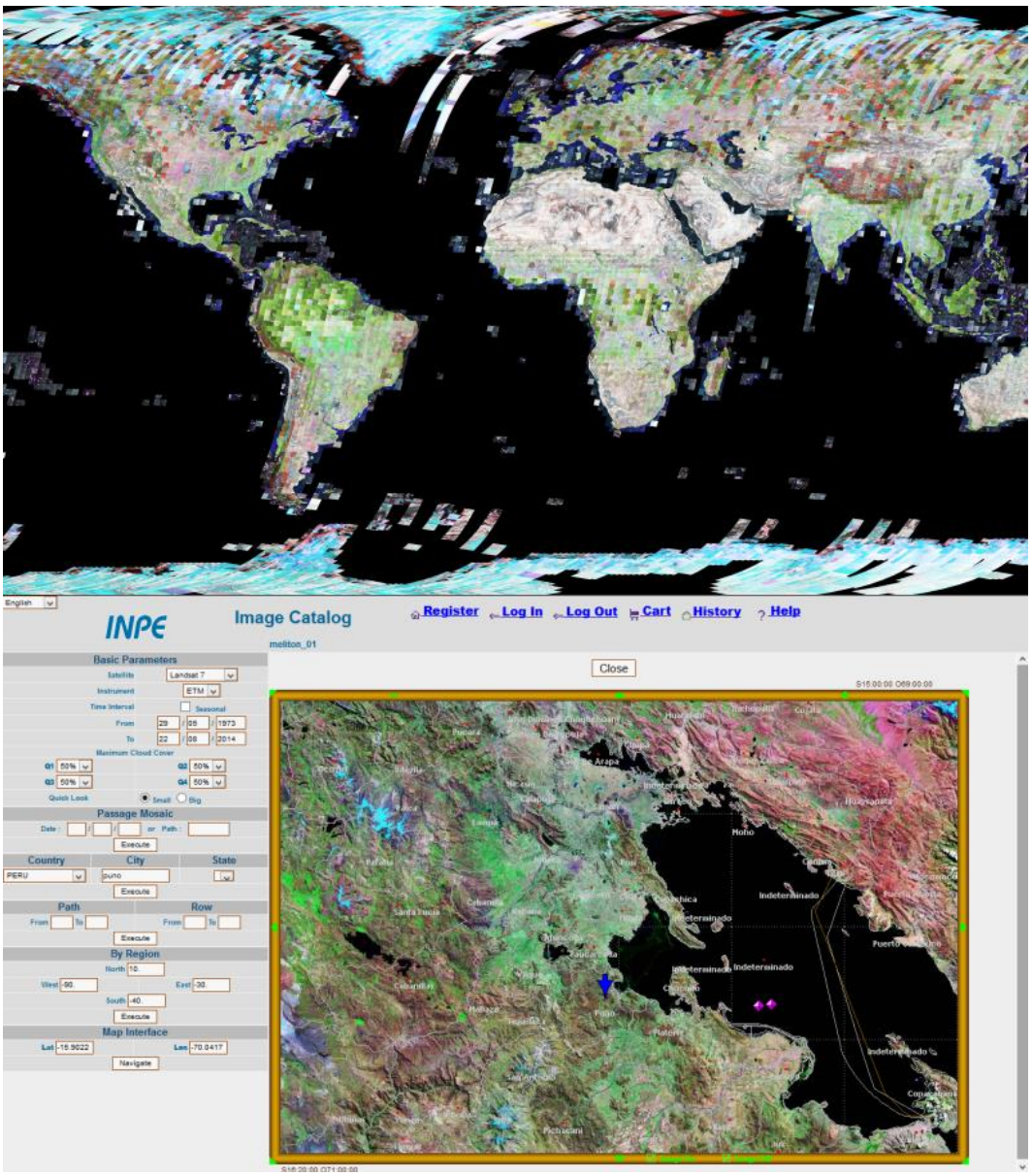
4.1 MÉTODO DE RECOPILACIÓN DE DATOS

La fuente de donde se descargaron las imágenes satelitales es proveída de forma gratuita por el INPE - Instituto Nacional de Investigaciones Espaciales (Brasil). Con la unión de recursos financieros y tecnológicos entre Brasil y China, con una inversión superior a US$ 300 millones, fue creado un sistema de responsabilidades divididas (30% brasileño y 70% chino), teniendo como objetivo la implantación de un sistema completo de sensoramiento remoto a nivel internacional. La unión entre los dos países es un esfuerzo bilateral para derrumbar las barreras que impiden el desenvolvimiento y la transferencia de tecnologías sensibles impuestas por los países desarrollados. El acuerdo conjunto rompió los estándares que restringían los acuerdos internacionales a la transferencia de tecnología y el intercambio entre investigadores de diferentes nacionalidades.

El catálogo busca ofrecer al usuario, facilidades para la obtención de imágenes por medio de criterios objetivos de selección y mecanismos simples y eficientes de acceso y download como muestra la Figura 35. Para sistemas semiautomáticos donde el usuario interactúa con la interfaz dirigiendo su búsqueda, en tiempo real, las operaciones solicitadas. Este sistema está basado

en una interfaz Web, accesible en www.dgi.inpe.br/CDSR, proyectada para una operación simple y de fácil comprensión por el usuario. El catálogo de imágenes de la DGI/INPE fue íntegramente concebido y desarrollado por la División de Procesamiento de Imágenes (DPI) conjuntamente con la División de Generación de Imágenes (DGI) del INPE.

FIGURA 40: Catalogo de imágenes INPE



FUENTE: INPE

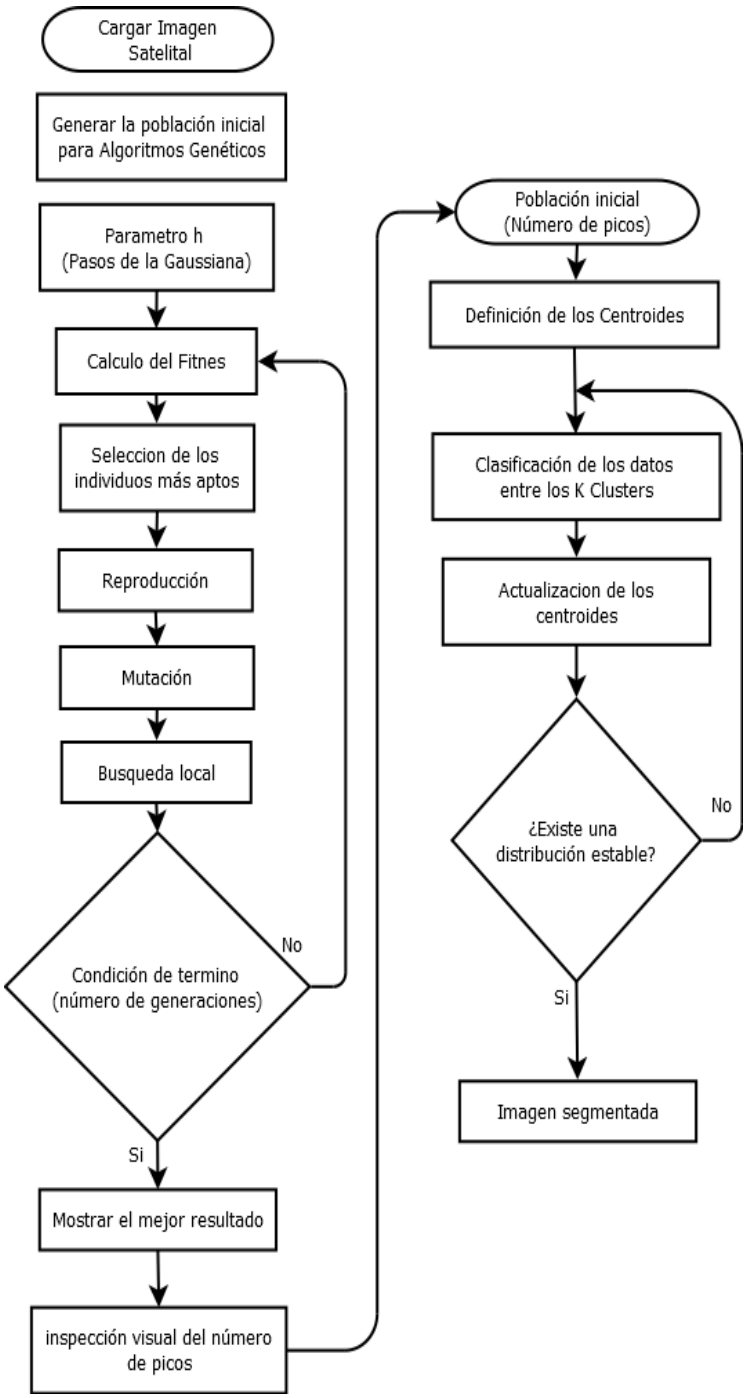
4.2 MÉTODO DE TRATAMIENTO DE DATOS

El tratamiento de las imágenes satelitales comprende dos pasos, el primero trata las imágenes mediante el software AGKM implementado en lenguaje Matlab. Este software tiene implementado los algoritmos K-medias y AGKM, a partir del procesamiento con este software es que se obtiene la entropía de cada imagen las que fueron almacenadas en una tabla y posteriormente evaluadas estadísticamente con asistencia del SPSS para probar la eficacia del algoritmo.

4.3 ALGORITMO AGKM (ALGORITMOS GENÉTICOS K-MEDIAS)

El algoritmo propuesto es el AGKM compuesto de dos partes, la primera parte se encarga de buscar el número ideal de conglomerados en la imagen satelital por medio de los Algoritmos Genéticos (valor de inicialización para K-medias), la segunda parte comprende el algoritmo K-medias que utiliza como entrada el número de conglomerados encontrados por los algoritmos genéticos, como se puede apreciar en el diagrama presentado en la Figura 41.

FIGURA 41: Diagrama del AGKM para segmentar imágenes



FUENTE: Elaboración propia

A seguir se presenta las etapas que se sigue en la ejecución del programa implementado para el presente trabajo.

Algoritmo: Algoritmos Genéticos K-medias - AGKM

Paso 1: Cargar y visualizar la imagen satelital.

Paso 2: Transformar la imagen a escala de grises.

Paso 3: Definir la ventana de función gaussiana.

Paso 4: Inicializar los algoritmos genéticos con el resto de parámetros restantes previamente establecidos en el código.

Paso 5: Visualizar el número de picos encontrados con los algoritmos genéticos.

Paso 6: Ingresar el número de conglomerados encontrados como parámetro inicial en el algoritmo K-medias.

Paso 7: Inicializar el algoritmo K-medias con el parámetro inicial del paso 6.

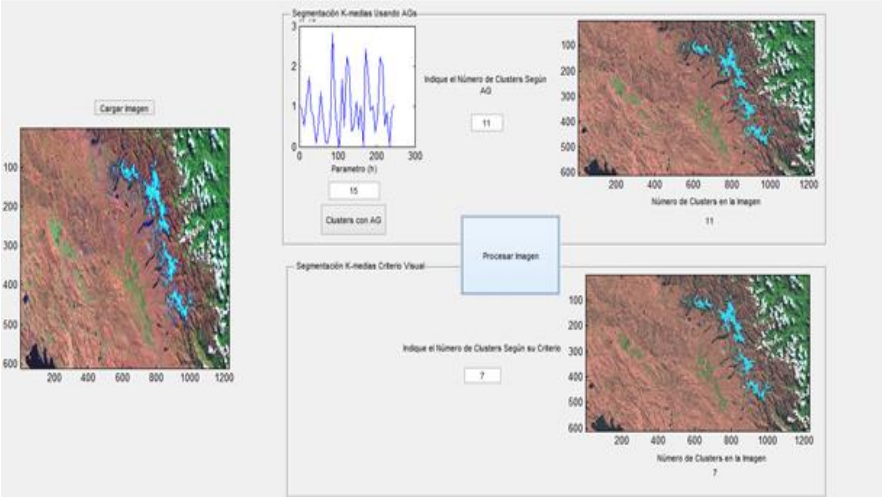
Paso 8: Paralelamente al paso 7 se ingresa el valor conveniente en el algoritmo K-medias, de acuerdo a la inspección visual de la imagen satelital.

Paso 9: Generar las imágenes segmentadas y almacenar.

Paso10: Compararar las imágenes procesadas con ambos métodos.

La Interfaz principal del algoritmo AGKM implementado en Matlab tiene la apariencia mostrada en la Figura 42. En la parte izquierda se carga la imagen a ser tratada, para después en la parte superior derecha se extrae el número de conglomerados existentes en la imagen con los algoritmos genéticos, esto se usará como valor de inicialización en el algoritmo AGKM, la parte inferior también usa un valor de entrada que se asignará de acuerdo a la observación viasual, finalmente el resultado se muestra en la parte derecha como dos imágenes. La imagen resultante del tratamiento con el AGKM presentado en la parte superior del extremo derecho y la imagen tratada con el algoritmo K-medias en parte del extremo derecho inferior, el código fuente de esta implementación se puede encontrar en el ANEXO 04 y ANEXO 05.

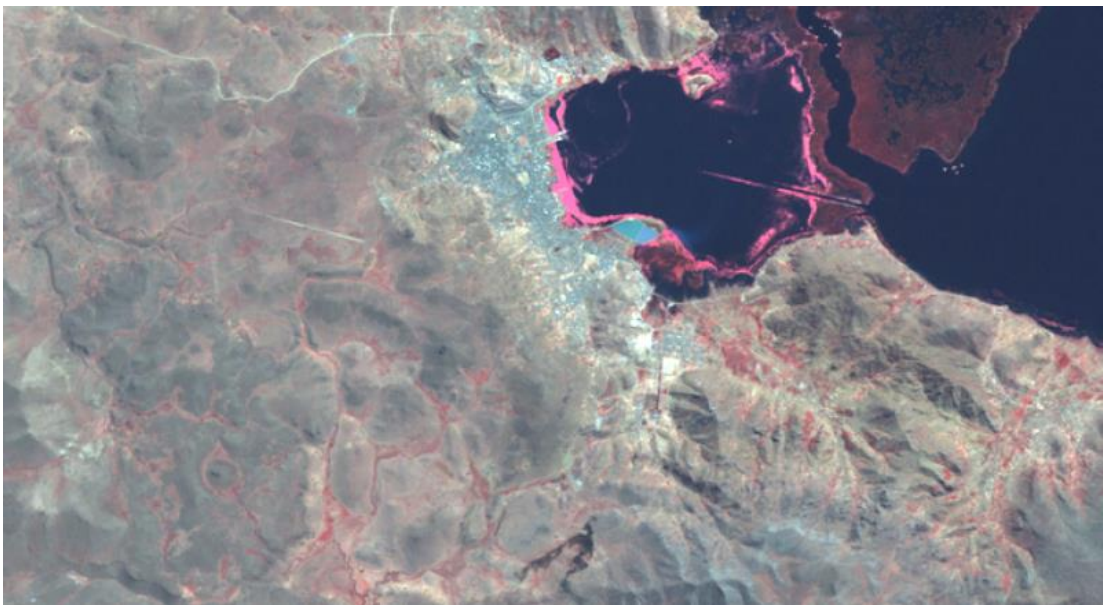
FIGURA 42: Máscara de la aplicación AGKM



FUENTE: Elaboración propia

Caso 1: Para este caso se utilizó una imagen satelital de la ciudad de puno, esta imagen está compuesta por un falso RGB (bandas 4-3-2), en la Figura 43 se aprecia la imagen satelital con algunas características que resaltan a simple vista como la parte del lago afectado por las lentejas de agua, área urbana, es posible interpretarla según características descritas en el Capítulo II.

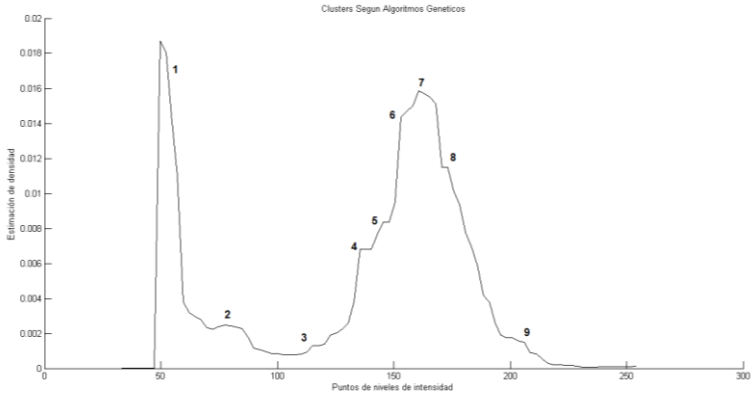
FIGURA 43: Imagen satelital de la ciudad de puno y alrededores



FUENTE: Elaboración propia

Una vez que se cargada la imagen se procedió a fijar el tamaño de la ventana de gauss con h = 1 y con este valor procedemos a explorar la imagen con los algoritmos genéticos que nos proporcionó como resultado la curva presentada en la Figura 44, esta curva muestra los 9 conglomerados como son enumerados, este es el valor de inicialización al algoritmo K-medias.

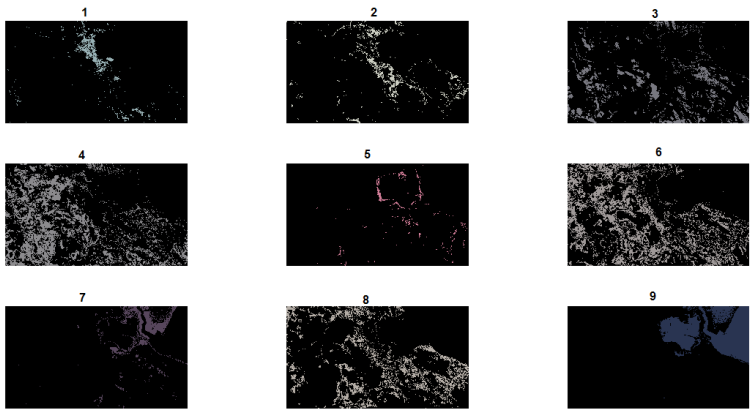
FIGURA 44: Número de conglomerados encontrados



FUENTE: Elaboración propia

Posterior a la ejecución del algoritmo AGKM se obtuvo como resultado la Figura 45 donde es posible apreciar en la imagen 1 la parte urbana, las imágenes 5 y 7 presentan el área afectada por la contaminación de la lenteja de agua, y finalmente la imagen 9 presenta la parte del lago.

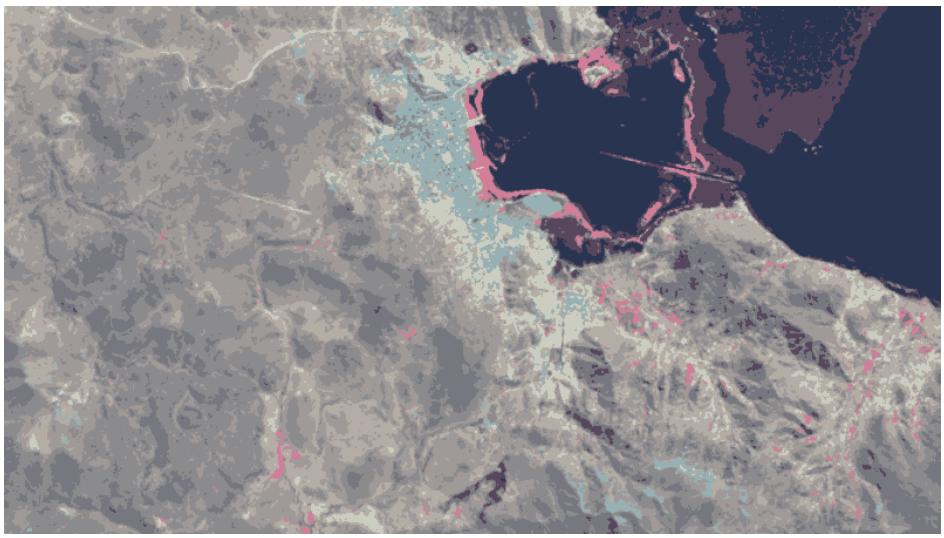
FIGURA 45: Representación de los 9 conglomerados encontrados



FUENTE: Elaboración propia

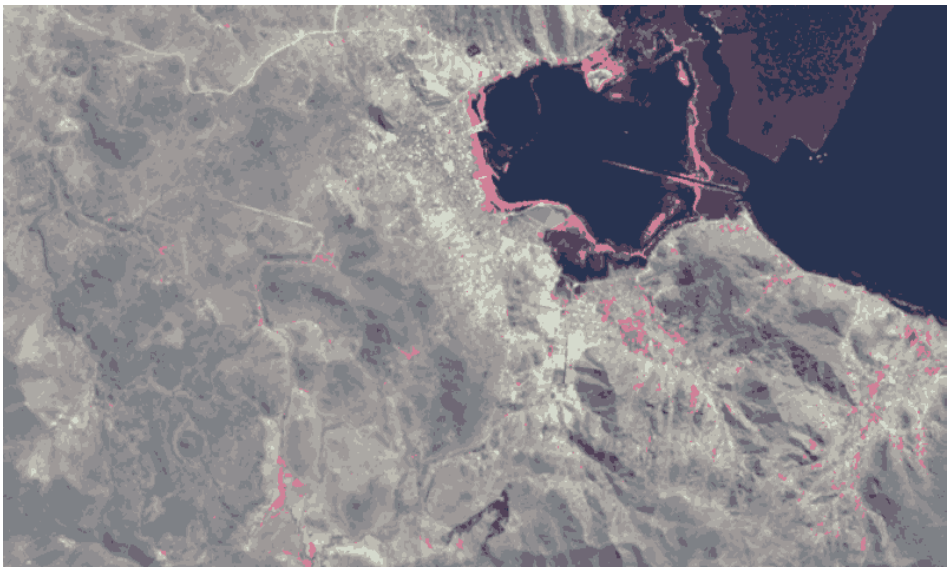
La imagen original mostrada en la Figura 43 tiene entropía 7.01952, al ser tratada con el algoritmo AGKM se muestra el resultado presentado en la Figura 46 con 9 conglomerados y entropía 4.21955, el tratamiento con el algoritmo K-medias es presentado en la Figura 47 con 10 conglomerados y entropia 4.5178.

FIGURA 46: Segmentacion utilizando AGKM



FUENTE: Elaboración propia

FIGURA 47: Segmentacion utilizando K-medias



FUENTE: Elaboración propia

4.4 CONTRASTE DE HIPÓTESIS

Luego de haber analizado los indicadores y características de las imágenes satelitales para la región Puno, estamos en condiciones de realizar el contraste de hipótesis planteada al principio de la investigación.

Los Algoritmos Genéticos K-medias (AGKM) obtiene un mejor desempeño en la segmentación de imágenes satelitales de la región Puno – 2013 en comparación con el algoritmo K-medias.

Para poder probar esta hipótesis se utilizó como indicador la entropía para cada imágen satelital de la muestra, la diminución de la entropía indica una homogeneidad encontrada en los pixeles. El contraste de hipótesis para demostrar la existencia de la diferencia de medias se realizó la prueba para dos muestras independientes de T-Sudent esta elección según el ANEXO 03.

1. **Hipótesis:**

$$H_0: \mu_{K-medias} = \mu_{AGKM}$$

$$H_1: \mu_{K-medias} \neq \mu_{AGKM}$$

2. **Nivel de significación:** $\alpha = 0.05$.

3. **Prueba estadística**: T-Student para dos muestras independientes.

$$|t_c| = \frac{\bar{x}_1 - \bar{x}_2 - 0}{\hat{S}_{\bar{x}_1 - \bar{x}_2}} \sim t_{\left((n_1+n_2-2),\frac{\alpha}{2}\right)}$$

4. **Región de rechazo (RR) y región de aceptación (RA):**

$H_0/RA: Si\ |t_c| \leq t_{\left((n_1+n_1-2),\frac{\alpha}{2}\right)}$, se rechaza H_0.

$H_0/RR: Si\ |t_c| > t_{\left((n_1+n_1-2),\frac{\alpha}{2}\right)}$, se acepta la H_a.

5. **Cálculo de la prueba:** Asumiendo que la diferencia de medias es 0. Antes de calcular el P-valor de debe cumplir los supuestos de normalidad e igualdad de varianzas.

Para el supuesto de normalidad se plantea la siguiente hipótesis:

H_0: Si P-valor ≥ α, entonces los datos provienen de una distribución normal.

H_1: Si P-valor < α, entonces los datos no provienen de una distribución normal.

Para calcular el P-valor se utilizó el software SPSS V21, obteniendo como resultado la Tabla 16, como n ≤ 30 entonces se utilizó el resultado para Shapiro-Wilk como se presenta en la Tabla 17

TABLA 16: Prueba de normalidad para el muestreo

Algoritmo		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
		Estadístico	Gl	Sig.	Estadístico	gl	Sig.
Entropía	Segmentación Kmedia	0.115	14	0.200*	0.973	14	**0.915**
	Segmentación AGKmedia	0.138	14	0.200*	0.945	14	**0.482**

*. Este es un límite inferior de la significación verdadera.

a. Corrección de la significación de Lilliefors

FUENTE: Elaboración propia

TABLA 17: Resultado de la prueba de normalidad para la muestra

NORMALIDAD		
P-valor para K-Medias = 0.915	>	$\alpha = 0.05$
P-valor para AGKM = 0.482	>	$\alpha = 0.05$
CONCLUSION: Como se observa que los P-valor son mayores que el nivel de significancia α, entonces aceptamos la H_0, por lo tanto podemos afirmar que los datos provienen de una distribución normal.		

FUENTE: Elaboración propia

Para el supuesto de igualdad de varianzas se plantea la siguiente hipótesis:

H_0: Si P-valor ≥ α, entonces las varianzas son iguales.

H_1: Si P-valor < α, entonces existe diferencia entre las varianzas.

Para calcular el P-valor para esta prueba se utilizó el software SPSS V21, obteniendo como resultado la Tabla 18, como se asume que las varianzas son iguales entonces P-valor es 0.523, este valor se utilizó en la interpretación del resultado Tabla 19.

TABLA 18: Prueba de igualdad de varianzas para la muestra

Prueba de muestras independientes

		Prueba de Levene para la igualdad de varianzas		Prueba T para la igualdad de medias				
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Error típ. de la diferencia
Entropía	Se han asumido varianzas iguales	0.419	0.523	2.842	36	0.009	0.824205	0.2900386
	No se han asumido varianzas iguales			2.842	25.681	0.009	0.824205	0.2900386

FUENTE: Elaboración propia

TABLA 19: Resultado sobre la igualdad de varianzas para la muestra

IGUALDAD DE VARIANZA		
P-valor = 0.523	>	$\alpha = 0.05$
CONCLUSION: Observamos que P-valor es mayor, por lo tanto podemos asumir que las varianzas son iguales		

FUENTE: Elaboración propia

$$|t_c| = \frac{\bar{x}_1 - \bar{x}_2 - 0}{\hat{s}_{\bar{x}_1 - \bar{x}_2}} = 2.842$$

Dónde:

$$\hat{s}_{\bar{x}_1 - \bar{x}_2} = \sqrt{\left(\frac{(n_1 - 1)\hat{s}^2_{\bar{x}_1} + (n_2 - 1)\hat{s}^2_{\bar{x}_2}}{n_1 + n_2 - 2}\right)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}$$

$n_1 = n_{K-medias} = 14$

$n_2 = n_{AGKM} = 14$

$\bar{x}_1 = \bar{x}_{K-medias} = 4.688$

$\bar{x}_2 = \bar{x}_{AGKM} = 3.864$

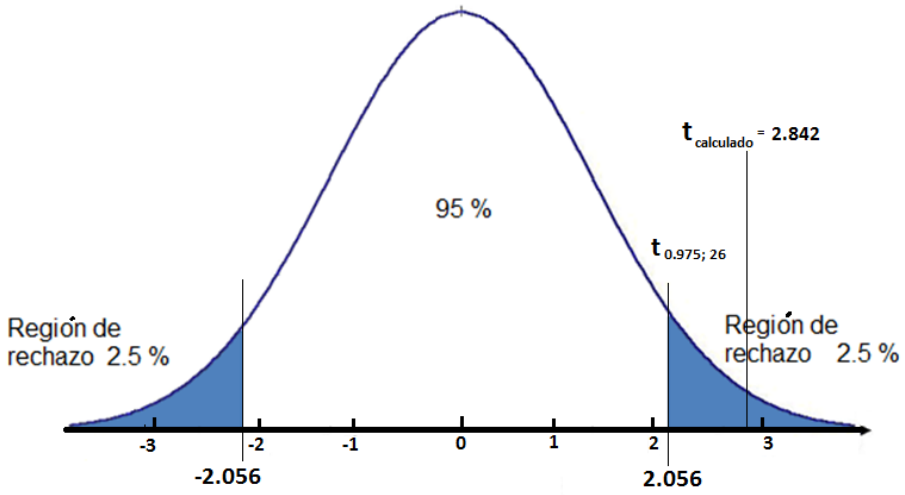
$\hat{s}^2_{\bar{x}_1} = \hat{s}^2_{\bar{x}_{K-medias}} = 0.655$

$\hat{s}^2_{\bar{x}_2} = \hat{s}^2_{\bar{x}_{AGKM}} = 0.523$

6. Decisión:

Se observa en la Figura 48 que $(|t_c| = 2.842) > (t_{(26,0.05)} = 2.056)$ según el ANEXO 01 y ANEXO 02, por lo tanto se acepta la hipótesis alterna que nos dice que existe diferencia $H_1: \mu_{K-medias} \neq \mu_{AGKM}$.

FIGURA 48: Curva de T-Student para la muestra estadística



FUENTE: Elaboración propia

CONCLUSIONES

Según la prueba estadística realizada con 28 muestras de imágenes satelitales, siendo la entropía nuestro parámetro de medición, se observa que el algoritmo AGKM es ligeramente superior al algoritmo K-medias.

Al implementarse el AGKM en Matlab, este permitió encontrar un numero de conglomerados adecuado para utilizar como variable de inicialización en el algoritmo K-medias, y así mejorar el proceso de segmentación en las imágenes satelitales de la región Puno.

Se aplicó la teoría de Algoritmos Genéticos K-medias para la segmentación de imágenes satelitales de la región Puno – 2013, el análisis de conglomerados usando Algoritmos genéticos fue de ayuda en el descubrimiento de la cantidad de conglomerados que se puede encontrar en un conjunto de datos, en este caso imágenes satelitales de la región Puno.

Como con cualquier otro algoritmo de conglomeración, el resultado de K-medias depende del conjunto de datos para satisfacer las necesidades del algoritmo, también influye la configuración inicial del algoritmo como el especificar adecuadamente el número de conglomerados inicial con el que comienza a trabajar. Simplemente trabaja bien en algunos conjuntos de datos mientras que falla en otros. El presente trabajo se centra en encontrar la cantidad ideal de conglomerados que puede existir en un conjunto de datos y que esto sirve como entrada inicial al algoritmo K-medias.

RECOMENDACIONES Y SUGERENCIAS

En este trabajo fueron presentados el algoritmo AGKM que reúne los Algoritmo Genético y el algoritmo K-medias, teniendo como indicador base la entropía que permite medir el nivel de segmentación de la imagen satelital, por lo tanto es necesario un estudio más profundo con relación a otras medidas como: Homogeneidad, disimilaridad, media, desviación estándar, energía, índice de rigurosidad, respecto al tiempo de procesamiento, etc.

El algoritmo AGKM no muestra una contundente mejoría con respecto al algoritmo K-medias, por tal motivo es necesario un estudio más exhaustivo en la parte de los algoritmos genéticos específicamente el kernel debido a que en el trabajo solo usamos la Gausiana, esta función kernel puede ser remplazada por las funciones Triangular, Espanechnicov, etc., para verificar y comparar la eficiencia del método.

Efectuar alteraciones en los algoritmos genéticos propuesto, por ejemplo, establecer múltiples puntos de cruce, teniendo como objetivo el mejor desempeño de los Algoritmos Genéticos.

BIBLIOGRAFÍA

Beasley, D. B. (1993). An Overview of Genetic Algorithms: Part 1, Fundamentals. *University Computing*, págs. 58-69.

Bow, S.-T. (1984). *Pattern Recognition: Applications to Large Data-Set Problems.* Marcel Dekker Incorporated, 1984.

Canny, J. (Nov de 1986). A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, págs. 679-698.

Chisholm, K. B. (1997). Machine learning using a genetic algorithm to optimise a draughts program board evaluation function. *Evolutionary Computation, 1997., IEEE International Conference on*, (págs. 715-720).

Chiu., A. F. (2005). Evaluation of segmentation algorithms for medical imaging. *In Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, (págs. 7186–7189).

Claudio Delrieux, G. R. (2001). Un Sistema Amigable para el Procesamiento de Imagenes Satelitales. *Jornadas Argentinas de Informática e Investigación Operativa* (págs. 99-106). Buenos Aires: JAIIO.

Coto, E. (2003). Métodos de Segmentación de Imágenes Médicas. *Lecturas en Ciencias de la Computación UCV*.

Dash, R. D. (2012). Comparative analysis of K-means and Genetic algorithm based data clustering. *International journal of Advanced Computer an mathematical science*, págs. 257-265.

Dit-Yan Yeung and Chow, C. (2002). Parzen-window network intrusion detectors. *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, (págs. 385-388 vol.4).

Ebecken, E. R. (2000). Applying a clustering genetic algorithm for extracting rules from a supervised neural network. *Neural Networks, IEEE - INNS - ENNS International Joint Conference on*, 3:3407.

Gadhoc, P. R. (2005). Um algoritmo genético de particionamento k-clustering em redes ad hoc. *XXXVII Simpósio Brasileiro de pesquisa Operacional*.

García L., P., & Sancho G, J. (2010). Estimación de densidad de probabilidad mediante ventanas de Parzen. *2010 III Jornadas de Introducción a la Investigación de la UPCT.* Universidad Politécnica de Cartagena.

Goldberg, D. E. (1989). *Genetic algorithm in search, optimization, and machine learning. Proc.* Addison Wesley.

Hans-Peter Kriegel, P. K. (March 2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1):1:1–1:58.

Hartigan, J. A., & Wong. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society, Series C* , 100–108.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* Cambridge, MA, USA: MIT Press.

Izidoro, S. C. (2008). *Determinação do Número de Agrupamentos em Conjuntos de Dados.* Rio Verde-Brasil.

Keren, D. (2004). *Advanced Topics in Computer Vision: 3D vision*. Recuperado el 01 de 09 de 2014, de Advanced Topics in Computer Vision: 3D vision: http://www.cs.haifa.ac.il/~dkeren/acv/optical-flow.pdf

Kiefer, T. .. (2000). *Remote Sensing and Image Interpretation.* New York: Willey & Sons.

Kriegel, H.-P. a. (2005). A generic framework for efficient subspace clustering of high-dimensional data. *Data Mining, Fifth IEEE International Conference on*, 8 pp.

Kudova, P. (2007). Clustering Genetic Algorithm. *23rd International Workshop on Database and Expert Systems Applications*, 0:138–142.

Landsat, N. (2003). *Global Land Cover Facility*. Recuperado el 2014, de http://glcf.umd.edu/data/landsat/

Llorente, I. A.-C. (2010). *El Satelite Landsat analisis visual de imagenes obtenidas del sensor ETM + satelite landsat.* valladolid: universidad de valladolid.

Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, págs. Vol 28, 129-137.

Lucchese, L., & Mitra, S. (1999). Unsupervised segmentation of color images based on k-means clustering in the chromaticity plane. *Content-Based*

Access of Image and Video Libraries, 1999. (CBAIVL '99) Proceedings. IEEE Workshop, págs. 74,78.

Macq, M. M. (2005). *Segmentation using a region-growing thresholding.* Proceedings of the SPIE 5672 (2005) 388–398.

Man, K., Tang, K., & Kwong, S. (1996). Genetic algorithms: concepts and applications [in engineering design]. *Industrial Electronics, IEEE Transactions on*, (págs. 519,534).

Maragathavalli, P. a. (2012). Automatic program instrumentation in generation of test data using genetic algorithm for multiple paths coverage. *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on*, (págs. 349-353).

Martin D., F. C. (July 2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proc. *8th Int'l Conf. Computer Vision, volume 2*, (págs. 416–423).

Maulik., U. (2009). Medical image segmentation using genetic algorithms. *Information Technology in Biomedicine, IEEE Transactions on*, págs. 13(2):166–173.

Mendes Filho, E. F. (1998). Algorithm genéticos. *www.icms.sc.usp.brp/~prico/index.html*.

Merzougui M, M. N. (2013). Image Segmentation using Isodata Clustering with Parameters Estimated by Evolutionary Approach. *Application to Quality Control. International Journal of Computer Applications*, (págs. 25-30).

Norvig, S. J. (1995). Artificial Intelligence: A Modern Approach. Prentice Hall.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *JSTOR: The Annals of Mathematical Statistics, Vol. 33, No. 3 (Sep., 1962), pp. 1065-1076* (pág. 3). Institute of Mathematical Statistics.

Pinto, J. O. (1998). *Cluster analysis using GA.* Proc. University of Tenessee.

Prabha, R. S. (2011). Refinement of k-means clustering using genetic algorithm. *Journal of Computer Applications (JCA), IV(2)*.

Purushothaman S., S. T. (2012). Segmentation of satellite images using fuzzy logic and hilbert huang transform. *International Journal of Engineering Research and Applications (IJERA)*, págs. 1020–1023.

Schoenmakers R. P H M, W. G. (1993). Use of landsat tm im- age segmentation for smoothing ers-1 sar imagery in combined multi-sensor landscape classification. *Geoscience and Remote Sensing Symposium, 1993. IGARSS '93. Better Understanding of Earth Environment., International*, (págs. 1228–1230 vol.3).

Schowengerdt, R. (1997). *Remote Sensing Models and Methods for Image Processing.* San Diego: Academic Press.

Serrada, A. P. (1996). Uma introducción a la computación evolutiva Proc. *www.geocities.com/igoryepes/.*

Shah, J. (1992). Properties of energy-minimizing segmentations. *SIAM Journal on Control and Optimization*, (págs. 30(1):99–111).

Shapiro, L. G. (2002). *Computer Vision.* Prentice Hall.

Sheikh, R. a. (2008). Genetic Algorithm Based Clustering: A Survey. *Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on*, (págs. 314-319).

Silverman, B. (1998). *Density Estimation for Statistics and Data Analysis.* London: Chapman and Hall.

Taghizadeh, M. a. (2011). A hybrid algorithm for segmentation of MRI images based on edge detection. *Soft Computing and Pattern Recognition (SoCPaR), 2011 International Conference of*, (págs. 107-111).

Tung, D.-C. T.-F.-T. (1995). Circular histogram thresholding for color image segmentation. *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, (págs. 673-676 vol.2).

Wang, Y.-W. Y.-H. (1999). Image segmentation based on region growing and edge detection. *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, (págs. 798-803 vol.6).

Weile, D. S. (Mar de 1997). Genetic algorithm optimization applied to electromagnetics: a review. *Antennas and Propagation, IEEE Transactions on*, págs. 343-353.

Whitley, D. (1994). A genetic algorithm tutorial. *Kluwer Academic Publishers* (págs. 65-85). Statistics and Computing.

Xian-Sheng Hua, L. L.-J. (2003). Content based photograph slide show with incidental music. *In Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, II–648–II–651 vol.2.

Xian-Sheng Hua, L. L.-J. (2004). Automatically converting photographic series into video. *In Proceedings of the 12th annual ACM international conference on Multimedia*, 708–715.

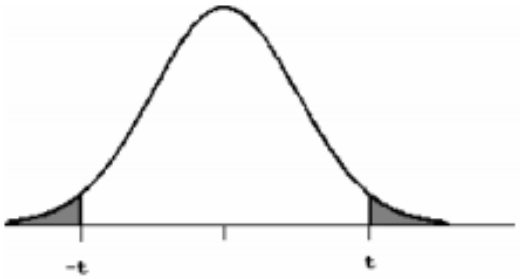
Xian-Sheng Hua, L. L.-J. (2006). Photo2video amp 8212 a system for automatically converting photographic series into video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 803–819.

Yamaoka, K. a. (2006). Image segmentation and pattern matching based FPGA/ASIC implementation architecture of real-time object tracking. *Design Automation, 2006. Asia and South Pacific Conference on*, (págs. 6 pp.-).

Young I.T., J. G. (2005). *Image Processing Fundamentals*. Recuperado el noviembre de 2014, de http://www.mif.vu.lt/atpazinimas/dip/FIP/fip.html

ANEXO

ANEXO 01: TABLA DE T-STUDENT PARA DOS COLAS



(a) El área de las dos colas está sombreada en la figura.
(b) Si H_A es direccional, las cabeceras de las columnas deben ser divididas por 2 cuando se acota el P-valor.

	ÁREA DE DOS COLAS						
gl	0,20	0,10	0,05	0,02	0,01	0,001	0,0001
1	3,078	6,314	12,706	31,821	63,657	636,619	6366,198
2	1,886	2,920	4,303	6,695	9,925	31,598	99,992
3	1,638	2,353	3,182	4,541	5,841	12,924	28,000
4	1,533	2,132	2,776	3,747	4,604	8,610	15,544
5	1,476	2,015	2,571	3,365	4,032	6,869	11,178
6	1,440	1,943	2,447	3,143	3,707	5,959	9,082
7	1,415	1,895	2,365	2,998	3,499	5,408	7,885
8	1,397	1,860	2,306	2,896	3,355	5,041	7,120
9	1,383	1,833	2,262	2,821	3,250	4,781	6,594
10	1,372	1,812	2,228	2,764	3,169	4,587	6,211
11	1,363	1,796	2,201	2,718	3,106	4,437	5,921
12	1,356	1,782	2,179	2,681	3,055	4,318	5,694
13	1,350	1,771	2,160	2,650	3,012	4,221	5,513
14	1,345	1,761	2,145	2,624	2,977	4,140	5,363
15	1,341	1,753	2,131	2,602	2,947	4,073	5,239
16	1,337	1,746	2,120	2,583	2,921	4,015	5,134
17	1,333	1,740	2,110	2,567	2,898	3,965	5,044
18	1,330	1,734	2,101	2,552	2,878	3,922	4,966
19	1,328	1,729	2,093	2,539	2,861	3,883	4,897
20	1,325	1,725	2,086	2,528	2,845	3,850	4,837
21	1,323	1,721	2,080	2,518	2,831	3,819	4,784
22	1,321	1,717	2,074	2,508	2,819	3,792	4,736
23	1,319	1,714	2,069	2,500	2,807	3,767	4,693
24	1,318	1,711	2,064	2,492	2,797	3,745	4,654
25	1,316	1,708	2,060	2,485	2,787	3,725	4,619
26	1,315	1,706	2,056	2,479	2,779	3,707	4,587
27	1,314	1,703	2,052	2,473	2,771	3,690	4,558
28	1,313	1,701	2,048	2,467	2,763	3,674	4,530
29	1,311	1,699	2,045	2,462	2,756	3,659	4,506
30	1,310	1,697	2,042	2,457	2,750	3,646	4,482
40	1,303	1,684	2,021	2,423	2,704	3,551	4,321
60	1,296	1,671	2,000	2,390	2,660	3,460	4,169
100	1,290	1,660	1,984	2,364	2,626	3,390	4,053
140	1,288	1,656	1,977	2,353	2,611	3,361	4,006
∞	1,282	1,645	1,960	2,326	2,576	3,291	3,891

ANEXO 02: TABLA DE VALORES PARA Z_α Y Z_β (PODER Y SEGUIRIDAD)

Nivel de confianza $(1-\alpha)$		
$1-\alpha$	Test unilateral	Test bilateral
0.80	0.84	1.282
0.85	1.04	1.440
0.90	1.28	1.645
0.95	1.65	1.960
0.98	1.96	2.240
0.99	2.33	2.576
Poder estadístico $(1-\beta)$		
β	$(1-\beta)$	Z_β
0.01	0.99	2.326
0.05	0.95	1.645
0.10	0.90	1.282
0.15	0.85	1.036
0.20	0.80	0.842
0.25	0.75	0.674
0.30	0.70	0.524
0.35	0.65	0.385
0.40	0.60	0.253
0.45	0.55	0.126
0.50	0.50	0.000

FUENTE: https://www.fisterra.com/mbe/investiga/9muestras/9muestras2.asp

ANEXO 03: PRUEBAS PARAMÉTRICAS Y NO PARAMÉTRICAS

	PRUEBAS NO PARAMÉTRICAS				PRUEBAS PARAMÉTRICAS
Var. Aleatoria / Var. Fija		Nominal Dicotómica	Nominal Politómica	Ordinal	Numérica
Estudio transversal y muestras independientes	Un grupo	x^2 Bondad de ajuste (Binomial)	x^2 Bondad de ajuste	x^2 Bondad de ajuste	T de Student (una muestra)
	Dos grupos	x^2 Bondad de ajuste (Corrección de Yates) (Test de Fisher)	x^2 de homogeneidad	U Mann-Withney	T de Student (muestras independientes)
	Más de dos grupos	x^2 Bondad de ajuste	x^2 Bondad de ajuste	H Kruskal-Wallis	ANOVA con un factor Intrasujetos
Estudio longitudinal y muestras Relacionadas	Dos medias	Mc Nemar	Q de Cochran	Wilcoxon	T de Student (muestras relacionadas)
	Más de dos medias	Q de Cochran	Q de Cochran	Friedman	ANOVA para medidas repetidas (Intrasujetos)

FUENTE: Elaboración propia

ANEXO 04: CÓDIGO FUENTE AGKM PRINCIPAL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
function varargout = principal(varargin)
% PRINCIPAL M-file for principal.fig
% Edit the above text to modify the response to help principal
% Last Modified by GUIDE v2.5 30-Oct-2014 09:10:31
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @principal_OpeningFcn, ...
                   'gui_OutputFcn',  @principal_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before principal is made visible.
function principal_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to principal (see VARARGIN)

% Choose default command line output for principal
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes principal wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = principal_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in CargarImagen.
function CargarImagen_Callback(hObject, eventdata, handles)
% hObject    handle to CargarImagen (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
try
   % filename = 'mapa_seg.png'
    %      pathname = 'E:\Dropbox\Tesis puno\matlab\guia_basica\'
    [filename,pathname] = uigetfile('*.png','Selecciona imagen para
abrir');

    if isequal(filename,0)
        %Do nothing yet
    else
        imagen=imread(fullfile(pathname, filename));
        %imhist(imagen)
       handles.myImage = imread(fullfile(pathname, filename));
        axes(handles.axes1);
          image(imagen);
          Entropia=entropy(imagen)

          %Numero de grupos imagen original
          set(handles.text11,'String',Entropia);

%convertir la imagen a escala de grises
         I=rgb2gray(imagen);
         banda1=I(:);
         vector1=banda1(:);
         kCenter=double(round([vector1]));
         imagek=kCenter;

         handles.dados1 = double(imagek);
    end
    guidata(hObject, handles);
catch
    msgbox('Error')
end

% --- Executes on button press in ProcesarImagen.
function ProcesarImagen_Callback(hObject, eventdata, handles)
% hObject    handle to ProcesarImagen (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
k = str2double(get(handles.Cluster,'string'));
imagen_in=handles.myImage;

[E num_grupos PathFolder_in]=kmedia_colorimage3(imagen_in,k);
axes(handles.axes2)
%lectura de la imagen
background = imread([PathFolder_in '\imagen_seg_ag.png']);
axis off;
%imshow(background);
imagesc(background);
%Numero de grupos
set(handles.text3,'String',num_grupos);
%entropia para imagen tratada con AG
set(handles.text9,'String',E);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% presenta la imagen
imagen_in1=handles.myImage;
k1 = str2double(get(handles.Cluster1,'string'));
[Entropia1 num_grupos1]=kmeans_cluster(imagen_in1,k1,PathFolder_in);
```

```
axes(handles.axes4)
background1 = imread([PathFolder_in '\imagen_seg.png']);

axis off;
%imshow(background);
set(handles.text7,'String',num_grupos1);
imagesc(background1);
%Plot de la entropia segun kmedias
set(handles.text13,'String',Entropia1);

% --- Executes on button press in ClustersconAG.
function ClustersconAG_Callback(hObject, eventdata, handles)
% hObject    handle to ClustersconAG (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    h = str2double(get(handles.Ventana_h,'string'));
    guidata(hObject, handles);
    datos=handles.dados1;

    [datos,Fhat,piso]=AGmulti(h,datos);
    plot(piso,Fhat,'parent',handles.axes3);

function Ventana_h_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.

function Ventana_h_CreateFcn(hObject, eventdata, handles)
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Cluster_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.

function Cluster_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function text3_CreateFcn(hObject, eventdata, handles)

function Grupos_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function Grupos_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Grupos (see GCBO)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes during object creation, after setting all properties.
function Cluster1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end

function Cluster1_Callback(hObject, eventdata, handles)
% --- Executes when figure1 is resized.

function figure1_ResizeFcn(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.

function text7_CreateFcn(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.

function text9_CreateFcn(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.

function text11_CreateFcn(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.

function text13_CreateFcn(hObject, eventdata, handles)
```

ANEXO 05: CÓDIGO FUENTE AGs

```
function [datos,Fhat,piso] = AGmulti(h,datos)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   AGmulti.m: Archivo del tratamiento de           %%%%%%%%%%%
%   de datos con Algoritmos Genéticos               %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% %%%%%%%% Dimension de los datos %%%%%%%%%%%%%%%%%
d =  size(datos,2);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%        Parametros de los Algoritmos Geneticos      %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Numero de individuos de la poblacion
maxpop = 100;

% Numero de generaciones
num_generaciones =1000

%Probabilidad de recombinacion (crossover)
prob_cross = 0.4;

%Probabilidad de Mutacion
prob_mutacion = 0.08;

%%%%%inicializando las variables secundarias %%%%
i = 0;
j = 0;
k = 0;
l = 0;
m = 0;

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%     Algoritmo Genetico                   %%%%%%%%%%%%%%%%%
%%%%     para dimensiones diferente de 2     %%%%%%%%%%%%%%%%%

%Creacion de la poblacion inicial %%%%%%%%%%%%%%%%%%%%%%
% El tamaño de la cadena de bits depende del calculo de la exponencial m
%2^(m-1) < (255-0)*(10^1)<=2^m resolviendo la ecuacion queda
%  2^(m-1) < 2550 <=2^m evaluando los intervalos donde quedarian este valor
%  2^(12-1) < 2550 <=2^12 entonces m=12 por tanto necesitaremos 12 bits
%es por ese motivo que trabajamos con 12 bits cada variable que en total
%son 36 bits

switch d
   case 1
     maxstring  = 12;
     Npob=maxpop;
     for k=1:maxpop
        for l=1:maxstring
           %probabilidad de padres aleatorios
           Chrom(k,l) = round(rand);
        end
     end
   case 2
     maxstring  = 24
     for k=1:maxpop
        for l=1:maxstring
           Chrom(k,l) = round(rand);
        end
     end
   case 3
     maxstring  = 36
     for k=1:maxpop
```

```
        for l=1:maxstring
            Chrom(k,l) = round(rand);
        end
      end
    case 4
      maxstring  = 48
      for k=1:maxpop
        for l=1:maxstring
            Chrom(k,l) = round(rand);
        end
      end
   otherwise
      maxstring  = 'Ainda não estou preparado'
end
% %%%%%%%%% Numero de Generaciones de los algoritmos geneticos
%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Numero de generaciones
Ngen=num_generaciones;
for gen=1:num_generaciones
switch d
    case 1
%%%%%%%%%%%%%% divisão em duas cadeias %%%%%%%%%%%%%%%%%
    ChromX = Chrom(1:maxpop,1:maxstring);
%%%%%%%%%%%% Decodificação para base decimal %%%%%%%%%%%
    decX1 = bin2dec(strcat((int2str(ChromX'))'));
     Xi1 = 0 + decX1*(255-0)/(2^12);
     Z= [Xi1];

    case 2
%%%%%%%%%%%%%% divisão em duas cadeias %%%%%%%%%%%%%%%%%
    ChromX1 = Chrom(1:maxpop,1:maxstring/2);
    ChromX2 = Chrom(1:maxpop,(maxstring/2)+1:maxstring);

%%%%%%%%%%%% Decodificação para base decimal %%%%%%%%%%%
    decX1 = bin2dec(strcat((int2str(ChromX1'))'));
    decX2 = bin2dec(strcat((int2str(ChromX2'))'));

%%%%%%%%%%%% variaveis independentes %%%%%%%%%%%%%%%%%%%%
    Xi1 = -19+((45)/(2^(maxstring/2)-1))*decX1;
    Xi2 = -19+((45)/(2^(maxstring/2)-1))*decX2;
    Z= [[Xi1] [Xi2]];

    maximo = max(Xi1); minimo = min(Xi1);
    maximo = max(Xi2); minimo = min(Xi2);
    normal_Xi1 = 26*((Xi1-minimo)/(maximo-minimo))-10;
    normal_Xi2 = 26*((Xi2-minimo)/(maximo-minimo))-10;
    Z = [[normal_Xi1] [normal_Xi2]];

    case 3
%%%%%%%%%%%%%% división en tres cadenas %%%%%%%%%%%%%%%%%
    ChromX1 = Chrom(1:maxpop,1:maxstring/3);
    ChromX2 = Chrom(1:maxpop,((1*maxstring)/3)+1:2*maxstring/3);
    ChromX3 = Chrom(1:maxpop,((2*maxstring)/3)+1:maxstring);
%%%%%%%%%%%% Decodificação de base binaria para base decimal %%%%%%%%%%%
    decX1 = bin2dec(strcat((int2str(ChromX1'))'));
    decX2 = bin2dec(strcat((int2str(ChromX2'))'));
    decX3 = bin2dec(strcat((int2str(ChromX3'))'));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%% Ecuacion para convertir el valor binario a decimal %%%%%%%%%%%%
%%%%%%%%%%%% reemplazando el valor de m = 12 para nuestro caso %%%%%%%%%%%%%
%%%%%%%%%%% donde las variables x, y, z toman valores de 0 a 255 %%%%%%%%%%%

    Xi1 = 0 + decX1*(255-0)/(2^12);
    Xi2 = 0 + decX2*(255-0)/(2^12);
```

```
    Xi3 = 0 + decX3*(255-0)/(2^12);
    Z= [[Xi1] [Xi2] [Xi3]];
     case 4
%%%%%%%%%%%%% divisão em duas cadeias %%%%%%%%%%%%%%%%%%
    ChromX1 = Chrom(1:maxpop,1:maxstring/4);
    ChromX2 = Chrom(1:maxpop,((1*maxstring)/4)+1:2*maxstring/4);
    ChromX3 = Chrom(1:maxpop,((2*maxstring)/4)+1:(3*maxstring)/4);
    ChromX3 = Chrom(1:maxpop,((3*maxstring)/4)+1:maxstring);
%%%%%%%%%%%% Decodificação para base decimal %%%%%%%%%%%
    decX1 = bin2dec(strcat((int2str(ChromX1'))'));
    decX2 = bin2dec(strcat((int2str(ChromX2'))'));
    decX3 = bin2dec(strcat((int2str(ChromX3'))'));
    decX3 = bin2dec(strcat((int2str(ChromX3'))'));
%%%%%%%%%%%% variaveis independentes %%%%%%%%%%%%%%%%%%%%
    Xi1 = -11+((1000)/(2^(maxstring/2)-1))*decX1;
    Xi2 = -11+((1000)/(2^(maxstring/2)-1))*decX2;
    Xi3 = -11+((1000)/(2^(maxstring/2)-1))*decX3;
    Xi4 = -11+((1000)/(2^(maxstring/2)-1))*decX3;
    Z= [[Xi1] [Xi2] [Xi3] [Xi4]];

    maximo = max(Xi1); minimo = min(Xi1);
    maximo = max(Xi2); minimo = min(Xi2);
    maximo = max(Xi3); minimo = min(Xi3);
    maximo = max(Xi4); minimo = min(Xi4);
    normal_Xi1 = 26*((Xi1-minimo)/(maximo-minimo))-10;
    normal_Xi2 = 26*((Xi2-minimo)/(maximo-minimo))-10;
    normal_Xi3 = 26*((Xi3-minimo)/(maximo-minimo))-10;
    normal_Xi4 = 26*((Xi4-minimo)/(maximo-minimo))-10;
    Z= [[normal_Xi1] [normal_Xi2] [normal_Xi3] [normal_Xi4]];
   otherwise
      maxstring  = 'Ainda não estou preparado';
end
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%% Función objetivo %%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
datosT=datos;
%valores de entrada, generados usando AGs
    Z_T=sort(Z');
    F = datosT;
% Separanado en n Vectores segun la dimension=n
M1=datosT(:,1);
%M2=datosT(2,:);
%M3=datosT(3,:);
% Obteniendo el tamaño del Vector
n1=length(M1);
%n2=length(M2);
%n3=length(M3);
% Generando los vectores linea de cada dimension
  x1=round(Z_T(1,:));
%  x2=Z_T(2,:);
%  x3=Z_T(3,:);
% Generando la matriz para extraer el valor minimo y maximo
%Xtemp=[x1;x2;x3];
Xtemp=[x1];
min_Xtemp = min(min(Xtemp));
max_Xtemp = max(max(Xtemp));
%generando eje x que llamamos piso para plotar la funcion resultante con las dimensiones
minimo_Z = min(min(Z_T));
maximo_Z = max(max(Z_T));
piso=linspace(minimo_Z, maximo_Z, maxpop);
%Generando vector de ceros para sumar los vectores correspondientes
Fhat=zeros(size(x1));
f=0;
for ii=1:n1
    f=exp(-((x1-M1(ii))/h).^2/2)/...
    (sqrt(2*pi)*h)/n1;
```

```
    Fhat = Fhat + f;
end
Fi = Fhat';
     As=Fi%/sum(Fi)
%%%%% ELITISMO: Substituición del menor valor de la generación actual por el mayor valor de la
generación %%%%%anterior
switch d
   case 1

      if gen > 1
         [valormin,indmin]=min(Fi);
         if valormax>valormin%---------------------------------------------------------->
            Chrom(indmin,:) = elite(1,:);
            Fi(indmin,:)    = elite(2,1);
            Xi1(indmin,:)   = elite(3,1);
            As(indmin,:)    = elite(4,1);
         end
      end
%%%%%%%%%% ELITISMO: preservación del mayor valor obtenido
      [valormax,indmax] = max(Fi);
      elite(1,:)  = Chrom(indmax,:);
      elite(2,1)  = Fi(indmax,1);
      elite(3,1)  = Xi1(indmax,1);
      elite(4,1)  = As(indmax,1);

   case 2

      if gen > 1
         [valormin,indmin]=min(Fi);
         if valormax>valormin
            Chrom(indmin,:) = elite(1,:);
            Fi(indmin,:)    = elite(2,1);
            Xi1(indmin,:)   = elite(3,1);
            Xi2(indmin,:)   = elite(4,1);
            As(indmin,:)    = elite(5,1);
         end
      end
%%%%%%%%%% ELITISMO: preservación  do mayor valor obtenido
      [valormax,indmax] = max(Fi);
      elite(1,:)  = Chrom(indmax,:);
      elite(2,1)  = Fi(indmax,1);
      elite(3,1)  = Xi1(indmax,1);
      elite(4,1)  = Xi2(indmax,1);
      elite(5,1)  = As(indmax,1);

   case 3

      if gen > 1
         [valormin,indmin]=min(Fi);
         if valormax>valormin
            Chrom(indmin,:) = elite(1,:);
            Fi(indmin,:)    = elite(2,1);
            Xi1(indmin,:)   = elite(3,1);
            Xi2(indmin,:)   = elite(4,1);
            Xi3(indmin,:)   = elite(5,1);
            As(indmin,:)    = elite(6,1);
         end
      end
%%%%%%%%%% ELITISMO: preservación del mayor valor obtenido
      [valormax,indmax] = max(Fi);
      elite(1,:)  = Chrom(indmax,:);
      elite(2,1)  = Fi(indmax,1);
      elite(3,1)  = Xi1(indmax,1);
      elite(4,1)  = Xi2(indmax,1);
      elite(5,1)  = Xi3(indmax,1);
      elite(6,1)  = As(indmax,1);
```

```
    case 4

       if gen > 1
          [valormin,indmin]=min(Fi);
          if valormax>valormin
             Chrom(indmin,:) = elite(1,:);
             Fi(indmin,:)    = elite(2,1);
             Xi1(indmin,:)   = elite(3,1);
             Xi2(indmin,:)   = elite(4,1);
             Xi3(indmin,:)   = elite(5,1);
             Xi4(indmin,:)   = elite(6,1);
             As(indmin,:)    = elite(7,1);
          end
       end
%%%%%%%%% ELITISMO: preservação do maior valor obtido
       [valormax,indmax] = max(Fi);
       elite(1,:)  = Chrom(indmax,:);
       elite(2,1)  = Fi(indmax,1);
       elite(3,1)  = Xi1(indmax,1);
       elite(4,1)  = Xi2(indmax,1);
       elite(5,1)  = Xi3(indmax,1);
       elite(6,1)  = Xi4(indmax,1);
       elite(7,1)  = As(indmax,1);

   otherwise
      maxstring  = 'Ainda não estou preparado'
end
%%%%%% maximo valor obtenido en la generación %%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      maior(gen)= max(Fi);
      menor(gen)= min(Fi);
      medio(gen)= sum(Fi)/maxpop;

%%%%%%%%%%%% Criterio de parada %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      if maior(gen)>=3.63162799484525e-06%1
      disp('maximo encontrado     ')
      maior(gen);
      break
      end

%%%%%%%%%%%%%%% Seleçción via ruleta %%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
for n=1:maxpop
   k=0;
   partsum=0;
   ran=rand*sum(As);

   while partsum <= ran & k ~= maxstring  % procura do campo na roleta
      k=k+1;
      partsum=partsum+As(k);
   end
%%%%%%%%%%%%% Rearranjo do selecionado %%%%%%%%%%%%%%%%%%%%%%%%
   parent(n,:)=Chrom(k,:);
end

child=[];

%%%%% Recombinaçión dos cromossomas localizados en la cadena parent %%%%
%%%%%%%%%%%% Probabilidad de recombinación de 80 %%%%%%%%%%
for c=1:2:maxpop
   % probabilidad de cruce
   if rand <= prob_cross %---------------------------------------------------------->

      % Escoje el punto de cruce
      jcross=1+round((rand)*(maxstring-1-1));
```

```
      Xop=0;
      Xop=jcross;%---------------------------------------------------->

      child(c,:) = [parent(c,1:jcross) parent(c+1,(jcross+1):maxstring)];
      child(c+1,:)=[parent(c+1,1:jcross) parent(c,(jcross+1):maxstring)];
    else
      child(c,:)=parent(c,:);
      child(c+1,:)=parent(c+1,:);
    end
end
%%%%%%%%% Mutaçión com  probabilidad de 1% para cada alelo%%%%%%%%%
for n=1:maxpop
  for k=1:maxstring
     ran=rand;
     if ran<=prob_mutacion%---------------------------------------------------->
        if child(n,k)==0
          child(n,k)=1;
        else
          child(n,k)=0;
        end
        Mop=child;%---------------------------------------------------->
     end
  end
end
%%%%%%%%%%%%%%%%%%% Nueva población %%%%%%%%%%%%%%%%%%
Chrom=child;

end
Chrom;
figure
%malla del grafico
%grid on;
hold on;
title('Clusters Segun Algoritmos Geneticos')
xlabel('Puntos de niveles de intensidad')
ylabel('Estimación de densidad')
plot(piso,Fhat,'k');

end
```