Universidad Nacional de Asunción

Procesamiento de Imágenes y Extracción de Características Morfológicas para Clasificación de Galaxias

José Zacarías Salinas Núñez

Universidad Nacional de Asunción
Facultad Politécnica
San Lorenzo, Paraguay
2016

Procesamiento de Imágenes y Extracción de Características Morfológicas para Clasificación de Galaxias

José Zacarías Salinas Núñez

Trabajo de Fin de Grado presentado como requisito para obtener el título de:
Ingeniero en Informática

Tutor:
D.Sc. Christian Emilio Schaerer

Co-tutores:
Ph.D. Miguel García-Torres
Ph.D. Horacio Legal Ayala

Universidad Nacional de Asunción
Facultad Politécnica
San Lorenzo, Paraguay
2016

Dedicado a mi mamá y a mi compañera de vida Juany, que siempre estuvieron a mi lado apoyándome en todo. A mis amigos que me ayudaron a llegar hasta el final.

Agradecimientos

De manera especial al Profesor Christian Schaerer por el tiempo, la paciencia y los conocimientos compartidos durante el proceso de investigación y desarrollo de este Trabajo de Fin de Grado.

A Miguel García-Torres por haberme iniciado en la astronomía y haber guiado mis primeros pasos en este fascinante camino.

Al Profesor Horacio Legal por haberme ayudado incansablemente en este proyecto de investigación.

A Waldemar Villamayor-Venialbo por haberme enseñado el método científico, y hacerme entender que tenía que limpiar mi mente de las cosas que ya daba por hecho sin ninguna demostración.

A Jorge Rodas Benítez por haberme dado los consejos que fueron de mucha utilidad y me ayudaron a avanzar notablemente con el trabajo.

Al Laboratorio de Computación Científica y Aplicada por el espacio brindado para realizar los estudios.

Al Consejo Nacional de Ciencia y Tecnología (Conacyt) por haber apoyado mediante la financiación del proyecto de investigación 14-INV-202.

Resumen

La clasificación de galaxias es una tarea importante en la astronomía, en el estudio a gran escala del universo. Esta tarea era tradicionalmente realizada de forma manual, pero como la astronomía ha experimentado una explosión de datos, se requieren del uso de nuevas técnicas acordes a este incremento en el volumen de los datos. En este trabajo analizamos el rendimiento de varios algoritmos de aprendizaje automático (*Bayes Net*, *Naïve Bayes*, *Support Vector Machine, Multilayer Perceptron*, K-nn, C4.5, *Logistic Model Tree, Random Forest* y *Random Tree*) en la clasificación automática de galaxias. Probamos los enfoques con 82 imágenes de galaxias cercanas del nuevo catálogo general (NGC, por sus siglas en inglés), y consideramos tres (E, S, Irr), cinco (E, S0, Sa+Sb, Sc+Sd, Irr) y siete (E, S0, Sa, Sb, Sc, Sd, Irr) tipos de galaxias.

Las imágenes son estandarizadas para remover el ruido, el efecto de la orientación y la traslación. Las características son extraídas por su apariencia morfológica (MF, por sus siglás en inglés), análisis de componentes principales (PCA, por sus siglas en inglés) y análisis de componentes independientes (ICA, por sus siglas en inglés). Las MFs están basadas en la percepción visual de las galaxias como elongación, factor de forma, convexidad, factor de forma rectangular, índice de asimetría, picos horizontales y verticales del histograma, ratio de circularidad, ratio de forma, ratio de compacidad, ratio de radio y firma de dispersión lumínica. PCA e ICA son extraídos de la matriz del *dataset* $C = AA^T$ y la traspuesta del *dataset* $C = A^TA$ donde una fila A representa una imagen convertida a un vector de una dimensión.

Los resultados experimentales muestran que al incrementar el número de tipos de galaxias, se degrada el rendimiento del modelo. La combinación de estandarización con PCA alcanza la mejor precisión (89 % con *Naïve Bayes* para 3 tipos de galaxias). Sin embargo, con MFs alcanza una precisión más baja (86 % con *Naïve Bayes* para 3 tipos de galaxias), este tipo de características permite un mejor entendimiento de las reglas del clasificador.

A pesar de los resultados prometedores, es necesario realizar más investigaciones para mejorar la clasificación cuando se incremente el número de tipos de galaxias. Una posible razón de los resultados pobres alcanzados clasificando 5 y 7 clases de galaxias podría ser por el pequeño número de instancias de algunas clases; por esto, incrementando el *dataset* podría ayudar a mejorar los resultados.

Palabras clave: Procesamiento de imágenes, minería de datos, aprendizaje automático, clasificación de galaxias.

Abstract

Galaxy classification is an important task in astronomy in the large scale study of the universe. Although this task traditionally is performed manually, astronomy has experienced an explosion of data that require the use of new techniques to deal with this increase of the data volume. In this work we analyze the performance of several machine learning approaches (Bayes Net, Naïve Bayes, Support Vector Machine, Multilayer Perceptron, K-nn, C4.5, Logistic Model Tree, Random Forest and Random Tree) on automated classification of galaxy images. We tested the approaches on 82 nearby galaxy images from the new general catalogue (NGC), and considered three (E, S, Irr), five (E, S0, Sa+Sb, Sc+Sd, Irr) and seven (E, S0, Sa, Sb, Sc, Sd, Irr) galaxy types.

The images are standardized to remove noise and the effect of orientation and translation. Features are extracted by morphological appearance (MF), Principal Component Analysis (PCA) and Independent Component Analysis (ICA). The MF are based on the perceived visual characteristics of the galaxy like elongation, form-factor, convexity, bounding-to-fill factor, asymmetry index, horizontal and vertical peaks of histogram, circularity ratio, form ratio, compact ratio, radius ratio and luminosity intensity signature. PCA and ICA are extracted from the dataset matrix $C = AA^T$ and the transpose of dataset $C = A^T A$ where a row of A represents an image converted into a one dimensional vector.

Experimental results show that increasing the number of galaxy types degrades the model performance. The combination of standardization with PCA leads to classification models with high accuracy (89 % with naive Bayes for 3 galaxy types). However, although morphological features lead to models with lower predictive power (86 % with naïve Bayes and 3 galaxy types), this type of features allow a better understanding of the classification rules.

Despite the promising results, more research is necessary to improve the classification when increasing the number of glaxy types. A possible reason for the poor result achieved for the 5 and 7 classes could be the small number of instances for some classes; therefore increasing the dataset could improve the results.

Keywords: Image processing, datamining, machine learning, galaxy classification.

Índice general

Índice de figuras

Índice de tablas

Acrónimos y Símbolos

Acrónimos

AI	Índice de asimetría, *asymmetry index.*
PCA	Análisis de componentes principales, *principal component analysis.*
MF	Características morfológicas, *morphological features.*
ICA	Análisis de componentes independientes, *independent component analysis.*
PC	Componentes principales. *principal components.*
IC	Componentes independientes, *independent components.*
FDL	Firma de dispersión lumínica.
S	Galaxia espiral, *spiral.*
E	Galaxia elíptica, *elliptical.*
Irr	Galaxia irregular, *irregular.*
S0	Galaxia lenticular.
El	Elongación.
FF	Factor de forma.
Conv	Convexidad.
FFR	Factor de forma rectangular.
PH	Picos horizontales.
PV	Picos verticales.

RCir	Ratio de circularidad.
SDSS	*Sloan Digital Sky Survey.*
RF	Ratio de forma.
RCom	Ratio de compacidad.
RR	Ratio de radio.
RGB	Rojo Verde Azul, *Red Green Blue.*
NGC	*New General Catalogue.*

Símbolos

Ai	Índice de asimetría.
Ph	Picos horizontales.
Pv	Picos verticales.
e	Elongación.
F	Factor de forma.
C_v	Convexidad.
R_1	Ratio de circularidad.
R_2	Ratio de forma.
R_3	Ratio de compacidad.
R_4	Ratio de radio.

Capítulo 1

Introducción

1.1. Historia

Antes de hablar de galaxias, primero debemos hablar de historia y evolución. Los astrónomos están convencidos en su gran mayoría, que el universo surgió a partir de una gran explosión (Teoría del *Big Bang*) ocurrida hace aproximadamente 13.800 millones de años [1]. Esta convicción se debe a la acumulación de evidencia observacional y a la propuesta del sacerdote belga y astrónomo *Georges Lemaître* de que un universo en expansión debería remontarse en el tiempo hasta un único punto de origen [2]. La teoría del *Big Bang* afirma que antes de la explosión, el universo se encontraba en un estado de muy alta densidad y temperatura, luego se expandió. Después de esta expansión inicial, el universo se enfrió lo suficiente para permitir la formación de las partículas subatómicas y más tarde simples átomos. Nubes gigantes de éstos elementos primordiales se unieron más tarde debido a la acción de la gravedad formando lo que hoy conocemos como estrellas y galaxias [3].

Etimológicamente, la palabra galaxia deriva del griego $\gamma\alpha\lambda\alpha\xi\iota\alpha\zeta$ que significa "lácteo". En las noches bien despejadas y en ausencia de luz artificial, se puede ver muy bien la zona central de nuestra galaxia en forma de una franja alargada, con color blanquecino, casi lechoso. Los romanos la llamaban *circulus lacteus*, más tarde Vía Láctea o "Camino de leche". El término "Vía" se debe a que la Vía Láctea sirvió de orientación a viajeros, navegantes y caminantes. Modernamente, luego de haberse descubierto la existencia de otras galaxias, la palabra "galaxia" pasó a ser utilizada para denominarlas genéricamente y "Vía Láctea" pasó a designar como nombre propio a la nuestra [4].

Anteriormente, las galaxias eran conocidas como nebulosas, el tamaño y la forma de éstas eran calculadas razonablemente bien, pero se pensaba que todas formaban parte de nuestra galaxia. La "Nebulosa de Andrómeda" era una de ellas, hasta que en el año 1864 el astrónomo británico *Sir William Huggins* observó su espectro y notó que no se parecía al esperado en un objeto nebuloso y sí al de uno hecho de estrellas. Sin embargo, siguió siendo considerada como nebulosa, a pesar de la evidencia. Recién en el año 1917, el astrónomo estadounidense *Heber Doust Curtis* descubrió una nova en Andrómeda, y buscando en placas fotográficas anteriores encontró 11 más, al parecer, todas ellas de magnitudes 10 veces más débiles que las novas registradas en la Vía Láctea. Con esto, supuso que el objeto se encontraba a unos 500.000 años luz de distancia y que tanto ella como otros objetos similares, conocidos hasta entonces como "nebulosas espirales", no eran nebulosas sino galaxias independientes.

En el año 1925, el astrónomo estadounidense *Edwin Hubble* encontró estrellas cefeidas (son estrellas que pulsan radialmente, variando tanto en temperatura como en diámetro para producir

cambios de brillo con periodos y amplitudes estables muy regulares, y sirven como indicadores de distancia para establecer escalas de distancias galácticas y extragalácticas) en fotografías de Andrómeda, dejando claro que tales objetos son en realidad galaxias similares a la nuestra, solo que a grandes distancias. Desde ese momento la "nebulosa de Andrómeda" pasó a ser conocida definitivamente como la "galaxia de Andrómeda".

Hubble también registró múltiples corrimientos al rojo y al azul de varios objetos del universo y en 1929 publicó un análisis de la velocidad radial, respecto a la Tierra, de las nebulosas cuya distancia había calculado estableciendo que, aunque algunas nebulosas extragalácticas tenían espectros que indicaban que se movían hacia la Tierra, la gran mayoría mostraba corrimientos hacia el rojo, que solo podían explicarse bajo la suposición de que se alejaban [5].

Hubble concluyó que la única explicación consistente con los corrimientos hacia el rojo registrados, era que, apartando al "grupo local" de galaxias cercanas, todas las nebulosas extragalácticas se estaban alejando y que cuanto más lejos se encontraban más rápidamente se alejaban. Esto llevó al astrónomo a elaborar el postulado que hoy conocemos como la "Ley de *Hubble*" acerca de la expansión del universo [6].

Cuando observamos el cielo y vemos los variados objetos celestes, no vemos su estado actual, sino su historia, vemos su pasado, y cuanto más lejos miramos, más atrás en el tiempo vemos. La luz viaja a 300.000 kilómetros por segundo aproximadamente, y cuando miramos la Luna (el objeto celeste más cercano), la vemos como era hace aproximadamente 1,2 segundos, al Sol, lo vemos como era hace 8,3 minutos, a la estrella más cercana "Alfa Centauri" la vemos como era hace 4,3 años y a la galaxia "Andrómeda" la vemos como era hace 2,56 millones de años. Por esto, para hacer de las distancias intergalácticas más manejables, la unidad de medida más usual es el "año luz", que representa la distancia recorrida por la luz durante 1 año.

Las galaxias son un conjunto de estrellas, nubes de gas, planetas, polvo cósmico, materia oscura y energía que permanecen unidos mediante la acción de la fuerza de gravedad y aislados de sistemas similares por grandes regiones de espacio vacío. La apariencia visual de las galaxias proporciona a los astrónomos mucha información sobre la composición y evolución de las mismas. Esta apariencia está en función de la edad de las mismas, de los elementos que están compuestas y de su proceso de formación, por lo que las galaxias más jóvenes todavía no tienen una estructura definida y son más pequeñas que las galaxias más viejas. Entre las galaxias más viejas, están las elípticas y las espirales, cuyo proceso de formación aún no se sabe con certeza, aunque existen teorías que explican dichas formas [7].

La primera teoría propone que la nube de gas colapsa para formar una galaxia, y su *spin* rotacional determina qué tipo de galaxia será. Así, algunos astrónomos creen que las galaxias espirales fueron formadas de nubes que tenían un *spin* rotacional significante y las galaxias elípticas fueron formadas de nubes que no tenían este *spin* rotacional [8]. La segunda teoría propone que las galaxias elípticas fueron formadas de las colisiones de galaxias espirales. Ésta teoría está apoyada en tres hechos interesantes: [i] que en el universo temprano, las galaxias estaban más cercas unas de otras de lo que están ahora, por lo que las colisiones probablemente fueron muy comunes, [ii] grandes galaxias elípticas se forman en *clusters* de muchas galaxias donde las colisiones son más probables, [iii] las galaxias elípticas no tienen mucho gas interestelar comparado con las espirales. Estas diferencias morfológicas de las galaxias llevó a *Hubble* a desarrollar una clasificación llamada "secuencia de *Hubble*" en el año 1936, en donde las galaxias son clasificadas por su forma aparente (Morfología Visual) y es la clasificación que usamos como referencia en este Trabajo de Fin de Grado (TFG).

La Secuencia de *Hubble*

Hubble ha basado su clasificación en fotografías de las galaxias tomadas con telescopios de la época (alrededor de 1936). Al principio creyó que las galaxias elípticas eran una forma inicial, y que posteriormente evolucionaban a espirales. Nuestro conocimiento actual sugiere que la situación podría ser opuesta, no obstante esta creencia quedó en la jerga de astrónomos que aún hablan de "tipo primitivo" o "tipo avanzado" cuando se refieren a galaxias elípticas y espirales, respectivamente. *Hubble* dividió los tipos de galaxias según la siguiente clasificación (**Figura 1.1**):

- **Galaxias elípticas**: (E0-7) tienen forma elíptica, con una distribución bastante uniforme de las estrellas por todas partes. El número indica el grado de excentricidad: las galaxias E0 son casi redondas, mientras que las E7 son muy aplanadas. El número indica solo la apariencia de la galaxia en el cielo, no su geometría real.

- **Galaxias lenticulares**: (S0 y SB0) parecen tener una estructura de disco con una concentración de estrellas central proyectándose de él. No muestran ninguna estructura espiral.

- **Galaxias espirales**: (Sa-d) tienen una concentración de estrellas central y un disco aislado que presenta brazos espirales. Los brazos están centrados alrededor de la protuberancia, variando de los muy arremolinados y poco definidos (Sa) a los muy sueltos y definidos (Sc y Sd). Así, mientras que en las primeras, la concentración central es muy pronunciada, en las últimas lo es bastante menos, y salvo excepciones, la cantidad de estrellas jóvenes y la proporción de gas van aumentando a lo largo de la secuencia.

- **Galaxias espirales barradas**: (SB0/a-d) tienen una estructura en espiral, similar a las galaxias espirales pero los brazos se proyectan desde el final de una "barra" central en lugar de emerger de una concentración central, como cintas en los extremos de una vara. De nuevo, SBa a SBd indica como de arremolinados están estos brazos y el grado de desarrollo de la concentración central, y de nuevo, salvo excepciones, al ir progresando en la secuencia, la cantidad de gas y estrellas jóvenes va aumentando.

- **Galaxias espirales intermedias**: (SAB0/a-c) tienen una morfología intermedia entre las galaxias espirales y las galaxias espirales barradas.

- **Galaxias irregulares**: (Irr) se dividen en Irr-I, que muestran estructura espiral deformada, e Irr-II para las galaxias que no encajan en ninguna otra categoría.

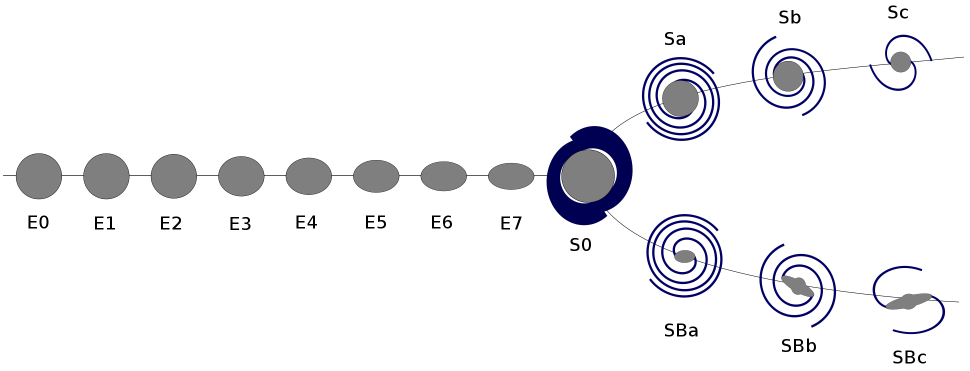


Figura 1.1: Secuencia de *Hubble*

1.2. Definición del Problema

Antes de la era de la información, las observaciones se hacían solo mirando a través del ocular de los telescopios o analizando con una lupa las placas fotográficas, por lo que las clasificaciones y categorizaciones de objetos celestes se realizaban de forma manual y lineal. En otras palabras, la categorización de los objetos se realizaba uno a uno.

Con la llegada de las tecnologías de información, la astronomía tuvo una rápida evolución hacia la automatización. La captura de datos ya no era un trabajo de "uno a uno" sino que empezaron a llegar grandes cantidades de datos de cientos y miles de objetos simultáneamente, tanto de distintos observatorios como de distintas fuentes (luz visible, rayos X, ultravioleta, rayos gamma, radiofrecuencias, entre otros).

Con esto, empieza un nuevo problema, y es la capacidad para analizar toda esta información en un tiempo razonable. La clasificación manual ya no es una tarea viable por lo que se necesitan de alternativas para lidiar con esta situación.

1.3. Fases de un Sistema de Clasificación de Imágenes de Galaxias

La clasificación morfológica automática de galaxias ya lleva como dos décadas de estudio. Existen varias propuestas pero todas se basan en un mismo esquema de procesos como se muestra a continuación:

a - **Preprocesamiento de Imágenes**: incluye generalmente un proceso de filtrado para eliminación de ruido (variación aleatoria de brillo o color que no corresponde con la realidad) y técnicas para realzar los detalles importantes y estandarizar el formato de las imágenes.

b - **Extracción de Características**: se logra identificando características asociadas a la morfología del objeto. Entre las características podemos citar [9]: elongación, factor de forma, convexidad, factor de forma rectangular e índice de asimetría. También existen otros tipos de características (no morfológicas) como lo son las obtenidas con el PCA [9, 10].

c - **Proceso de Clasificación**: se logra utilizando técnicas de minería de datos junto con las características extraídas de las galaxias para identificar a qué clase o grupo pertenecen.

1.4. Importancia

La clasificación de galaxias representa una tarea importante en la astronomía porque los astrofísicos frecuentemente hacen uso de grandes catálogos de información, los cuales usan para probar teorías existentes o formular nuevas conjeturas que expliquen los procesos físicos que gobiernan la formación de galaxias, la formación de estrellas y la naturaleza del universo. Aunque esta tarea era realizada manualmente, con el incremento de los *Surveys* autónomos (estudios o exploraciones del espacio por medio de telescopios), la astronomía ha experimentado una explosión de datos que requiere el uso de nuevas técnicas que puedan analizar de forma automática las imágenes astronómicas.

1.5. Justificación

Una fascinante discusión se está produciendo entre los astrónomos. ¿Quién iba a creer que este campo de investigación podría ser el primero en ser automatizado en un par de décadas?. El ratio en

el que las máquinas hacen descubrimientos ya excede al de los humanos. Con esta automatización, los telescopios inteligentes serán capaces de hacer fotos y luego editarlas y catalogar los datos en tiempo real, para finalmente descargarlos en un observatorio virtual para que los astrónomos lo analicen.

En este sentido, el principal trabajo de los astrónomos ha variado del tradicional tener el ojo puesto en el cielo estableciendo objetivos y haciendo estadísticas, a centrarse principalmente en hacer ciencia. En otras palabras, se centrarán más en buscar patrones que en descubrir objetos espaciales, elaborar nuevas teorías basadas en los descubrimientos y descubrir las ramificaciones e implicaciones de objetos que desafían las normas establecidas. Esto debería ayudar al progreso de los investigadores de forma considerable, y solo será posible con una nueva generación de observatorios capaces de operar de forma autónoma.

1.6. Objetivo General

El Objetivo general del presente TFG es proponer una clasificación morfológica automática de galaxias utilizando características morfológicas (MF, por sus siglas en inglés), componentes principales (PC, por sus siglas en inglés) y componentes independientes (IC, por sus siglas en inglés) para reconocer el arquetipo al que pertenece una galaxia dada y organizar estos arquetipos en un esquema simple que pueda ser interpretado en términos físicos y de evolución de las mismas.

1.7. Objetivos Específicos

1 - Explorar la efectividad de distintos extractores de características utilizados en el estado del arte y proponer nuevos métodos de extracción de MFs.

2 - Comparar el rendimiento de diferentes algoritmos de aprendizaje automático supervisado.

3 - Formalizar el PCA e ICA como extractores de características.

4 - Analizar el desempeño del PCA e ICA y su discusión.

5 - Realizar selección de atributos para MFs, PCs e ICs.

6 - Realizar análisis y comparación de resultados de aplicar clasificación con MFs, PCs e ICs.

7 - Formular trabajos futuros en base a los resultados obtenidos.

1.8. Estructura del Trabajo

El documento se ha organizado en cinco capítulos. Luego de presentar una introducción al trabajo junto con los objetivos del mismo en el Capítulo 1, se describen los procesos de un sistema de clasificación automático en el Capítulo 2. El Capítulo 3 presenta los algoritmos utilizados en este trabajo de clasificación. El Capítulo 4 presenta los resultados experimentales. En el Capítulo 5 se extraen las principales conclusiones y se proponen trabajos futuros relacionados al presente TFG. Se anexan dos apéndices, uno para detallar los códigos utilizados en la elaboración del sistema de preprocesamiento de imágenes y extracción de características, otro para mostrar la publicación realizada en el *CCiS 2016*.

Capítulo 2

Fundamento Teórico

En este capítulo describiremos en detalle cada una de las tres fases de un sistema de clasificación automática de galaxias, entre las que incluiremos métodos del estado del arte y nuevas propuestas.

2.1. Preprocesamiento de Imágenes

Las imágenes de galaxias generalmente difieren en tamaño, colores, formato, orientación y en la mayoría de las veces, la galaxia contenida en la imagen no está centrada. Para evitar esta heterogeneidad, en esta fase creamos imágenes invariantes al color, orientación y posición. Primero, aplicamos un filtro RGB a escala de grises mediante la siguiente ecuación:

$$y = 0{,}587\, g + 0{,}299\, r + 0{,}114\, b \tag{2.1}$$

donde g, r y b representan los colores verde, rojo y azul respectivamente, y es la intensidad del pixel en escala de grises siguiendo la recomendación del estándar 601 de la Unión Internacional de Telecomunicaciones [11]. En la **Figura 2.1** podemos ver algunos ejemplos de imágenes de distintos tipos de galaxias en diferentes posiciones que son obtenidas en escala de grises mediante la ecuación (2.1) desde una base de datos de un observatorio virtual.

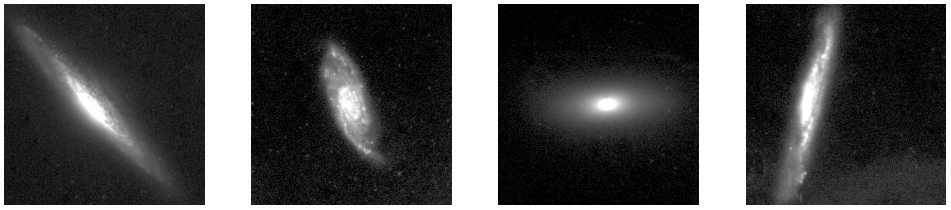


Figura 2.1: Imágenes en escala de grises

Segundo, un umbral determinado es aplicado para binarizar la imagen y remover ruido con (2.2)

$$B(i,j) = \begin{cases} 1 & \text{si} \quad I(i,j) > \tau \\ 0 & \text{otro caso} \end{cases} \tag{2.2}$$

donde I es la imagen original (**Figura 2.1**), B la imagen binarizada, τ es el umbral y los índices i y j representan las filas y columnas de los pixeles de la imagen. En este TFG asignamos a τ el valor 50. En la **Figura 2.2** podemos ver el resultado de aplicar binarización a la imagen original.

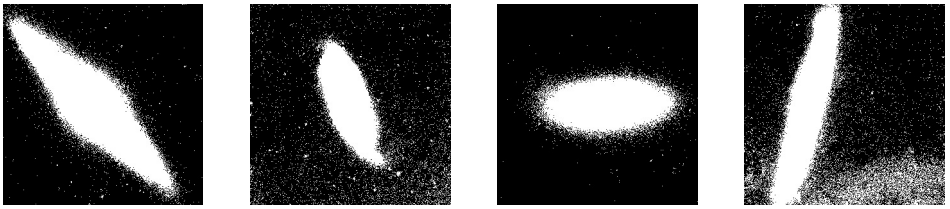


Figura 2.2: Imágenes binarizadas

Tercero, un filtro de apertura es aplicado a la imagen binarizada para remover el ruido restante con (2.3)

$$A = B \circ E = (B \ominus E) \oplus E \tag{2.3}$$

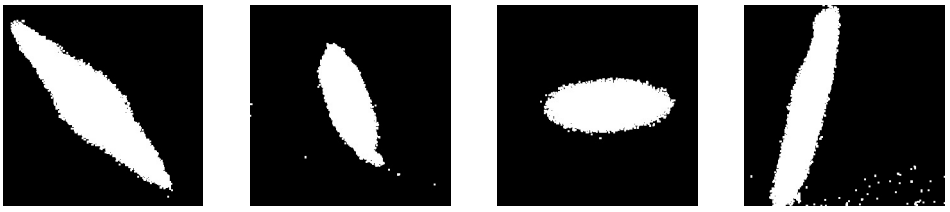


Figura 2.3: Imágenes filtradas

donde B es la imagen binarizada, E es el elemento estructurante y los operadores $\ominus$ y $\oplus$ representan erosión y dilatación respectivamente. En este TFG aplicamos el filtro de apertura utilizando una matriz de unos de tamaño 2 x 2 como elemento estructurante y realizando una sola iteración. En la **Figura 2.3** vemos las imágenes luego de realizar un filtro de apertura para eliminar la mayor cantidad de ruido posible.

Cuarto, los pixeles de la imagen original que corresponden con los pixeles de la imagen con mayor contorno obtenida con la ecuación (2.3) es extraída con (2.4)

$$Y = I \cap A \tag{2.4}$$

donde I es la imagen original, A el resultado de hacer apertura con (2.3) y el operador $\cap$ representa la intersección. En la **Figura 2.4** vemos el resultado de interceptar la imagen original con la imagen filtrada utilizada como máscara.

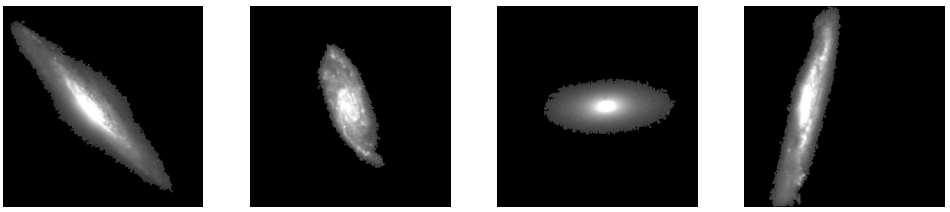


Figura 2.4: Región de interés extraída

Quinto, la galaxia es centrada al recuadro de la imagen con una transformación afín (2.5), con la matriz de transformación $\mathbf{M}$ como se muestra en (2.6)

$$z = \mathbf{M}x + w \tag{2.5}$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}, \quad \begin{cases} t_x = cg_x - ci_x, \\ t_y = cg_y - ci_y \end{cases} \tag{2.6}$$

donde z es la imagen transformada, w corresponde a la traslación, $\mathbf{M}$ corresponde a los cambios de escala, rotaciones y sesgos, cg y ci son el centro de la galaxia y el centro del recuadro de la imagen respectivamente. En la **Figura 2.5** vemos las galaxias trasladadas al centro del cuadro.

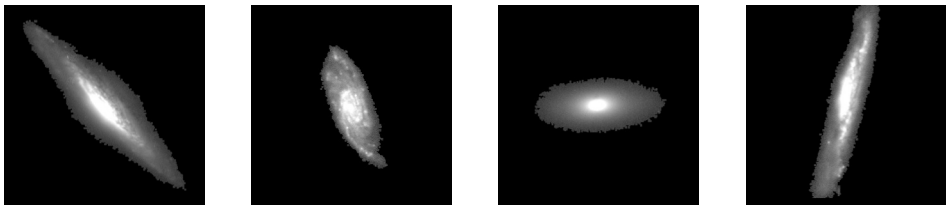


Figura 2.5: Imágenes trasladadas al centro

Finalmente, la galaxia es rotada para alinear el eje principal con el eje horizontal, el ángulo del eje principal es calculado con (2.7)

$$\alpha = \arctan(a_x/a_y) \tag{2.7}$$

donde a_x y a_y son los valores de fila y columna del eje principal. En la **Figura 2.6** vemos las galaxias rotadas con el eje mayor alineado con la horizontal.

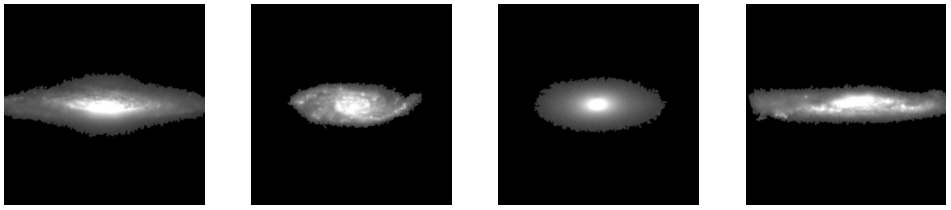


Figura 2.6: Imágenes alineadas con la horizontal

2.2. Extracción de Características

En esta fase, varias características morfológicas y no morfológicas son extraídas de las imágenes procesadas.

2.2.1. Características Morfológicas

Las MFs están basadas en la percepción visual de las galaxias. Los resultados experimentales obtenidos en [9] demuestran que las MFs son más efectivas que los PCs. Por esta razón, usaremos ICA como extractor de características para verificar si los ICs tienen un mejor desempeño. A continuación se definen las fórmulas utilizadas para la extracción de MFs.

La elongación (e) es definida como:

$$e = \frac{(a-b)}{(a+b)} \tag{2.8}$$

donde a y b son los ejes mayor y menor de la elipse que envuelve a la galaxia.

El factor de forma (F_f) es definido como:

$$F_f = \frac{A(I)}{P(I)} \tag{2.9}$$

donde $A(I)$ es el área de la imagen I y $P(I)$ es el perímetro de la imagen I.

La convexidad (C_v) está definida por:

$$C_v = \frac{P(I)}{P(B_r(I))} \tag{2.10}$$

donde $B_r(I)$ es el mínimo rectángulo que envuelve a la galaxia.

El factor de forma rectangular (F_r) es definido mediante la siguiente ecuación:

$$F_r = \frac{A(B_r(I))}{A(I)}. \tag{2.11}$$

El índice de asimetría (A_i) es definido según:

$$A_i = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}|A(i,j) - F(i,j)|}{256\, m\, n} \tag{2.12}$$

donde A es la imagen de la galaxia, F es la imagen de la galaxia rotada 180 grados, m y n son el alto y ancho respectivamente en pixeles del mínimo rectángulo que envuelve a la galaxia. El rango de intensidad de los pixeles en escala de grises va de 0 a 255, entonces, el denominador es multiplicado por 256 para normalizar la ecuación.

Una de las principales contribuciones del presente TFG es la propuesta de otras MFs, a las cuales llamamos picos horizontales (Ph) y picos verticales (Pv) del histograma, y la firma de dispersión lumínica (FDL). También haremos uso de las propuestas realizadas en [12] Ratio de Circularidad, Ratio de Forma, Ratio de Compacidad y Ratio de Radio.

Los picos horizontales están definidos como:

$$Ph = \frac{d}{dt}Hh(Rh \cap I) \tag{2.13}$$

donde Rh es la recta que cruza el eje horizontal de la galaxia, I es la imagen de la galaxia y Hh es el histograma de la intersección entre Rh e I.

Los picos verticales están definidos como:

$$Pv = \frac{d}{dt}Hv(Rv \cap I) \tag{2.14}$$

donde Rv es la recta que cruza el eje vertical de la galaxia y Hv es el histograma de la intersección entre Rv e I.

Una desventaja que introduce (2.9) es la posibilidad de variación de la proporción entre el área y el perímetro cuando se consideran formas similares de tamaños diferentes, por más que las imágenes sean de galaxias del mismo tipo. Por ejemplo (y solo a modo de analogía) si consideramos

un cuadrado de lado 2 y otro de lado 3, la ecuación (2.9) devolvería 0,5 y 0,75 respectivamente. Podríamos adaptar esta medida para hacerla adimensional utilizando la siguiente ecuación:

$$Cir = \frac{A}{P^2} \tag{2.15}$$

donde A es el área y P el perímetro. Esto daría entonces el valor de 1/16 para ambos cuadrados. Esta proporción adquiere su máximo valor cuando la forma del objeto es circular. En este caso usaremos (2.16).

$$Cir = \frac{\pi r^2}{(2\pi r)^2} = \frac{1}{4\pi} \tag{2.16}$$

Para hacer que la medida esté entre 0 y 1 podríamos, por tanto, escalar multiplicando por 4π. Los geógrafos se valen de esto y lo denominan ratio de circularidad (R_1):

$$R_1 = \frac{4\pi A}{P^2}. \tag{2.17}$$

El ratio de forma (R_2) está dado por (2.18)

$$R_2 = \frac{4A}{\pi l^2} \tag{2.18}$$

donde l es la longitud de la línea que une los dos puntos más distantes de la forma.

El ratio de compacidad (R_3) está dado por (2.19)

$$R_3 = \frac{A}{\pi R^2} \tag{2.19}$$

donde R es el radio del círculo más pequeño que rodea la forma.

El ratio de radio (R_4) está dado por (2.20)

$$R_4 = \frac{r}{R} \tag{2.20}$$

donde r es el radio del círculo mayor que pueda insertarse en la forma.

Por último, creamos el método de extracción de característica FDL, y consiste en darle un peso a los pixeles de la imagen de acuerdo a su distancia al centro de la misma. La distancia es medida en pixeles de forma horizontal o vertical a los ejes horizontal o vertical respectivamente del punto central, y la considerada es la mayor distancia entre las dos. El peso de cada pixel está dado por la siguiente ecuación:

$$FDL = \frac{1}{n} I \tag{2.21}$$

donde

$$n = n_{i-1} + 8_i \tag{2.22}$$

representa la cantidad de pixeles a una misma distancia al centro de la imagen e I es la intensidad en escala de grises del pixel considerado. De esta forma, por cada distancia al centro de la imagen, habrá un máximo de 255 de intensidad, sin importar cuantos pixeles estén a la misma distancia. El valor 8 es porque el punto central está rodeado por 8 pixeles, a la distancia de 1 pixel cada uno, y cuando aumenta la distancia, habrán 8 pixeles más que el nivel anterior. Por esto, el valor de n es igual a 8 por la distancia al eje más lejano, más el valor de n a una distancia de un pixel menos al centro de la imagen.

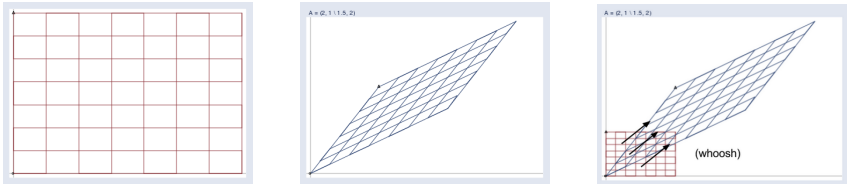
2.2.2. Análisis de Componentes Principales

El PCA es un método estadístico que transforma un número de variables posiblemente correlacionadas a un número más pequeño de variables no correlacionadas o PCs. El PCA es usado generalmente para reducir la dimensionalidad de un conjunto de datos mientras retiene la mayor cantidad de información posible. PCA es una herramienta para buscar patrones en datos de muchas dimensiones como lo son las imágenes.

Los datos habitualmente se organizan en una matriz X de $n \times p$ dimensiones donde n es el número de filas (observaciones) y p representa el número de columnas (variables). El objetivo del PCA es sintetizar la información contenida en las p columnas de la matriz de datos X, es decir, las variables, tal que la dimensión del problema se reduzca.

El PCA se basa en un procedimiento matemático que utiliza transformación ortogonal (en términos estadísticos: ortogonalidad = independencia) para transformar el conjunto de variables originales en un conjunto de nuevas variables llamadas PCs.

Cuando hablamos de PCs, no podemos dejar de mencionar los autovectores y autovalores. Un autovector es un vector que responde a una matriz como si esa matriz fuera un coeficiente escalar [13]. Si queremos realizar la transformación de la matriz normal **Figura 2.7(a)** a la matriz **Figura 2.7(b)**, los autovectores nos indicarán la dirección hacia donde la matriz está cambiando. En la **Figura 2.7(c)** podemos ver las matrices superpuestas.



(a) (b) (c)

Figura 2.7: Ejemplo de espacio de matrices extraído de [13]. (a) Matriz normal, (b) Matriz proyectada, (c) Matrices superpuestas.

La **Figura 2.8** muestra el comportamiento de los autovectores de la matriz **Figura 2.7(a)** siendo transformada a la matriz **Figura 2.7(b)**, donde se puede notar que los autovectores son aquellos vectores que no cambian de dirección, y que son perpendiculares entre sí.

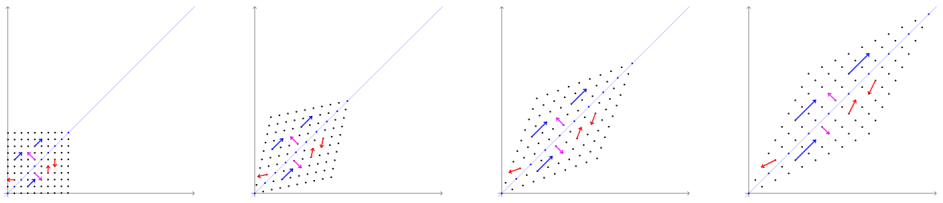
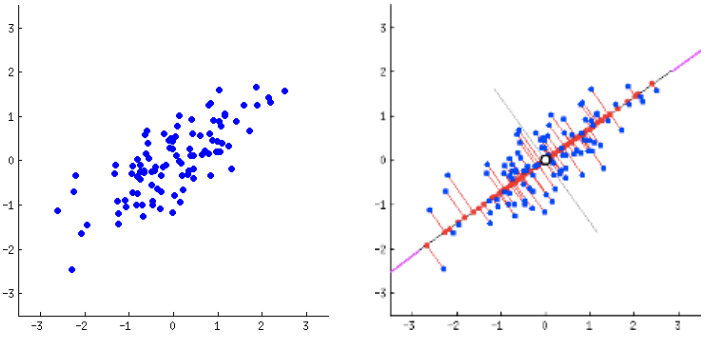


Figura 2.8: Transformación de espacio de una matriz [13]

A cada autovector le corresponde un autovalor, que indica la varianza asociada a dicho vector. Con el PCA lo que hacemos es obtener los autovectores ordenados por su varianza, es decir, ordenados por sus autovalores de forma descendente. Si consideramos una matriz de dimensión 2 con

datos dispersos, como se muestra en la **Figura 2.9(a)**, al aplicar PCA a los datos obtendremos los PCs marcados en rojo, el primer PC sería el vector que divide el diagrama de manera que explique la mayor varianza y el segundo el vector perpendicular al primero como se muestra en la **Figura 2.9(b)**.



(a) (b)

Figura 2.9: Gráficos de dispersión extraídos de [13]. (a) Gráfico de dispersión. (b) Componentes principales.

El autovector con el mayor autovalor representa la dirección en la que los datos tienen la máxima varianza. Es por esto, que si hallamos los autovectores de la matriz de covarianza de un conjunto de datos, obtenemos los PCs.

El *dataset* utilizado en este TFG consiste en N imágenes de m pixeles de ancho por m pixeles de alto correspondientes a galaxias cercanas del nuevo catálogo general utilizadas en [14]. Organizaremos nuestra matriz X en N filas u observaciones (en nuestro caso galaxias) y $m \times m$ columnas o variables (pixeles de las imágenes de galaxias remuestreadas en un vector de una sola dimensión). El PCA se lleva a cabo a través de la descomposición en valores singulares (SVD, por sus siglas en inglés) de la matriz de covarianzas ($p \times p$). Este proceso genera la siguiente matriz de covarianza

$$C = A^T A \tag{2.23}$$

donde A representa una fila de la matriz X. Esta matriz C, entonces, tendrá una dimensión de $p \times p$, y sabiendo que $p = m \times m$, determinar los p^2 autovectores y autovalores se vuelve una tarea impracticable. En [15] y [16] sugieren resolver este problema calculando la matriz de covarianza mediante la siguiente ecuación:

$$L = AA^T \tag{2.24}$$

ya que los autovectores de la matriz L serían una combinación lineal de los autovectores de la matriz C. *OpenCV* nos provee un método eficiente para obtener los PCs de una matriz con miles de columnas, por lo que hallaremos los PCs tanto del *dataset* original como de la traspuesta del mismo.

Un problema que podría tener el PCA es, que cuando las variables de la matriz de entrada no muestran correlación, los resultados no pueden ser muy buenos, por lo que para este caso, la técnica que se puede utilizar es una variante del PCA, el método ICA detallado a continuación.

2.2.3. Análisis de Componentes Independientes

El objetivo fundamental del ICA es proporcionar un método que permita encontrar una representación lineal de los datos no gaussianos de forma que las componentes sean estadísticamente independientes o lo más independientes posible. Una representación de este tipo permite obtener la estructura fundamental de los datos en muchas aplicaciones, incluidas la extracción de características y la separación de señales.

Existen varios métodos y algoritmos que permiten obtener la matriz de transformación ICA, entre las que podemos citar *Informax, Comon's, FastICA* y *JADE* [17]. En este TFG utilizaremos el algoritmo denominado *FastICA* propuesto por *Aapo Hyvärien* en la Universidad de Tecnología de Helsinki [18]. Es un algoritmo basado en el método del punto fijo y es adecuado para su realización en lenguajes de simulación matemática. Desde el punto de vista del rendimiento de los algoritmos que implementan ICA, se ha demostrado empíricamente que existen diferencias muy pequeñas y que todos obtienen un óptimo muy similar de ICs [19].

Mientras PCA maximiza la varianza, ICA minimiza mayores órdenes de dependencia. El número de variables debe ser mayor o igual al número de observaciones, para nuestro caso, la matriz de origen debe tener mayor o igual número de columnas que de filas. Al igual que en [19], realizaremos una implementación alternativa, en la que alimentaremos al algoritmo *FastICA* con la matriz de covarianza de la traspuesta del *dataset*.

2.3. Proceso de Clasificación

En esta fase, haremos uso de las técnicas de aprendizaje automático (*machine learning*). El aprendizaje automático es una rama de la IA, cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. Lo que hace es generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos, por tanto, es un proceso de inducción del conocimiento. En muchas ocasiones, este campo se solapa con el de la estadística, ya que las dos disciplinas se basan en el análisis de datos. El aprendizaje automático puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos.

Algunos sistemas de aprendizaje automático intentan eliminar toda necesidad de intuición o conocimiento experto de los procesos de análisis de datos, a estos se los denominan sistemas de aprendizaje automático no supervisados. Otros tratan de establecer un marco de colaboración entre el experto y la computadora, a estos se los denominan sistemas de aprendizaje automático supervisados.

En este TFG, utilizaremos los algoritmos de aprendizaje automático supervisado para clasificar las imágenes de galaxias disponibles en el *dataset*. Esta decisión se debe a que queremos clasificar las galaxias en base a la secuencia de *Hubble*, por lo que como expertos, tendremos que indicarle a los algoritmos de clasificación qué tipos de galaxias son las que se encuentran en el *dataset*, que serán utilizadas como datos de entrenamiento, y sirvan de aprendizaje para que puedan utilizar este conocimiento para poder clasificar nuevas imágenes no categorizadas.

Los resultados de los clasificadores son validados con la técnica de validación cruzada (*cross-validation*). Esta técnica es utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes

particiones. La validación cruzada es una manera de predecir el ajuste de un modelo a un hipotético conjunto de datos de prueba cuando no disponemos del conjunto explícito de datos de prueba. Existen varios tipos de validaciones cruzadas, en este TFG, utilizaremos la validación cruzada de K iteraciones o (*K-fold cross-validation*).

2.3.1. *Naïve Bayes*

El Naïve Bayes, también conocido como clasificador bayesiano ingenuo, es un clasificador probabilístico fundamentado en el teorema de *Bayes* y algunas hipótesis simplificadoras adicionales. Como consecuencia de estas simplificaciones se suelen resumir en la hipótesis de independencia entre las variables predictoras, que recibe el apelativo de ingenuo (*naïve*) [20]. *Naïve Bayes* asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, dada la clase variable.

2.3.2. Redes Bayesianas

Una red bayesiana es un modelo de grafo probabilístico que representa un conjunto de variables aleatorias y sus dependencias condicionales a través de un grafo acíclico dirigido. Los nodos del grafo representan variables aleatorias en el sentido de *Bayes*, las mismas pueden ser cantidades observables, variables latentes, parámetros desconocidos o hipótesis [21]. Las aristas representan dependencias condicionales. Los nodos que no se encuentran conectados representan variables las cuales son condicionalmente independientes de las otras. Cada nodo tiene asociado una función de probabilidad que toma como entrada un conjunto particular de valores de las variables padres del nodo y devuelve la probabilidad de la variable representada por el nodo.

2.3.3. Redes Neuronales

Son una base importante para el desarrollo de la IA. Son inspiradas en el comportamiento de las neuronas y conexiones del cerebro humano tratando de crear un programa, sistema o máquina que sea capaz de solucionar problemas difíciles, actuar de forma humana, y realizar trabajos pesados mediante técnicas algorítmicas convencionales [22].

2.3.4. Máquinas de Vectores de Soporte

Las máquinas de vectores de soporte (SVM, por sus siglas en inglés) son un conjunto de algoritmos de aprendizaje supervisado desarrollados por *Vladimir Vapnik* y su equipo en los laboratorios de AT&T. Una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases a 2 espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte [23]. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas a una o a la otra clase. Dado un conjunto de ejemplos de entrenamiento, podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra.

2.3.5. K-vecinos más Cercanos

El K-nn es un método de clasificación supervisada que sirve para estimar la función de densidad $F(\frac{x}{C_j})$ de las predictoras x por cada clase C_j [24]. Es un método de clasificación no paramétrico que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento x pertenezca a la clase C_j a partir de la información proporcionada por el conjunto de prototipos.

2.3.6. Árboles de Decisión

Es un modelo de predicción utilizado en el ámbito de la IA. Dado un *dataset*, se fabrican diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema [25]. En aprendizaje basado en árboles de decisión, se utiliza un árbol de decisión como modelo predictivo que mapea observaciones sobre un objeto a conclusiones sobre el valor objetivo del objeto. Los modelos de árbol, donde la variable de destino puede tomar un conjunto finito de valores se denominan árboles de clasificación.

Como aquí no profundizaremos en estos métodos, solo citaremos los algoritmos que utilizaremos para clasificar con la herramienta WEKA [26]. En la **Figura 2.10** vemos cómo están agrupados los algoritmos utilizados para clasificación en este TFG.

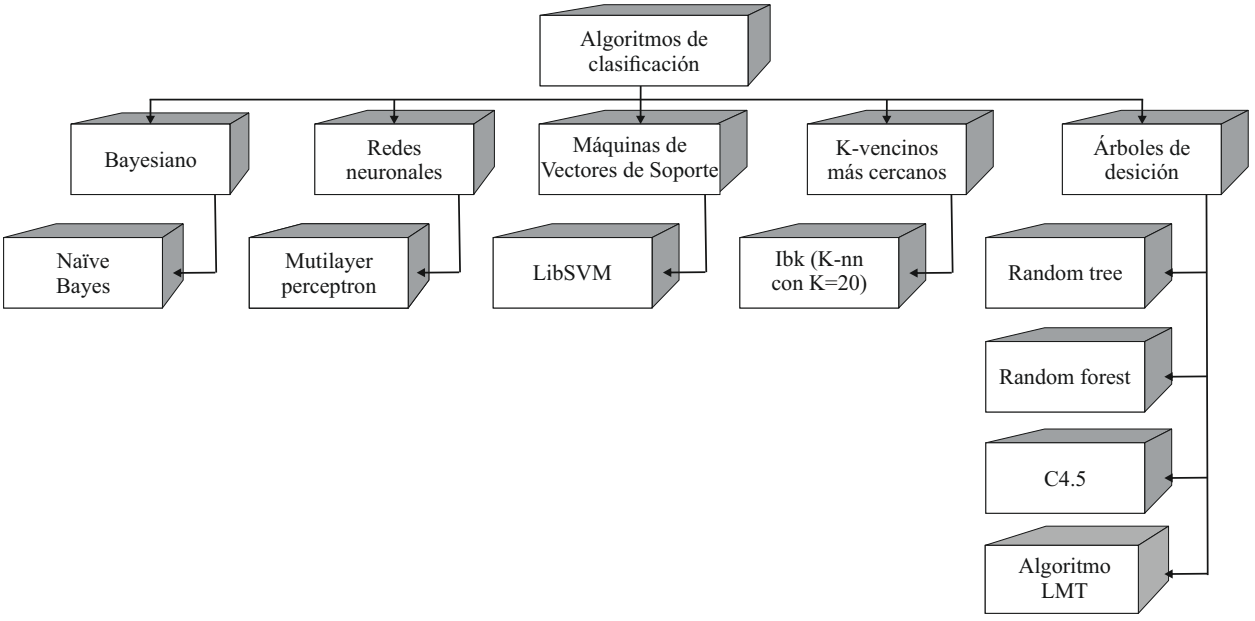


Figura 2.10: Algoritmos de clasificación agrupados por tipo

2.4. Resumen

En el presente capítulo se han presentado las fases del proceso de clasificación de imágenes de galaxias. En la fase de preprocesamiento se han abordado las distintas técnicas aplicadas a las imágenes para estandarizarlas, luego, en la fase de extracción de características se han presentado los distintos métodos de extracción tanto de características morfológicas como las no morfológicas. Se han explicado los tres métodos propuestos en el presente TFG para la extracción de MFs (Ph, Pv y FDL). Finalmente en la fase de clasificación, vimos los distintos algoritmos que pueden ser utilizados para realizar el proceso de clasificación dado un conjunto de datos de entrenamiento.

Capítulo 3

Sistema de Clasificación

El proceso de clasificación de galaxias se basa en técnicas de *machine learning*, por lo que hay que preparar los datos para que los algoritmos puedan aprender de éstas. Las técnicas de *machine learning* cuentan con tres pasos, selección de datos, procesamiento de datos y transformación de datos. La selección la realizamos manualmente, el procesamiento lo realizamos con *OpenCV* y la transformación lo hacemos utilizando los lenguajes de programación *Java* y R. Para cada paso, utilizamos las siguientes herramientas:

1 - *OpenCV*: Para el preprocesamiento de imágenes

2 - *Java*: Para la extracción de MFs y el PCA

3 - R: Para el ICA

4 - WEKA: Para ejecutar los algoritmos de clasificación

Las imágenes utilizadas en este TFG son las mismas que las utilizadas en [14], y consiste en 82 imágenes de galaxias cercanas del nuevo catálogo general. De estas galaxias, 66 son del tipo espiral (S), 14 son del tipo elípticas (E) y 2 pertenecen al tipo irregular (Irr). Cada imagen tiene una dimensión de 313 pixeles de alto por 313 pixeles de ancho. Estas imágenes están rotadas, no centradas y con ruido, por lo que la fase de preprocesamiento de las imágenes se realizó como se explica a continuación.

3.1. Preprocesamiento de Imágenes

Primeramente se realiza el preprocesamiento de la imagen, que consiste en la estandarización del formato de entrada. El **Algoritmo 1** muestra el pseudocódigo correspondiente para este proceso. El código fuente implementado se puede consultar en el **Apéndice A**: Preprocesamiento de imágenes.

Algoritmo 1 Preprocesamiento de Imágenes

1: *imgGris* ← leerImagenEscalaGrises()
2: *imgBinarizado* ← binarizarImagen(imgGris)
3: *imgFiltrado* ← filtroApertura(imgBinarizado)
4: *imgInterseccion* ← intersectarImagenes(imgGris, imgFiltrado)
5: *centroImagen* ← obtenerCentroObjeto(imgInterseccion)
6: *imgTrasladada* ← trasladarImagen(imgInterseccion, centroImagen, centroCuadro)
7: *imgRotada* ← alinearImagenHorizontal(imgTrasladada)

Luego de realizar este preprocesamiento, tenemos las 82 imágenes estandarizadas, sin ruidos, centradas y alineadas con la horizontal. La siguiente fase consiste en extraer las características. Utilizamos tres tipos de características: MF, PC e IC.

3.2. Extracción de Características

En esta sección se presenta el proceso de extracción de las características, tanto para las MFs, como para los PCs e ICs.

3.2.1. Características Morfológicas

El **Algoritmo 2** muestra el pseudocódigo correspondiente a la extracción de MFs. Desde la elongación hasta el ratio de radio, la característica es una simple ecuación matemática; para la firma de dispersión lumínica, se hace un recorrido por todos los pixeles de una imagen. El código fuente implementado se puede consultar en el **Apéndice A**: Extracción de MFs.

Algoritmo 2 Extracción de MFs

```
1: elongacion ← (ancho - alto) / (alto + ancho)
2: factorForma ← area / perimetro
3: convexidad ← perimetro / perimetroRectEnvolvente
4: factorFormaRectangular ← perimetroRectEnvolvente / area
5: indiceAsimetria ← (imgGris - imgRotada) / 256
6: picosHorizontales ← contarPicos(intersectar(imgGris, lineaHorizontal))
7: picosVerticales ← contarPicos(intersectar(imgGris, lineaVertical))
8: ratioCircularidad ← (4 × π × area)/perimetro^2
9: ratioForma ← (4 × area)/(π × ancho^2)
10: ratioCompacidad ← area/(π × radioExterior^2)
11: ratioRadio ← radioExterior/radioInterior
12:
13: centroImagen ← getCentro(imgRotada)
14: for x ∈ {1, ..., M} do
15:     for y ∈ {1, ..., M} do
16:         if centroImagen.x = x and centroImagen.y = y then
17:             total ← imgRotada(x, y)
18:         else
19:             difX = centroImagen.x − x
20:             difY = centroImagen.y − y
21:             if difX >= difY then
22:                 max = difX
23:             else
24:                 max = difY
25:             end if
26:             total ← total + ((1/(8 ∗ max)) ∗ imgRotada(y, x)
27:         end if
28:     end for
29: end for
30: fdl ← total
```

La extracción de PCs, se hizo en base a trabajos previos como [10, 9, 15].

3.2.2. Componentes Principales

Posteriormente se realiza la extración de los PCs, para ello se necesita construir una matriz *dataset* en la que cada fila corresponde a la imagen de una galaxia convertida en un vector de una sola dimensión, como se muestra en el **Algoritmo 3**. El código fuente implementado se puede consultar en el **Apéndice A**: Extracción de PCs: considerando todo el *dataset*.

Algoritmo 3 Extracción de PCs: considerando todo el *dataset*

for $i \in \{1, \ldots, N\}$ **do**
    $dataset \leftarrow$ `leerImagen()`
end for
$autovectores \leftarrow$ `pca(`$dataset$`)`
Hacer de $pcs[1, \ldots, N][1, \ldots, N]$ una nueva matriz
for $i \in \{1, \ldots, N\}$ **do**
    for $j \in \{1, \ldots, autovectores.filas\}$ **do**
        $pc = 0$
        for $j \in \{1, \ldots, autovectores.columnas\}$ **do**
            $pc = pc + (dataset[i][k] * autovectores[j][k])$
        end for
        $pcs[i][j] = pc$
    end for
end for

La salida de este algoritmo es una matriz de 82 filas (imágenes) por 82 columnas (PCs), donde cada fila contiene los PCs de una galaxia. En [15, 16] usan los autovectores como PCs, pero con el objetivo de reconstruir una imagen, no de clasificar, por lo que aquí probaremos si son útiles para realizar clasificación. Otro método de extracción de PCs que usaremos es calculando los autovectores de la matriz de covarianza de la traspuesta del *dataset* utilizando la ecuación (2.24). El **Algoritmo 4** es una variación del **Algoritmo 3**, donde el PCA es calculado a partir de la matriz de covarianza del *dataset*. El código fuente implementado se puede consultar en el **Apéndice A**: Extracción de PCs: considerando la traspuesta del *dataset*.

Algoritmo 4 Extracción de PCs: considerando la traspuesta del *dataset*

for $i \in \{1, \ldots, N\}$ **do**
    $dataset \leftarrow$ `leerImagen()`
end for
$traspuesta = dataset^T$
$covarianzas \leftarrow$ `covar(`$traspuesta$`)`
$autovectores \leftarrow$ `eigen(`$covarianzas$`)`
Hacer de $pcs[1, \ldots, N][1, \ldots, N]$ una nueva matriz
for $i \in \{1, \ldots, N\}$ **do**
    for $j \in \{1, \ldots, autovectores.filas\}$ **do**
        $pc = 0$
        for $j \in \{1, \ldots, autovectores.columnas\}$ **do**
            $pc = pc + (covarianzas[i][k] * autovectores[j][k])$
        end for
        $pcs[i][j] = pc$
    end for
end for

3.2.3. Componentes Independientes

La extracción de ICs la realizamos con el lenguaje R a partir de la matriz de covarianza de la traspuesta del *dataset* calculada con *Java*, la misma utilizada para el segundo método de extracción de PCs. En el **Algorimto 5** construimos la matriz *dataset* con las imágenes de las galaxias, calculamos la traspuesta del *dataset*, hallamos la matriz de covarianza de la traspuesta y a partir de allí hallamos los ICs. El código fuente implementado se puede consultar en el **Apéndice A**: Extracción de ICs.

Algoritmo 5 Extracción de ICs

1: **for** $i \in \{1, \ldots, N\}$ **do**
2: `    dataset ← leerImagen()`
3: **end for**
4: $traspuesta = dataset^T$
5: *covarianzas* ← `covar(`*traspuesta*`)`
6: *ica* ← `fastICA(`*covarianzas*`)`

3.3. Proceso de Clasificación

Para ejecutar los algoritmos de clasificación con WEKA, se requiere de una matriz de datos, donde las filas corresponden a las galaxias de nuestro *dataset*, y las columnas corresponden a los atributos de dichas galaxias, donde estos atributos dependen del método de extracción de características que utilicemos. Esta matriz la guardamos en formato CSV como se muestra en las siguientes tablas. En la **Tabla 3.1** vemos un ejemplo del archivo CSV donde se consideran las primeras 5 imágenes (n2683, n2715, n2768, n2775 y n3077) del *dataset* ordenadas alfabéticamente. Las columnas El, Conv, IA y RCir corresponden a 4 MFs de ejemplo calculadas para cada una de dichas galaxias y la última columna indica el tipo de clasificación al que pertenece. Cabe resaltar que esta última columna, es la que nosotros agregamos manualmente y como expertos en el tema definimos la categoría de cada imagen u observación.

Galaxia	El	Conv	IA	RCir	Tipo
n2683	0,534314	1,145729	0,042295	0,181692	s
n2715	0,498182	1,116556	0,10404	0,288267	s
n2768	0,410959	1,45445	0,049899	0,221978	e
n2775	0,109312	1,392022	0,055555	0,282845	s
n3077	0,1	0,993447	0,090259	0,55995	i

Tabla 3.1: Ejemplo de Matriz de Características Morfológicas

En la **Tabla 3.2** vemos los valores correspondientes a los PCs para cada galaxia ordenados por el valor de sus respectivos autovalores de forma descendente, podemos notar que hay valores positivos y negativos. Si la puntuación (*score*) es positiva, un valor más alto de esa puntuación es asociado con una mayor puntuación en el componente, y si la puntuación es negativa, entonces un valor más alto es asociado con una puntuación más baja en el componente.

Galaxia	PC1	PC2	PC3	PC4	Tipo
n2683	1312069158	87312225	−80796896	25203817	s
n2715	1245247167	62580924	−5198030	−9275295	s
n2768	1104224907	24236544	−9931471	−852261	e
n2775	1056757257	−46580226	36150165	8881730	s
n3077	1287226961	−8149457	−62517606	−10747336	i

Tabla 3.2: Ejemplo de Matriz de Componentes Principales

En la **Tabla 3.3** vemos los valores correspondientes a los ICs para cada galaxia. A diferencia de los PCs, los signos no tienen efecto ya que solo son considerados los valores absolutos.

Galaxia	IC1	IC2	IC3	IC4	Tipo
n2683	3,1785319	0,9509241	0,5147636	−0,4484955	s
n2715	1,6124588	2,8713187	0,8146027	−0,4632517	s
n2768	1,7659388	1,6480520	1,7528659	−0,5219518	e
n2775	−1,3355395	−0,8135287	−0,4530657	1,9292335	s
n3077	1,4521995	0,83737414	0,5738964	−0,0674617	i

Tabla 3.3: Ejemplo de Matriz de Componentes Independientes

Luego desde la aplicación WEKA, ejecutamos el *explorer* y en la pestaña *Preprocess* abrimos nuestro archivo CSV, a continuación pasamos a la pestaña *Classify*, con el botón *Choose* seleccionamos el algoritmo de clasificación que probaremos. En la sección *Test options* seleccionamos *Cross-validation* y ejecutamos la clasificación con el botón *Start*. En la sección *Classifier output* veremos el resultado del algoritmo, y el valor que esperamos es el de *Correctly Classified Instances*. Éste es el valor que nos interesa después de ejecutar una clasificación automática con un algoritmo dado, ya que con estos resultados podemos realizar comparaciones entre algoritmos de clasificación y entre los tipos de características utilizadas para cada algoritmo de clasificación.

3.4. Resumen

En este capítulo vimos los algoritmos utilizados en cada fase del proceso de clasificación automática de galaxias, también vimos cómo están construidos los archivos CSV para realizar el proceso de clasificación y mostramos los pasos realizados para evaluar la efectividad de un algoritmo de clasificación.

Capítulo 4

Resultados Experimentales

En este capítulo mostramos los resultados de aplicar clasificación con los distintos tipos de características estudiadas en este TFG. Una vez extraídas todas las MFs, PCs e ICs, procedemos a comparar los resultados. Para realizar la clasificación, agrupamos las galaxias en grupos de tres (S, E, Irr), cinco (E, S0, Sa+Sb, Sc+Sd, Irr) y siete (E, S0, Sa, Sb, Sc, Sd, Irr) clases, y evaluamos los distintos algoritmos de clasificación.

4.1. Clasificación Usando Características Morfológicas

Para comparar el desempeño de los algoritmos de clasificación con las MFs, probamos con cada característica por separado, y luego realizamos una selección de características cuya combinación nos daría mejores resultados que utilizándolas por separado. Para la selección de características utilizamos el método de selección de atributos de WEKA realizando búsqueda exhaustiva (*ExhaustiveSearch*) y como evaluador la selección de atributos basada en correlación (CFS, por sus siglas en inglés). Con esto, realizamos 2^n combinaciones de características, donde n indica el número de características disponibles, para así buscar la mejor combinación de las mismas, es decir, las que proveen el mejor porcentaje de aciertos. En la **Tabla 4.1** podemos ver los resultados de realizar clasificación con los atributos por separado y en la **Tabla 4.2** vemos los resultados de realizar selección de atributos con búsqueda exhaustiva. Podemos notar, que algunos atributos dan buenos porcentajes de aciertos con ciertos algoritmos de clasificación, y con otros algoritmos dan porcentajes de aciertos más bajos. También podemos ver que usando atributos combinados, podemos obtener un mejor porcentaje de aciertos que utilizando atributos de forma individual.

En la **Tabla 4.1** el porcentaje de aciertos más alto alcanzado, fue con la característica IA y el algoritmo de clasificación RF. En promedio, la características con mejores resultados es el IA, y los algoritmos con mejores resultados son SVM, MP, K-nn y LMT.

En la **Tabla 4.2** podemos ver que el máximo porcentaje de aciertos se logró combinando IA, PH y RCirc, junto con el algoritmo de clasificación MP alcanzando un máximo de 86,58 % de efectividad. Este tipo de búsqueda de combinaciones solo se puede realizar cuando la cantidad de atributos es relativamente pequeña, porque a medida que vamos agregando más atributos, el tiempo de ejecución se duplica por cada nueva característica. Combinar atributos para obtener mejores resultados tampoco es una regla, ya que podemos notar que algunos algoritmos devuelven porcentajes inferiores que utilizando atributos individuales. Por ejemplo, con los algoritmos C4.5 y RF obtenemos porcentajes más bajos combinando atributos que con atributos por separado como vemos en la **Tabla 4.2**.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
El	80.48	80.48	80.48	80.48	80.48	80.48	80.48	60.97	60.97
FF	80.48	80.48	80.48	80.48	80.48	80.48	80.48	65.85	65.85
Conv	80.48	73.17	80.48	80.48	80.48	79.26	80.48	70.73	70.73
FFR	80.48	80.48	80.48	80.48	80.48	80.48	80.48	67.07	67.07
IA	76.82	**81.70**	80.48	**82.92**	80.48	**82.92**	**82.92**	**84.14**	**84.14**
PH	79.26	78.04	**81.70**	80.48	**81.70**	78.04	**82.92**	79.26	81.70
PV	80.48	80.48	76.82	80.48	80.48	80.48	80.48	75.60	76.82
RCir	80.48	70.73	80.48	80.48	**81.70**	80.48	80.48	73.17	73.17
RF	80.48	80.48	80.48	80.48	80.48	80.48	80.48	60.97	60.97
RCom	80.48	79.26	80.48	79.26	80.48	76.82	**81.70**	71.95	71.95
RR	78.04	78.04	80.48	80.48	80.48	78.04	78.04	67.07	67.07
FDL	80.48	80.48	80.48	80.48	80.48	80.48	80.48	67.07	67.07

Tabla 4.1: Clasificación de galaxias S, E e Irr con atributos por separado

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
IA PH RCir	81.70	**85.36**	84.14	**86.58**	84.14	79.26	**85.36**	82.92	71.95

Tabla 4.2: Clasificación de galaxias S, E e Irr combinando los mejores atributos

A medida que aumentamos el número de tipos de galaxias que queremos clasificar, van disminuyendo nuestros porcentajes de aciertos. En la **Tabla 4.3**, vemos que el mejor porcentaje de aciertos se logró con IA y K-nn. En promedio, el atributo con mejores resultados fue IA, y el algoritmo con mejores resultados fue BN. En la **Tabla 4.4** podemos ver que con la combinación de los mejores atributos, no se logró un resultado mejor que el máximo de atributos individuales, aunque, en promedio, los atributos combinados tienen mejores porcentajes de aciertos.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
El	43.90	30.48	43.90	43.90	40.24	41.46	43.90	24.39	24.39
FF	43.90	45.12	40.24	45.12	42.68	41.46	43.90	24.39	24.39
Conv	43.90	42.68	43.90	36.58	36.58	43.90	32.92	37.80	37.80
FFR	43.90	42.68	43.90	45.12	41.46	43.90	41.46	21.95	21.95
IA	**53.65**	46.34	43.90	**52.43**	**57.31**	48.78	50.00	45.12	45.12
PH	42.68	47.56	47.56	46.34	43.90	41.46	**52.43**	**51.21**	45.12
PV	43.90	32.92	45.12	45.12	41.46	47.56	45.12	42.68	40.24
RCir	46.34	42.68	43.90	36.58	36.58	42.68	39.02	25.60	25.60
RF	43.90	31.70	37.80	40.24	40.24	41.46	43.90	23.17	23.17
RCom	43.90	43.90	43.90	45.12	43.90	48.78	43.90	28.04	28.04
RR	43.90	42.68	43.90	46.34	45.12	47.56	46.34	19.51	19.51
FDL	43.90	50.00	46.34	48.78	46.34	37.80	40.24	39.02	39.02

Tabla 4.3: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr con atributos por separado

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
IA RCir	**53.65**	41.46	45.12	47.56	**51.21**	**52.43**	47.56	37.80	32.92

Tabla 4.4: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr combinando los mejores atributos

En la **Tabla 4.5** podemos ver que el máximo porcentaje de aciertos se logró con las características IA y PH juntos con el algoritmo MP. En promedio, el atributo con mejores aciertos es IA y el algoritmo con mejor desempeño es el MP. En la **Tabla 4.6** vemos que la combinación de los mejores atributos no supera al máximo logrado con los atributos por separado.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
El	24.39	13.41	21.95	26.82	15.85	25.60	24.39	18.29	18.29
FF	24.39	19.51	26.82	26.82	28.04	31.70	31.70	24.39	24.39
Conv	24.39	25.60	21.95	32.92	26.82	30.48	26.82	19.51	19.51
FFR	24.39	19.51	23.17	21.95	28.04	19.51	19.51	10.97	10.97
IA	30.48	**35.36**	26.82	**42.68**	**39.02**	**34.14**	**39.02**	**34.14**	**34.14**
PH	24.39	31.70	30.48	**42.68**	**39.02**	28.04	29.26	28.04	29.26
PV	24.39	30.48	21.95	30.48	32.92	24.39	32.92	21.95	24.39
RCir	25.60	21.95	23.17	**36.58**	**36.58**	31.70	29.26	18.29	18.29
RF	24.39	26.82	18.29	24.39	20.73	20.73	23.17	13.41	13.41
RCom	24.39	26.82	25.60	30.48	29.26	26.82	29.26	20.73	20.73
RR	24.39	32.92	26.82	31.70	31.70	23.17	25.60	17.07	17.07
FDL	24.39	26.82	29.26	24.39	30.48	29.26	23.17	31.70	31.70

Tabla 4.5: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr con atributos por separado

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
IA RCir	31.70	29.26	24.39	**34.14**	30.48	30.48	**37.80**	31.70	30.48

Tabla 4.6: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr combinando los mejores atributos

4.2. Clasificación Usando Componentes Principales

Para comparar el desempeño de los algoritmos de clasificación con los PCs, probamos con cada PC por separado, y luego realizamos una selección de PCs cuya combinación nos daría mejores resultados que utilizándolos por separado. Para la selección de PCs utilizamos el método de selección de atributos de WEKA realizando búsqueda con el algoritmo "*GreedyStepwise*", ya que no podemos realizar una búsqueda exhaustiva debido al tiempo que tomaría evaluar las 2^{82} combinaciones posibles. Por esto, tal vez los mejores resultados alcanzados con este algoritmo de búsqueda no sean

los mejores que se podrían alcanzar con una búsqueda exhaustiva.

Existen tantos PCs como imágenes en el *dataset*, pero los PCs calculados están ordenados por su relevancia, esto es, por el porcentaje de variabilidad que determinan sus autovalores. Como los primeros cinco componentes determinan el 93,97 % de la variabilidad de los datos, solo utilizamos los cinco primeros componentes para realizar las pruebas individuales, luego realizamos una evaluación considerando los 82 PCs y también realizamos selección de atributos.

Como utilizamos dos métodos para extracción de PCs, en las **Tablas 4.7** al **4.15** mostramos los resultados de utilizar los PCs del *dataset* y en las **Tablas 4.16** al **4.24** mostramos los resultados de utilizar los PCs de la traspuesta del *dataset*. Las **Tablas 4.7, 4.8** y **4.9** muestran los resultados de clasificar las galaxias considerando tres tipos, S, E e Irr.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC1	80.49	80.49	80.49	80.49	80.49	80.49	80.49	65.85	65.85
PC2	80.49	80.49	80.49	80.49	80.49	80.49	80.49	62.20	62.20
PC3	80.49	79.27	80.49	80.49	80.49	80.49	80.49	67.07	67.07
PC4	80.49	78.05	80.49	80.49	80.49	80.49	80.49	71.95	71.95
PC5	80.49	80.49	79.27	80.49	80.49	80.49	80.49	69.51	69.51

Tabla 4.7: Clasificación de galaxias E, S e Irr con los PCs por separado

En la **Tabla 4.7** vemos que la mayoría de los algoritmos de clasificación tienen una efectividad similar, alcanzando aciertos de 80,49 %, pero los algoritmos RF y RT tienen rendimientos más bajos. En la **Tabla 4.8** podemos ver que el algoritmo NB destaca sobre los demás con un nivel de acierto del 82,92 %, y los algoritmos que mostraban un rendimiento bajo cuando consideramos los PCs por separado, RF y RT, mejoran su rendimiento al considerar todos los PCs.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
Todos	78.04	**82.92**	80.48	80.48	80.48	76.82	74.39	**81.70**	80.48

Tabla 4.8: Clasificación de galaxias E, S e Irr usando todos los PCs

En la **Tabla 4.9** vemos los resultados de la clasificación obtenidos con los PCs seleccionados utilizando selección de atributos. Podemos notar que el algoritmo NB mantiene una pequeña ventaja sobre los demás y sólo NB, SVM y K-nn obtienen resultados superiores a 80 %. Los componentes seleccionados dependen del algoritmo *GreedyStepwise* utilizado para realizar la búsqueda de los mejores atributos.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC15 PC75 PC77 PC79	78.04	**82.92**	80.48	78.04	80.48	76.82	78.04	78.04	73.17

Tabla 4.9: Clasificación de galaxias E, S e Irr seleccionando los mejores PCs

En las **Tablas 4.10, 4.11** y **4.12** vemos los resultados de clasificar las galaxias considerando cinco tipos, E, S0, Sa+Sb, Sc+Sd e Irr. En la **Tabla 4.10** vemos los resultados de clasificar considerando los PCs por separado, el mejor resultado se alcanzó con el algoritmo K-nn, sin embargo, en promedio no fue el mejor, ya que quienes alcanzaron el promedio más alto fueron los algoritmos BN y LMT. También podemos ver que los algoritmos RF y RT tienen muy bajo rendimiento.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC1	47.56	45.12	43.90	43.90	43.90	46.34	50.00	36.59	36.59
PC2	43.90	36.59	42.68	39.02	36.59	37.80	40.24	35.37	35.37
PC3	43.90	43.90	41.46	45.12	**51.22**	48.78	47.56	32.93	32.93
PC4	43.90	40.24	43.90	41.46	37.80	41.46	42.68	31.71	31.71
PC5	43.90	42.68	43.90	41.46	35.37	45.12	40.24	39.02	39.02

Tabla 4.10: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr con los PCs por separado

En la **Tabla 4.11** vemos el resultado utilizando todos los PCs, y podemos notar que al igual que para la clasificación de tres tipos, el algoritmo con mejor resultado es el NB alcanzando 56,1 % y los algoritmos RF y RT mejoraron sus resultados.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
Todos	46.34	**56.10**	43.90	43.90	43.90	42.68	41.46	**53.66**	42.68

Tabla 4.11: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr usando todos los PCs

En la **Tabla 4.12** vemos los resultados luego de aplicar selección de atributos. En este caso, es el algoritmo RF el que obtiene el mejor resultado, alcanzando 56,1 % de aciertos, igualando al resultado del NB con todos los PCs.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC1 PC15 PC61 PC72 PC75 PC79 PC80	46.34	**51.22**	43.90	43.90	37.80	50.00	46.34	**56.10**	51.22

Tabla 4.12: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr seleccionando los mejores PCs

En las **Tablas 4.13, 4.14** y **4.15** vemos los resultados de clasificar las galaxias considerando siete tipos, E, S0, Sa, Sb, Sc, Sd e Irr. En la **Tabla 4.13** vemos los resultados de clasificar considerando los PCs por separado, los porcentajes de aciertos más altos se alcanzaron con los algoritmos NB, MP y K-nn, pero el algoritmo con mejor promedio es el NB.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC1	23.17	31.71	24.39	26.83	28.05	20.73	30.49	17.07	17.07
PC2	24.39	28.05	29.27	26.83	20.73	26.83	28.05	20.73	20.73
PC3	24.39	**34.15**	23.17	32.93	32.93	23.17	29.27	23.17	23.17
PC4	24.39	30.49	25.61	**34.15**	**34.15**	21.95	19.51	15.85	15.85
PC5	23.17	29.27	29.27	24.39	25.61	25.61	23.17	23.17	23.17

Tabla 4.13: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr con los PCs por separado

En la **Tabla 4.14** vemos los resultados de clasificar considerando todos los PCs, el porcentaje más alto se logró con el algoritmo NB, seguido por el RF. El resto de algoritmos dieron resultados muy bajos.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
Todos	23.17	**43.90**	24.39	26.83	24.39	29.27	26.83	**39.02**	21.95

Tabla 4.14: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr usando todos los PCs

En la **Tabla 4.15** vemos el resultado de clasificar utilizando selección de atributos, el algoritmo con mejor resultado es el NB pero el porcentaje de aciertos es muy inferior que considerando todos los PCs.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC35 PC80	23.17	**34.15**	24.39	21.95	28.05	21.95	20.73	24.39	21.95

Tabla 4.15: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr seleccionando los mejores PCs

Los PCs no tienen el mismo comportamiento en un sistema de clasificación como lo tienen en un sistema de reconstrucción de imágenes. El PCA se hace con la intención de reducir la dimensionalidad, pero como podemos ver en estas Tablas, usar el segundo, tercero, cuarto o quinto componente principal, no tiene casi diferencias con respecto al primero. Usar todos los PCs disponibles, o los atributos seleccionados podrían mejorar los resultados, pero tampoco significa que la combinación de atributos sea siempre mejor que un atributo individual. No podemos determinar de forma anticipada cuál sería la combinación perfecta de PCs que nos ayude a obtener los mejores resultados. Podemos visualizar los autovectores generados por el PCA en la **Figura 4.1**. Estas figuras son un equivalente a los *eigenfaces* nombrados por [15], solo que en este caso, no corresponde a rostros, sino a galaxias, por lo que los podemos nombrar *eigengalaxias*.

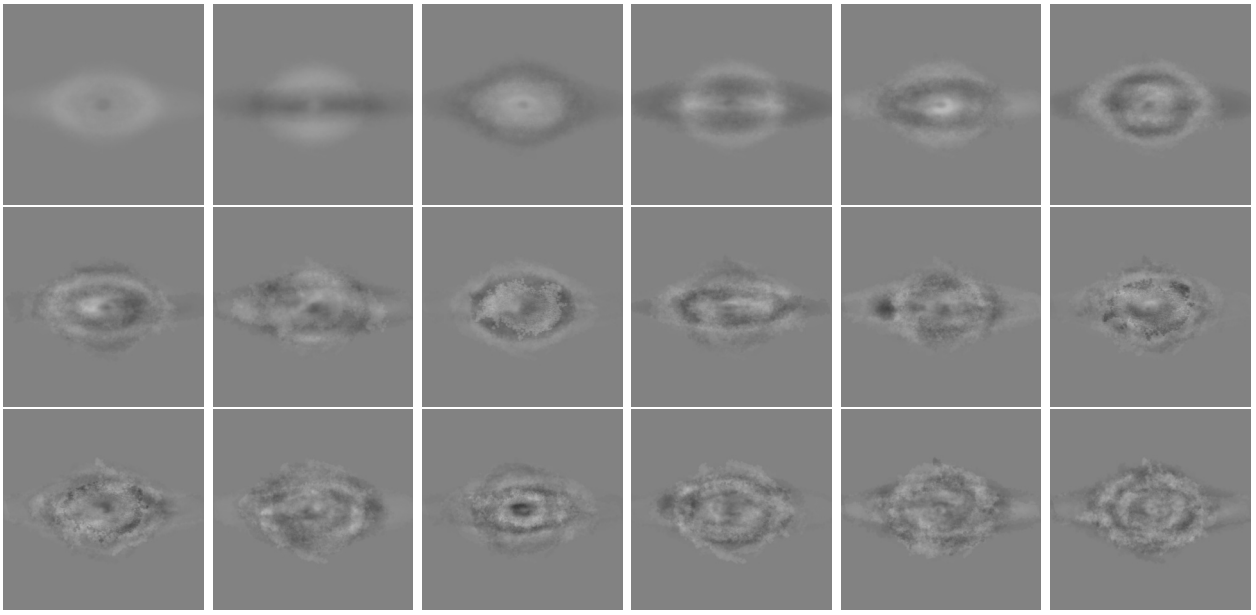


Figura 4.1: Primeros 16 autovectores del *dataset*. Eigengalaxias

En las **Tablas 4.16** al **4.24** vemos los resultados de aplicar PCA a la traspuesta del *dataset*. Como las pruebas anteriores, el comportamiento es el mismo, devuelve buenos resultados al clasificar pocos tipos de galaxias pero va disminuyendo su efectividad a medida que aumentamos los tipos de galaxias que queremos clasificar. Las **Tablas 4.16, 4.17** y **4.18** muestran los resultados de clasificar las galaxias considerando tres tipos, S, E e Irr. En la **Tabla 4.16** se consideran los PCs por separado, y como se puede ver, los resultados son casi iguales a los de la **Tabla 4.7**, siendo igualmente los algoritmos RF y RT los de resultados más bajos.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC1	80.49	80.49	80.49	80.49	80.49	80.49	80.49	62.20	62.20
PC2	80.49	80.49	80.49	80.49	80.49	79.27	80.49	76.83	76.83
PC3	80.49	80.49	80.49	79.27	80.49	79.27	80.49	69.51	69.51
PC4	80.49	79.27	80.49	80.49	80.49	80.49	80.49	71.95	71.95
PC5	80.49	79.27	80.49	80.49	80.49	76.83	80.49	69.51	69.51

Tabla 4.16: Clasificación de galaxias E, S e Irr con los PCs por separado

La **Tabla 4.17** sí muestra una mejora respecto a la **Tabla 4.8**, se puede ver que el resultado del algoritmo NB es 6,28 % mejor, alcanzando un 89,02 % de aciertos. También podemos ver que los algoritmos que tienen un bajo rendimiento cuando consideramos los PCs por separado, como RF y RT, tienen mejores resultados al considerar todos los PCs.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
Todos	73.17	**89.02**	80.49	**81.71**	80.49	75.61	70.73	**81.71**	80.49

Tabla 4.17: Clasificación de galaxias E, S e Irr usando todos los PCs

En la **Tabla 4.18** vemos los resultados de luego de aplicar selección de atributos. Los algoritmos con mejores resultados fueron SVM y K-nn alcanzando 80,49 % de efectividad.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC3 PC11 PC15 PC42 PC76 PC78 PC80	73.17	79.27	80.49	76.83	80.49	73.17	76.83	78.05	76.83

Tabla 4.18: Clasificación de galaxias E, S e Irr seleccionando los mejores PCs

Las **Tablas 4.19, 4.20** y **4.21** muestran los resultados de clasificar las galaxias considerando cinco tipos, E, S0, Sa+Sb, Sc+Sd e Irr. En la **Tabla 4.19** vemos el resultado de la clasificación considerando los PCs por separado. Podemos notar que el mejor porcentaje de aciertos se logró con el algoritmo K-nn, sin embargo, los algoritmos con mejores promedios son NB y LMT.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC1	37.80	45.12	43.90	41.46	48.78	40.24	48.78	31.71	31.71
PC2	43.90	43.90	43.90	41.46	41.46	37.80	40.24	42.68	42.68
PC3	37.80	47.56	43.90	43.90	48.78	34.15	48.78	37.80	37.80
PC4	43.90	41.46	43.90	39.02	34.15	42.68	42.68	25.61	25.61
PC5	43.90	43.90	43.90	37.80	42.68	37.80	40.24	37.80	37.80

Tabla 4.19: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr con los PCs por separado

En la **Tabla 4.20** vemos el resultado de la clasificación considerando todos los PCs, y los mejores resultados logrados fueron con los algoritmos NB y RF, alcanzando 57,32 % de efectividad.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
Todos	43.90	**57.32**	43.90	43.90	43.90	**53.66**	41.46	**57.32**	29.27

Tabla 4.20: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr usando todos los PCs

En la **Tabla 4.21** vemos los resultados de la clasificación luego de realizar selección de atributos. En este caso, el mejor porcentaje de aciertos se logró con el NB alcanzando 59,76 % de efectividad. Pero si comparamos con la **Tabla 4.12**, podemos ver que no siempre la selección de atributos nos devolverá un resultado mejor que considerando todos los PCs.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC1 PC3 PC62 PC73 PC76 PC78 PC80 PC81	43.90	**59.76**	43.90	40.24	50.00	42.68	47.56	50.00	36.59

Tabla 4.21: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr seleccionando los mejores PCs

Las **Tablas 4.22, 4.23** y **4.24** muestran los resultados de clasificar las galaxias considerando siete tipos, E, S0, Sa, Sb, Sc, Sd e Irr. En la **Tabla 4.22** vemos los resultados de la clasificación considerando los PCs por separado, tanto el mejor resultado como el mejor promedio se alcanzó con el algoritmo K-nn con un máximo de 40,24 % de aciertos.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC1	24.39	29.27	24.39	30.49	26.83	17.07	30.49	17.07	17.07
PC2	24.39	25.61	24.39	21.95	24.39	18.29	26.83	25.61	25.61
PC3	24.39	31.71	24.39	31.71	28.05	21.95	24.39	20.73	20.73
PC4	24.39	31.71	24.39	**37.80**	32.93	30.49	**35.37**	29.27	29.27
PC5	24.39	15.85	24.39	21.95	**40.24**	26.83	23.17	21.95	21.95

Tabla 4.22: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr con los PCs por separado

En la **Tabla 4.23** vemos los resultados de clasificar considerando todos los PCs. El algoritmo con mejor resultado es el NB con 41,46 % de aciertos seguido por LMT con 39,02 % de aciertos.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
Todos	30.49	**41.46**	24.39	**34.15**	24.39	32.93	**39.02**	**35.37**	26.83

Tabla 4.23: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr usando todos los PCs

En la **Tabla 4.24** podemos notar algo muy particular, y es que la selección de atributos devolvió un solo atributo como mejor combinación, dando al NB el mejor resultado.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
PC81	30.49	**37.80**	24.39	28.05	30.49	25.61	**34.15**	29.27	26.83

Tabla 4.24: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr seleccionando los mejores PCs

Podemos notar que si consideramos la traspuesta de la matriz del *dataset*, tenemos mejores resultados al clasificar tres y cinto tipos de galaxias que si consideramos el *dataset* original, además, el cálculo de los PCs es mucho más simple porque la matriz de covarianza solo tiene 82 filas por 82 columnas, pero al clasificar siete tipos de galaxias, tuvimos resultados más bajos que considerando el *dataset* original. También podemos notar que los algoritmos RF y RT mejoran sustancialmente cuando se consideran todos los PCs. Algo llamativo es que el algoritmo SVM mantiene casi el mismo porcentaje de aciertos tanto para PCs por separado, todos los PCs juntos o con PCs obtenidos con selección de atributos. Por más que los resultados del SVM no sean los mejores, son buenos resultados y su constancia sería una ventaja para utilizarlo con PCs.

4.3. Clasificación Usando Componentes Independientes

En las **Tablas 4.25** al **4.33** vemos los resultados de utilizar ICs para realizar la clasificación. Las **Tablas 4.25, 4.26** y **4.27** muestran los resultados de clasificar las galaxias considerando tres tipos, S, E e Irr. En la **Tabla 4.25** vemos los resultados de clasificar considerando los ICs por separado. Los porcentajes son similares a los obtenidos utilizando PCs, por lo que no se puede apreciar ninguna ventaja o mejora.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
IC1	80.49	78.05	80.49	80.49	80.49	80.49	80.49	74.39	74.39
IC2	80.49	80.49	80.49	80.49	80.49	80.49	80.49	74.39	74.39
IC3	80.49	79.27	80.49	80.49	80.49	80.49	80.49	80.49	80.49
IC4	80.49	79.27	80.49	80.49	80.49	80.49	80.49	73.17	73.17
IC5	80.49	80.49	80.49	76.83	80.49	74.39	79.27	73.17	73.17

Tabla 4.25: Clasificación de galaxias E, S e Irr con los ICs por separado

En la **Tabla 4.26** vemos los resultados de clasificar considerando todos los ICs, algo notable es que el algoritmo NB que suele dar buenos resultados con PCs, tiene muy bajo porcentaje de aciertos con ICs. En este caso, los algoritmos MP y RF son los que devuelven mejores porcentajes de aciertos, alcanzando 82,93 % de aciertos.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
Todos	71.95	65.85	80.49	**82.93**	80.49	80.49	80.49	**82.93**	74.39

Tabla 4.26: Clasificación de galaxias E, S e Irr usando todos los ICs

En la **Tabla 4.27** vemos los resultados de la clasificación luego de realizar selección de atributos. En esta ocasión, podemos ver que el mejor resultado se obtuvo con el algoritmo C4.5.

	3-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
IC2 IC18 IC34 IC66 IC76 IC80	79.27	60.98	80.49	78.05	80.49	81.71	78.05	78.05	71.95

Tabla 4.27: Clasificación de galaxias E, S e Irr seleccionando los mejores ICs

Las **Tablas 4.28, 4.29** y **4.30** muestran los resultados de clasificar las galaxias considerando cinco tipos, E, S0, Sa+Sb, Sc+Sd e Irr. En la **Tabla 4.28** vemos los resultados de la clasificación considerando los ICs por separado. Los mejores resultados se obtuvieron con los algoritmos MP y K-nn, alcanzando 54,88 % de aciertos. El algoritmo con mejor promedio es el K-nn. De nuevo, los algoritmos RF y RT, al igual que con PCs, tienen muy baja efectividad.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
IC1	45.12	48.78	50.00	41.46	**54.88**	41.46	46.34	34.15	34.15
IC2	43.90	43.90	42.68	36.59	40.24	39.02	42.68	34.15	34.15
IC3	43.90	**51.22**	40.24	43.90	46.34	42.68	43.90	36.59	36.59
IC4	43.90	34.15	43.90	41.46	37.80	41.46	37.80	25.61	25.61
IC5	40.24	**52.44**	47.56	**54.88**	**53.66**	47.56	**53.66**	35.37	35.37

Tabla 4.28: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr con los ICs por separado

En la **Tabla 4.29** vemos los resultados de la clasificación considerando todos los ICs. El mejor resultado se alcanza con el algoritmo SVM, llegando a 54,88 % de aciertos.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
Todos	47.56	45.12	**54.88**	47.56	43.90	50.00	**52.44**	47.56	48.78

Tabla 4.29: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr usando todos los ICs

En la **Tabla 4.31** vemos los resultados luego de aplicar selección de atributos, en este caso, el mejor porcentaje de aciertos se logró con el algoritmo SVM alcanzando 52,44 %.

	5-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
IC18 IC34 IC68 IC80	43.90	45.12	**52.44**	47.56	47.56	46.34	45.12	48.78	36.59

Tabla 4.30: Clasificación de galaxias E, S0, Sa+Sb, Sc+Sd e Irr seleccionando los mejores ICs

Las **Tablas 4.31, 4.32** y **4.33** muestran los resultados de clasificar las galaxias considerando siete tipos, E, S0, Sa, Sb, Sc, Sd e Irr. En la **Tabla 4.31** vemos los resultados de la clasificación considerando los ICs por separado, dando mejores resultados con los algoritmos NB y K-nn, alcanzando 40,24 % de aciertos.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
IC1	21.95	**35.37**	**34.15**	30.49	**35.37**	24.39	**36.59**	14.63	14.63
IC2	24.39	**36.59**	28.05	32.93	26.83	30.49	25.61	21.95	21.95
IC3	24.39	**40.24**	25.61	**36.59**	**40.24**	30.49	29.27	23.17	23.17
IC4	24.39	28.05	30.49	30.49	29.27	15.85	19.51	20.73	20.73
IC5	23.17	31.71	**34.15**	30.49	32.93	29.27	32.93	26.83	26.83

Tabla 4.31: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr con los ICs por separado

En la **Tabla 4.32** vemor los resultados de la clasificación considerando todos los ICs. El mejor resultado se logró con el algoritmo LMT llegando a alcanzar 39,02 % de aciertos.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
Todos	23.17	32.93	**34.15**	32.93	24.39	30.49	**39.02**	**34.15**	**36.59**

Tabla 4.32: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr usando todos los ICs

En la **Tabla 4.33** vemos que hay un solo IC luego de la selección de atributos, dando el mejor resultado con el algoritmo LMT, con 35,37 % de aciertos.

	7-class								
	BN	NB	SVM	MP	K-nn	C4.5	LMT	RF	RT
IC80	23.17	**34.15**	29.27	32.93	29.27	31.71	**35.37**	21.95	18.29

Tabla 4.33: Clasificación de galaxias E, S0, Sa, Sb, Sc, Sd e Irr seleccionando los mejores ICs

Los ICs tienen el mismo comportamiento que los PCs para la clasificación, con la diferencia que la calidad de los resultados son de menor calidad.

4.4. Resumen

En este capítulo vimos los resultados experimentales de realizar el proceso de clasificación para todos los tipos de características considerados y los distintos algoritmos de clasificación utilizados. Vimos que los resultados varían de acuerdo a la combinación de *algoritmo de clasificación - tipos de galaxias - características* y cuáles son las mejores y peores combinaciones. También vimos que la mayoría de los algoritmos mejoran cuando se consideran todas las características combinadas y no por separado.

Capítulo 5

Conclusiones y Trabajos Futuros

5.1. Conclusiones

Este Trabajo de Fin de Grado ha abordado el estudio de la clasificación automática de galaxias. Las principales contribuciones y conclusiones se resumen a continuación:

- La clasificación de galaxias es un tema actual de estudio, la calidad de los resultados siempre dependerá del previo preprocesamiento de imágenes y extracción de características, como así también la selección de buenos datos de entrenamiento.
- Las MFs más efectivas resultaron ser aquellas que tienen en cuenta el interior de la imagen, como IA y PH, y no solo la forma exterior de los mismos, como El, Conv y el RForm. La propuesta realizada FDL es una característica que tiene en cuenta el interior de la imagen, pero no resultó ser mejor que IA o PH, esto podría ser porque solo tiene en cuenta la distancia de los pixeles al centro de la imagen y no su orientación.
- Los PCs con mayor varianza dan resultados más acertados que los PCs con menor varianza, aunque en estas pruebas no pudimos notar tanta diferencia debido al tamaño de nuestro *dataset* y también porque los datos están desbalanceados. La cantidad de PCs que se deben considerar dependen de los autovalores, donde un autovalor mayor a 1 indica que el PC representa mayor varianza que la contabilidaza por una de las variables originales. Algo que podemos resaltar es la mejora en la calidad de los resultados si utilizamos la traspuesta del *dataset*, y esto es algo llamativo, ya que los PCs son los autovectores de la matriz de covarianza, por lo que en este caso, la matriz de covarianza L sería una matriz de dimensiones 82 x 82 y procesar esta matriz es bastante rápido.
- Los ICs muestran el mismo comportamiento que los PCs, solo que con una calidad de resultados un poco inferior.
- Los resultados de estas pruebas están sujetas a los datos de entrenamiento, y los porcentajes de aciertos podrían ser un poco sensibles debido a la cantidad de tipos de galaxias que utilizamos, por lo que para tener una mejor evaluación, los datos de entrenamiento deberían ser más equilibrados, es decir, poseer un *dataset* con cantidades equivalentes de galaxias espirales, elípticas e irregulares. También sería conveniente utilizar más imágenes de entrenamiento en el *dataset*.
- Entre las MFs, los PCs y los ICs, son las MFs las que nos aseguran una buena clasificación. La cuestión sería utilizar una MF o una combinación de características que arrojen buenos

resultados con los datos de entrenamientos, y para esto, deberíamos escoger características que tengan en cuenta la superficie de la imagen.

- Una buena característica es aquella que separa correctamente a las observaciones de una muestra, por ejemplo, si tenemos un grupo de 10 personas adultas y 10 niños, la altura sería una buena característica ya que sirve para diferenciar los dos grupos, de esta forma, dada la altura de un individuo, con dicho dato se podría clasificar fácilmente a qué grupo o clase pertenece. Para el caso de las galaxias, es necesario una característica que diferencie bien entre espirales, elípticas e irregulares, y las características estudiadas en este TFG, aunque realizan una buena separación entre clases, existen muchos valores atípicos (*outliers*) que hacen que los clasificadores confundan los tipos.
- Si sumamos los promedios de los algoritmos de clasificación, el *Naïve Bayes* es el que provee los mejores resultados en comparación a los demás métodos, con 10 promedios superiores, seguido por SVM, K-nn, LMT y RF con 3 promedios superiores cada uno, BN y C4.5 solo con 1 promedio superior cada uno. El algoritmo RT no fue mejor en ningún caso. Algo que también podemos notar es que el método SVM es más efectivo con ICs.
- Los algoritmos RF y RT muestran bajos niveles de efectividad cuando se consideran atributos por separado, pero si se considera una combinación de MFs o muchos PCs o ICs mejoran sustancialmente. El algoritmo SVM muestra un comportamiento muy estable, devolviendo buenos valores para la mayoría de los casos.
- Como resultado del TFG se ha realizado la presentación de un Póster Científico en la prestigiosa Conferencia de Ciencias Computacionales Interdisciplinarias *(CCIS 2016)*. Se adjunta el Póster en el **Apéndice B**.

5.2. Trabajos Futuros

El ICA es un tema de investigación que ha sido escasamente abordado en este contexto. Por dicho motivo existen algunos tópicos que pueden profundizarse dentro de esta línea de investigación, entre los que se citan los más relevantes a juicio de este alumno:

- El ICA fue imposible de calcular para todo el *dataset* por la insuficiencia de memoria, por eso solo se pudo calcular a partir de la matriz de covarianza de la traspuesta del *dataset*, siguiendo el segundo método de extracción de PCs. Por dicho motivo, se podría investigar una forma más efectiva de poder extraer los ICs de la matriz original del *dataset*.
- En cuanto al PCA, se podrían realizar pruebas hallando los PCs de imágenes en distintas longitudes de onda, es decir, tener varias capas de la misma imagen, pero correspondiente a observaciones de telescopios diferentes, y no los PCs del *dataset*. También hay que cuidar la cantidad de variables a utilizar con el *Naïve Bayes*, ya que agregando más variables puede hacer que disminuya su efectividad debido a datos redundantes.
- En cuanto a los MFs, se podrían buscar nuevos métodos de extracción, que estén centrados en el contenido interior de la imagen y no solo en su forma, considerar la intensidad de los pixeles también es una buena decisión.
- Para tener una mejor estimación de la efectividad de los extractores de características y los algoritmos de clasificación utilizados en este TFG, se podría realizar las mismas pruebas pero con el *dataset* del SDSS que contiene miles de imágenes de galaxias.

Bibliografía

[1] Encyclopedia Britannica, "Big-bang model". https://www.britannica.com/topic/big-bang-model. Retrieved August, 2016.

[2] David L. Block, "Georges Lemaître and Stigler's Law of Eponymy". 2012.

[3] Baugh, C.; Frenk, C., "How are galaxies made?", physicsweb.org, 1999.

[4] Harper, D. "galaxy". Online Etymology Dictionary. Retrieved November, 2016.

[5] Marmet L., "On the Interpretation of Red-Shifts: A Quantitative Comparison of Red-Shift Mechanisms II". 2016.

[6] Hubble E., "A relation between distance and radial velocity among extra-galactic nebulae". Mount Wilson Observatory, Carnegie Institution of Washington. 1929

[7] Eggen, O. J.; Lynden-Bell, D.; Sandage, A. R., "Evidence from the motions of old stars that the Galaxy collapsed", Astrophysical Journal, vol. 136, p. 748, November, 1962.

[8] B. A. Vorontsov-Velyaminov, "The Large Scale Structure of the Universe", International Astronomical Union. 1978.

[9] S. Kasivajhula, N. Raghavan, H. Shah. "Morphological Galaxy Classification Using Machine Learning". 2007.

[10] J. de la Calleja, O. Fuentes. "Automated Classification of Galaxy Images," Instituto Nacional de Astrofísica, Óptica y Electrónica, Luís Enrique Erro 1. 2004.

[11] International Telecommunication Union, "Recommendation BT. 601", Available from: https://www.itu.int. Retrieved August, 2016.

[12] K. E. Selkirk, "Pattern and Place: An Introduction to the Mathematics of Geography", Cambridge Univ. Press, Cambridge, New York, 1982.

[13] Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org. Retrieved August, 2016.

[14] Z. Frei, "Automatic morphological classification of galaxies" Institute of Physics, Eötvös University. 1999.

[15] M. A. Turk, "Face Recognition Using Eigenfaces" Vision and Modeling Group, The Media Laboratory, Massachusetts Institute of Technology. 1991.

[16] M. A. Turk, Alex Pentland, "Eigenfaces for Recognition" Vision and Modeling Group, The Media Laboratory, Massachusetts Institute of Technology. 1991.

[17] M. A. Vicente, C. Fernández, A. Gil, L. Pavá, "Equivalencia entre ICA y PCA como métodos de extracción de características en reconocimiento visual basado en apariencia", Dpto. de Ingeniería de Sistemas Industriales, Universidad Miguel Hernández. Alicante. España. 2007

[18] Aapo Hyvärinen, Juha Karhunen, Erkki,Oja. "Independent component analysis" (1st ed.). New York: J. Wiley, 2001.

[19] M. Delbracio, M. Mateu, "Identificación utilizando PCA, ICA y LDA". 2006.

[20] Rish, Irina. An empirical study of the naive Bayes classifier (PDF). IJCAI Workshop on Empirical Methods in AI, 2001.

[21] Ben-Gal, Irad. "Bayesian Networks". En Ruggeri, Fabrizio; Kennett, Ron S.; Faltin, Frederick W. Encyclopedia of Statistics in Quality and Reliability, 2007.

[22] M.G. Pose, "Introducción a las Redes de Neuronas Artificiales", Dpto. Tecnologías de la Información y las Comunicaciones, Universidade da Coruña. 2007.

[23] J.L. Alba Castro, "Máquinas de Vectores de Soporte", Dpto. de Teoría de Señal y Comunicaciones, Universidad de Vigo. 2006.

[24] Altman, N. S. "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician. 1992.

[25] Lior Rokach, Oded Maimon. "Data mining with decision trees: theory and applications", World Scientific, 2008.

[26] Witten, I. H., Frank, E., "Data mining: practical machine learning tools with Java implementations", Morgan Kaufmann, San Francisco, 2000.

Apéndice A: Código Fuente

En este apéndice se detallan los códigos *Java* y R utilizados para realizar los distintos algoritmos mostrados en el capítulo 3. Todo el código hecho en *Java* fue realizado con la librería *OpenCV* versión 3.0.

Preprocesamiento de imágenes

Código fuente detallando el pseudocódigo del Capítulo 3.

```
// Leemos la imagen en escala de grises
Mat srcGray = Imgcodecs.imread(<nombre_archivo>, Imgcodecs.CV_LOAD_IMAGE_GRAYSCALE);
/* Binarizamos con un umbral para eliminar ruido de fondo
 * THRESHOLD = 50
 */
Mat binaryImg = new Mat();
Imgproc.threshold(srcGray, binaryImg, THRESHOLD, 255, Imgproc.THRESH_BINARY);
// Definimos un elemento estructurante de 2x2
Mat kernel = new Mat();
kernel.ones(new Size(2, 2), CvType.CV_8U);
/*
 * Realizamos apertura a la imagen binaria para eliminar posibles
 * puntos no filtrados por el umbral de binarizacion
 */
Imgproc.morphologyEx(binaryImg, openImg, Imgproc.MORPH_OPEN, kernel);
// Obtenemos los contornos de los objetos detectados en la imagen
List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
Mat hierarchy = new Mat();
Imgproc.findContours(image, contours, hierarchy, Imgproc.RETR_EXTERNAL,
    Imgproc.CHAIN_APPROX_NONE);
/*
 * Ubicamos la posicion del objeto con la mayor area,
 * asumimos que correspondera al area del objeto de interes
 */
double largestArea = 0;
int largestContourIndex = 0;
for (int k = 0; k < contours.size(); k++) {
   double a = f.getArea(contours.get(k));
   if (a > largestArea) {
      largestArea = a;
      largestContourIndex = k;
   }
}
// Obtenemos el objeto con el area mas grande
Mat biggestContour = new Mat(filtredImg.size(), CvType.CV_8U, new Scalar(0));
```

```
Imgproc.drawContours(biggestContour, contours, largestContourIndex, new Scalar(255, 255,
    255), Core.FILLED);
/*
 * Intersectamos la matriz original con la imagen de mayor area para
 * extraer de la imagen original el detalle que nos interesa
 */
Mat intersectionMat = new Mat(srcGray.size(), CvType.CV_8U, new Scalar(0, 0, 0));
Core.bitwise_and(srcGray, biggestContour, intersectionMat);
// Convertimos el contorno del objeto de interes a formato de puntos
MatOfPoint2f newContourMat = new
    MatOfPoint2f(contours.get(largestContourIndex).toArray());
/*
 * Envolvemos el objeto de interes con una elipse,
 * el resultado nos da la posicion del centro de la elipse
 * y el angulo de inclinacion de la misma
 */
RotatedRect boundingEllipse = Imgproc.fitEllipse(newContourMat);
// Calculamos el punto central del cuadro de imagen
Point center = new Point(intersectionMat.size().width / 2, intersectionMat.size().height
    / 2);
/*
 * Construimos una matriz de traslacion para mover el
 * centro de la imagen de interes al centro del cuadro
 * de imagen
 */
Mat m = new Mat(2, 3, CvType.CV_32F);
m.put(0, 0, 1);
m.put(0, 1, 0);
m.put(1, 0, 0);
m.put(1, 1, 1);
m.put(0, 2, center.x - boundingEllipse.center.x);
m.put(1, 2, center.y - boundingEllipse.center.y);
// Trasladamos la imagen al centro del cuadro
Mat translateImg = new Mat(srcGray.size(), CvType.CV_8U, new Scalar(0, 0, 0));
Imgproc.warpAffine(intersectionMat, translateImg, m, intersectionMat.size());
/*
 * Creamos una matriz de rotacion utilizando el centro del cuadro
 * de imagen y el angulo de inclinacion de la elipse, este angulo
 * representa los grados respecto a la vertical, medidos en
 * sentido horario, por lo que tenemos que restar 90 grados para
 * tener el angulo respecto a la horizontal
 */
Mat rotationMatrix = new Mat(translateImg.size(), CvType.CV_8U);
rotationMatrix = Imgproc.getRotationMatrix2D(center, boundingEllipse.angle - 90, 1);
Mat rotatedImg = new Mat(translateImg.size(), CvType.CV_8U, new Scalar(0, 0, 0));
/* Rotamos la imagen para alinear el eje mayor de la elipse con la
 * horizontal
 */
Imgproc.warpAffine(translateImg, rotatedImg, rotationMatrix, rotatedImg.size());
```

Extracción de MFs:

Código fuente detallando el pseudocódigo del Capítulo 3.

```
/*
 * Calculamos la elongacion
 */
double a = boundingEllipse.size.width / 2;
double b = boundingEllipse.size.height / 2;
double elongation = (a - b) / (b + a);
/*
 * Calculamos el factor de forma
 */
List<MatOfPoint> contourPoints = new ArrayList<MatOfPoint>();
Imgproc.findContours(rotatedImg, contourPoints, hierarchy, Imgproc.RETR_EXTERNAL,
    Imgproc.CHAIN_APPROX_NONE);
double area = Imgproc.contourArea(new MatOfPoint2f(contourPoints.get(0).toArray()),
    false);
double perimeter = Imgproc.arcLength(new MatOfPoint2f(contourPoints.get(0).toArray()),
    false);
double formFactor = area / perimeter;
/*
 * Calculamos la convexidad
 */
Rect rec = Imgproc.boundingRect(contourPoints.get(0));
double convexity = perimeter / (2 * rec.width + 2 * rec.height);
/*
 * Calculamos el factor de forma rectangular
 */
double bff = (rec.width * rec.height) / area;
/*
 * Calculamos el indice de asimetria
 */
// Rotamos la imagen 180 grados
Mat flippedImg = new Mat(rotatedImg.size(), CvType.CV_8U, new Scalar(0, 0, 0));
Core.flip(rotatedImg, flippedImg, -1);
// Recortamos las imagenes para tener una asimetria no muy baja
Rect recRotated = Imgproc.boundingRect(contourPoints.get(0));
List<MatOfPoint> contourPointsFlipped = new ArrayList<MatOfPoint>();
Imgproc.findContours(flippedImg, contourPointsFlipped, hierarchy, Imgproc.RETR_EXTERNAL,
    Imgproc.CHAIN_APPROX_NONE);
Rect recFlipped = Imgproc.boundingRect(contourPointsFlipped.get(0));
Mat croppedRotated = rotatedImg.submat(recRotated);
Mat croppedFlipped = flippedImg.submat(recFlipped);
// Hallamos la diferencia de intensidad de pixeles en ambas matrices
double intensity;
double indexAsymmetry = 0;
for (int x = 0; x < croppedRotated.size().height; x++) {
   for (int y = 0; y < croppedRotated.size().width; y++) {
      intensity = Math.abs(croppedRotated.get(x, y)[0] - croppedFlipped.get(x, y)[0]);
       indexAsymmetry += intensity;
   }
}
// Normalizamos el indice de asimetria
indexAsymmetry /= (256 * croppedFlipped.size().width * croppedFlipped.size().height);
```

```
/*
 * Calculamos los picos horizontales y verticales
 */
// Dibujamos la linea horizontal que cruza el centro de la imagen
Mat hLine = new Mat(rotatedImg.size(), CvType.CV_8U, new Scalar(0, 0, 0));
Imgproc.line(hLine, new Point(0, rotatedImg.size().height / 2), new
    Point(rotatedImg.size().width, rotatedImg.size().height / 2),
new Scalar(255, 255, 255), 1, Imgproc.LINE_8, 0);
// Intersectamos la imagen con la linea horizontal
Mat hIntersectionLine = new Mat(rotatedImg.size(), CvType.CV_8U, new Scalar(0, 0, 0));
Core.bitwise_and(rotatedImg, hLine, hIntersectionLine);
// Dibujamos la linea vertical que cruza el centro de la imagen
Mat vLine = new Mat(rotatedImg.size(), CvType.CV_8U, new Scalar(0, 0, 0));
Imgproc.line(vLine, new Point(rotatedImg.size().width / 2, 0), new
    Point(rotatedImg.size().width / 2, rotatedImg.size().height), new Scalar(255, 255,
    255), 1, Imgproc.LINE_8, 0);
// Intersectamos la imagen con la linea vertical
Mat vIntersectionLine = new Mat(rotatedImg.size(), CvType.CV_8U, new Scalar(0, 0, 0));
Core.bitwise_and(rotatedImg, vLine, vIntersectionLine);
// Contamos los picos de intensidad en el eje horizontal
int diff = 0;
int change = 1;
/* Para regular el nivel de sensibilidad del detector de picos
 * consideramos 1 como maxima sensibilidad y 255 como minima
 * sensibilidad. La sensibilidad afecta mucho cuando entre pixeles
 * consecutivos, hay pequenhas diferencias de intensidad que son
 * imperceptibles a la vista humana pero afecta al contador
 */
int sensibility = 2;
boolean peak = false;
int contHorizontalPeaks = 0;
for (int p = 1; p < hLine.size().width; p++) {
   diff = (int) (hIntersectionLine.get((int) (hLine.size().height / 2), p)[0] -
       hIntersectionLine.get((int) (hLine.size().height / 2), p - 1)[0]);
   if (diff < 0 && change > 0) {
      if (Math.abs(diff) > sensibility) {
         peak = true;
         change = -1;
      }
   } else if (diff > 0 && change < 0) {
      if (Math.abs(diff) > sensibility) {
         peak = false;
         change = 1;
      }
   } else if (diff == 0) {
      peak = false;
   }
   if (peak) contHorizontalPeaks++;
   peak = false;
}
// Contamos los picos de intensidad en el eje vertical
change = 1;
peak = false;
int contVerticalPeaks = 0;
for (int p = 1; p < vLine.size().height; p++) {
```

```
   diff = (int) (vIntersectionLine.get(p, (int) (vLine.size().width / 2))[0] -
       vIntersectionLine.get(p - 1, (int) (vLine.size().width / 2))[0]);
   if (diff < 0 && change > 0) {
      if (Math.abs(diff) > sensibility) {
         peak = true;
         change = -1;
      }
   } else if (diff > 0 && change < 0) {
      if (Math.abs(diff) > sensibility) {
         peak = false;
         change = 1;
      }
   } else if (diff == 0) {
      peak = false;
   }
   if (peak) contVerticalPeaks++;
   peak = false;
}
/*
 * Calculamos el ratio de circularidad
 */
double r1 = (4 * Math.PI * area) / Math.pow(perimeter, 2);
/*
 * Calculamos el ratio de forma
 */
double r2 = (4 * area) / (Math.PI * Math.pow(a, 2));
/*
 * Calculamos el ratio de compacidad
 */
/*
 * Envolvemos el objeto de interes con un circulo,
 * el resultado nos da la posicion del centro de la elipse
 * y el angulo de inclinacion de la misma
 */
float[] radius = new float[1];
Imgproc.minEnclosingCircle(newContourMat, center, radius);
double r3 = area / (Math.PI * Math.pow((int) radius[0], 2));
/*
 * Calculamos el ratio de radio
 */
double dist, maxdist = -1;
for(int ic = 0; ic < rotatedImg.cols(); ic++) {
   for(int jc = 0; jc < rotatedImg.rows(); jc++) {
      dist = Imgproc.pointPolygonTest(newContourMat, new Point(ic, jc), true);
      if(dist > maxdist) {
         maxdist = dist;
         center = new Point(ic, jc);
      }
   }
}
double r4 = maxdist / radius[0];
/*
 * Calculamos la firma de dispersion luminica
 */
// Obtenemos el punto central de la imagen
```

```
Point centerLDS = new Point(rotatedImg.size().width / 2, rotatedImg.size().height / 2);
double total = 0;
int difX;
int difY;
double max;
for (int xIndex = 0; xIndex < rotatedImg.width(); xIndex++) {
   for (int yIndex = 0; yIndex < rotatedImg.height(); yIndex++) {
      if (((int) centerLDS.x == xIndex) && ((int) centerLDS.y == yIndex)) {
         total += rotatedImg.get(yIndex, xIndex)[0];
      } else {
         difX = Math.abs((int) centerLDS.x - xIndex);
         difY = Math.abs((int) centerLDS.y - yIndex);
         max = (difX >= difY) ? (difX > 0) ? difX : difY : (difY > 0) ? difY : difX;
         total += ((1d / (8d * max)) * rotatedImg.get(yIndex, xIndex)[0]);
      }
   }
}
```

Extracción de PCs: considerando todo el *dataset*

Código fuente detallando el pseudocódigo del Capítulo 3.

```
Mat mean = new Mat();
Mat eigenvectors = new Mat();
/*
 * Esta matriz contendra tantas filas como imagenes hayan
 * y tantas columnas como MxN pixeles tenga cada imagen
 */
Mat images = new Mat(numFiles, height * width, CvType.CV_8U, new Scalar(0, 0, 0));
for (int i = 0; i < files.length; i++) {
   Mat src = Imgcodecs.imread(files[i].toString(), Imgcodecs.CV_LOAD_IMAGE_GRAYSCALE);
    // Convertimos la matriz de imagen a un vector
   (src.reshape(0,1)).copyTo(images.row(i++));
}
// Calculamos el PCA
Core.PCACompute(images, mean, eigenvectors);
// Cargamos las componentes principales en una matriz de 82x82
double[][] pcs = new double[files.length][files.length];
double pc;
for (int i = 0; i < files.length; i++) {
   for (int j = 0; j < eigenvectors.rows(); j++) {
      pc = 0;
      for (int k = 0; k < eigenvectors.cols(); k++) {
         pc += (images.get(i, k)[0] * eigenvectors.get(j, k)[0]);
      }
      pcs[i][j] = pc;
   }
}
```

Extracción de PCs: considerando la traspuesta del *dataset*

Código fuente detallando el pseudocódigo del Capítulo 3.

```
Mat covar = new Mat();
Mat covarMean = new Mat();
Core.transpose(images, images);
/*
 * Calculamos la matriz de covarianza de la traspuesta de
 * la matriz de imagenes, ya que se hace intratable calcular
 * de la matriz original. De esta forma tenemos una matriz
 * de covarianza de IxI donde I corresponde a la cantidad de
 * imagenes, en lugar de (MxN)x(MxN) donde M y N son el ancho
 * y alto de cada imagen
 */
Core.calcCovarMatrix(images, covar, covarMean, Core.COVAR_NORMAL | Core.COVAR_ROWS);

Mat eigenvalues = new Mat();
Mat eigenvectors = new Mat();
/* Calculamos los autovectores y autovalores de
 * la matriz de covarianza
 */
Core.eigen(covar, eigenvalues, eigenvectors);

double[][] pcs = new double[files.length][files.length];
double pc;
for (int i = 0; i < files.length; i++) {
   for (int j = 0; j < eigenvectors.rows(); j++) {
      pc = 0;
      for (int k = 0; k < eigenvectors.cols(); k++) {
         pc += (covar.get(i, k)[0] * eigenvectors.get(j, k)[0]);
      }
      pcs[i][j] = pc;
   }
}
```

Extracción de ICs:

Código fuente detallando el pseudocódigo del Capítulo 3.

```
covar = read.csv("<matriz_covarianza.csv>", header = FALSE, sep = ",", dec = ".", fill =
    TRUE)
ica <- fastICA(covar, 82, alg.typ = "parallel", fun = "logcosh", alpha = 1, method =
    "C", row.norm = TRUE, maxit = 10000, tol = 0.0001, verbose = TRUE)
write.table(ica, file="<matriz_ica.csv>", sep = ",", eol = "\n", dec = ".")
```

Apéndice B: Publicación realizada

A continuación se anexa la publicación realizada en el marco del presente TFG.

a) **Salinas J. Z.**, García-Torres M., Schaerer C. E., Legal H., and Divina F. "Automatic Morphological Classification of Galaxy Images", in *The Conference of Computational Interdisciplinary Science (CCiS 2016)*, São José dos Campos - SP, Brazil, Nov. 2016.

Automatic Morphological Classification of Galaxy Images

[1]Salinas J. Z., [2]García-Torres M., [1]Schaerer C. E., [1]Legal H., [2]Divina F.

1: National University of Asunción, San Lorenzo, Paraguay
2: Pablo de Olavide University, Seville, Spain

Introduction

Galaxy classification is an important task in astronomy in the large scale study of the universe. Although this task traditionally is done manually, astronomy has experienced an explosion of data that require the use of new techniques to deal with this increase of the data volume.

Objectives

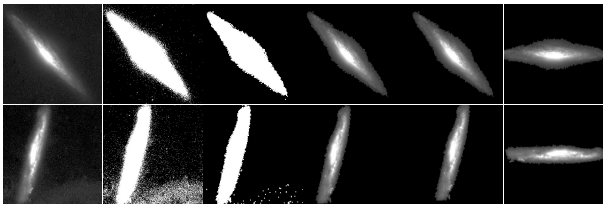
We analyzed the performance of several machine learning approaches (Naïve Bayes, Support Vector Machine and C4.5) on automated classification of galaxy images. We considered three (E, S, Irr), five (E, S0, Sa+Sb, Sc+Sd, Irr) and seven (E, S0, Sa, Sb, Sc, Sd, Irr) galaxy types.

Classification System

The classification system architecture is divided into three main phases:

1 Image Preprocessing

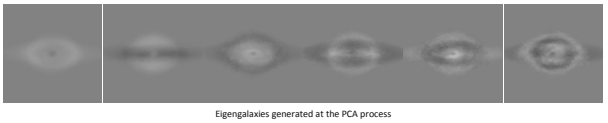
The images are standardized to remove noise and the effect of orientation and translation.



Original Image | Image Binarized | Noise removed | ROI extracted | Image translated | Image rotated

2 Feature Extraction

Features are extracted by morphological appearance (MF), Principal Component Analysis (PCA) and Independent Component Analysis (ICA). The MF are based on the perceived visual characteristics of the galaxy like elongation, form-factor, convexity, bounding-to-fill factor, asymmetry index, horizontal and vertical peaks of histogram. PCA and ICA are extracted from covariance matrix of the transpose of dataset $C = A^T A$ where a row of A represent an image converted into a 1 dimensional vector.



Eigengalaxies generated at the PCA process

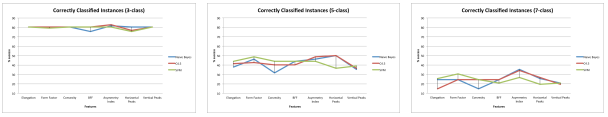
3 Classification

The classification process is performed with the WEKA library using Naïve Bayes and C4.5 algorithms and the downloaded package LibSVM for Support Vector Machine algorithm. We used 10-folds cross-validation for doing all the experiments.

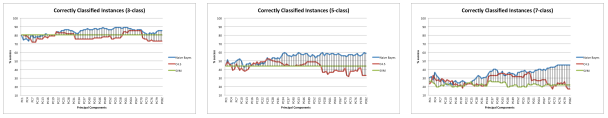
Experimental Results

Experimental results show that increasing the number of galaxy types degrades the model performance.

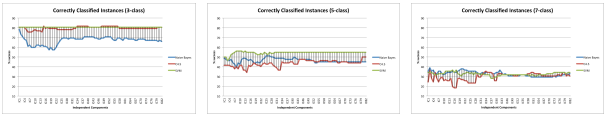
Morphological Features



Principal Component Analysis



Independent Component Analysis



Conclusions

Given a feature subset, different classification algorithms can yield different performance measures.

Despite the promising results, more research is necessary to improve the classification when increasing the number of galaxy types. A possible reason of the poor results achieved for classes 5 and 7 could be the small number of instances for some classes; therefore increasing the dataset could improve the results. In this research, best results were achieved with Naïve Bayes using PCA features.

References

Z. Frei, "Automatic morphological classification of galaxies", Institute of Physics, Eötvös University.
J. de la Calleja, O. Fuentes, "Automated Classification of Galaxy Images", Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro 1. 2004.
S. Kasivajhula, N. Raghavan, H. Shah, "Morphological Galaxy Classification Using Machine Learning".
L. Shamir, "Automatic morphological classification of galaxy images", Laboratory of Genetics, NIA-NIH, 2005.
M.A. Turk, A.P. Pentland, "Face Recognition Using Eigenfaces", in Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, 586-591, 1991
M. M. Pawlik, V. Wild, "Shape Asymmetry: a morphological indicator for automatic detection of galaxies in the post-coalescence merger states", School of Physics and Astronomy of St. Andrews, U.K., 2016.

Acknowledgments

- CES and HL acknowledge PRONII-CONACYT.
- JZS, CES and HL acknowledge financial support of PROCIENCIA-CONACYT under research project #14-INV-202.
- Facultad Politécnica.
- Laboratorio de Computación Científica y Aplicada LCCA.

Contact: jzsalinas@gmail.com