



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS  
SUPERIORES ACATLÁN

Algoritmo de Fusión de Imágenes  
Hiperespectrales Satelitales Mediante  
Regresiones Basadas en Factores de Modelos  
Bilineales

TESIS

QUE PARA OBTENER EL TÍTULO DE:  
**Licenciado en Matemáticas Aplicadas y  
Computación**

PRESENTA:

**María Fernanda Beltrán Llorente**

ASESOR:

Mtro. Mariano José Juan Rivera Meraz

Junio, 2017



Santa Cruz, Acatlán, Edo. de México



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



*A mi madre, Carmen Llorente,  
tu amor incondicional siempre me ha hecho fuerte, gracias por darme lo  
mejor de ti en todo momento y por aceptarme y amarme tal como soy.*

*A mi padre, Ricardo Beltrán,  
sin tus exigencias y tu mano dura nunca habría sabido encontrar el camino  
adecuado, gracias por todas tus enseñanzas y por estar para mi cuando te  
necesito.*

*A mi hermana, Catalina,  
mi amiga, mi compañera de vida, mi ejemplo a seguir, sin ti jamás hubiera  
llegado tan lejos, gracias por cuidar de mi desde que tengo memoria.*

*A Dios,  
porque nunca me abandonas.*



# Agradecimientos

A mis abuelitos, **Andy y Gelita**, por consentirme y quererme, por ser grandes ejemplos a seguir y porque me han apoyado en todo momento.

A mis tíos **Ángeles, Laura y Armando**, por todo su cariño, por ser mis amigos y recorrer conmigo este camino, por todos sus consejos y por que siempre se preocupan porque esté bien.

A mis primos, **Paco, Mariana, Blanca y Ofelia**, por todas las aventuras que hemos vivido juntos, hacen que mi vida sea muy divertida, simplemente me encanta tenerlos.

A mi mejor amiga, **Itzia Calalpa**, porque nunca has perdido la fe en mi, me has apoyado en cada momento, has estado en mis mejores y peores momentos, es un placer tenerte como amiga y compartir mis triunfos contigo.

Al **Dr. Mariano Rivera**, por darme la oportunidad de trabajar con usted, por el tiempo y apoyo que me brindo.

A todos mis amigos, por los buenos momentos que hemos compartido juntos, por su apoyo y porque siempre están cuando los necesito.

Al **Dr. Eloy Loza** , al **Dr. Andrés Orduz** , al **Dr. Jorge Vasconcelos** y al **Ing. Rubén Romero** por tomarse le tiempo de revisar este trabajo.



# Índice general

<b>Introducción</b>	<b>v</b>
<b>1. Antecedentes</b>	<b>1</b>
1.1. Imágenes VIIRS	1
1.1.1. Características de las imágenes del sensor VIIRS	4
1.1.2. ¿Quién hace uso de las imágenes VIIRS en México?	7
1.2. Algoritmo CIMAT-CONABIO	9
<b>2. Regresión sobre componentes</b>	<b>15</b>
2.1. Mínimos Cuadrados	15
2.2. Análisis de Componentes Principales	19
2.3. Regresión con Componentes Principales	25
2.4. Regresión de Mínimos Cuadrados Parciales	28
2.5. Algoritmo de fusión de imágenes propuesto	31
2.5.1. Creación de máscaras	32
2.5.2. Reducción de escala	32
2.5.3. Predicción de las bandas de alta resolución	34
2.5.4. Normalización	36
2.5.5. Variantes	37
2.6. Implementación en Python	38
<b>3. Resultados</b>	<b>41</b>
3.1. Descripción de los datos	41
3.2. Imagen 1	42
3.3. Imagen 2	47

3.4. Imagen 3 . . . . .	50
3.5. Imagen 4 . . . . .	52
3.6. Imagen 5 . . . . .	55
<b>Conclusiones y Trabajo futuro</b>	<b>59</b>
<b>Apéndices</b>	<b>61</b>
.1. Mínimos Cuadrados . . . . .	63
.2. Análisis de Componentes Principales . . . . .	64
.3. Base de datos “ <i>Places Rated</i> ” . . . . .	65
.4. NIPALS . . . . .	67
<b>Bibliografía</b>	<b>69</b>

# Introducción

El análisis de la cobertura terrestre ha tenido un gran auge en las últimas décadas, debido a que las actividades humanas han realizado cambios drásticos a la cobertura de nuestro planeta. Por lo anterior se han puesto en marcha diversos proyectos que tienen como principal objetivo llevar a cabo el monitoreo de las coberturas globales, para los cuales se hace uso de diferentes tipos de sensores a bordo de satélites que han sido lanzados desde la década de los ochentas. Entre ellos existen el sensor llamado Radiómetro de Imágenes Visibles Mediante Infrarrojos Suite (VIIRS, por sus siglas en inglés *Visible Infrared Imaging Radiometer Suite*), cuyas imágenes son procesadas y distribuidas en México gracias a que la Comisión Nacional para el Conocimiento y Uso de la Biodiversidad (CONABIO) cuenta con una antena y un equipo de trabajo. Las imágenes de este sensor son de gran interés ya que es relativamente nuevo en comparación con sus antecesores y por lo tanto las imágenes que provee son mejores y más fáciles de manejar.

En este trabajo presento la propuesta de un Algoritmo de Fusión de Imágenes Satelitales VIIRS, como alternativa a los algoritmos de Fusión que existen en la actualidad. El objetivo principal de este algoritmo es la fusión de bandas de baja resolución y de alta resolución para obtener versiones de alta resolución de las bandas de baja resolución predecibles. Esta propuesta busca obtener resultados similares o mejores a los obtenidos con algoritmos predecesores, pero usando diferentes métodos matemáticos y representación de los datos.

La presente tesis se divide en tres capítulos. En el primer capítulo se explica de manera breve qué son las imágenes VIIRS, cuál es su importancia y su uso en México; además de exponer un Algoritmo de Fusión de Imágenes

Satelitales previamente desarrollado por investigadores del Centro de Investigación en Matemáticas (CIMAT). En el segundo capítulo se plantean los objetivos del algoritmo propuesto en este trabajo. Se presenta el desarrollo del mismo y algunos detalles de su implementación en lenguaje de programación Python. En el tercer y último capítulo se muestran experimentos realizados con datos reales obtenidos a través de la CONABIO. El objetivo de los experimentos es hacer un análisis del desempeño del nuevo método. En las conclusiones se analizan las contribuciones del trabajo desarrollado y finalmente se expone el trabajo futuro que podría realizarse como continuación de la presente tesis.

# Capítulo 1

## Antecedentes

### 1.1. Imágenes VIIRS

El material expuesto en este capítulo es un resumen de las fuentes [1], [14], [15] y [22].

Las actividades humanas han cambiado drástica y rápidamente la cobertura de nuestro planeta; hace ya varias décadas se puso en manifiesto que los cambios de cobertura y el uso de suelo influyen directamente en los ciclos hidrológicos, en la pérdida de biodiversidad, en la erosión de los suelos y en el aumento de gases; los cuales a su vez incrementan el efecto invernadero. Los cambios climatológicos provocan el aumento y la intensidad de los desastres naturales en todo el planeta como los son incendios, inundaciones, huracanes y sequías. Por esta razón, diversos grupos especializados, gobiernos e instancias internacionales se han unido para abordar esta problemática.

Desde finales de la década de los ochentas se llevaron a cabo proyectos creados para el monitoreo de coberturas globales, como el *International Geosphere Biosphere Programme* (IGBP), que permitió el mapeo de la cobertura terrestre usando los datos del sensor *Advanced Very High Resolution Radiometer* (AVHRR) [7]. El uso de este sensor permitió crear bases de datos globales con más de dos décadas de datos gracias a la adquisición diaria de imágenes. Sin embargo los datos AVHRR no son los más idóneos para

estudios de la cobertura terrestre, porque tienen baja resolución espacial, lo cual implica notables problemas geométricos y radiométricos en las imágenes resultantes.

A pesar de no ser los más idóneos, los datos AVHRR tuvieron resultados suficientemente alentadores para motivar la creación de sensores diseñados específicamente para el monitoreo de las coberturas terrestres. Dentro de estos proyectos de percepción remota el más ambicioso es el *Earth Observing System* (EOS) de la NASA (Administración Nacional de Aeronáutica y del Espacio), cuyo principal objetivo es la observación continua de los cambios globales, lo cual incluye el estudio integrado de la atmósfera, de los océanos y de la superficie terrestre [15].

Dentro del proyecto EOS sobresale el sensor llamado Espectroradiómetro para Imágenes de Resolución Moderado, MODIS [15], por sus siglas en inglés *Moderate Resolution Imaging Spectroradiometer*, el cual fue creado por un equipo científico interdisciplinario. Este sensor está a bordo de las plataformas Terra y Aqua. Sus datos cuentan con alta sensibilidad radiométrica, buena calidad geométrica, alta resolución temporal y son distribuidos de manera gratuita a través de diferentes portales de Internet. Los datos crudos pueden obtenerse de manera directa contando con la tecnología necesaria; en México se cuenta con una antena y un equipo de trabajo de procesamiento y distribución dentro de la CONABIO (Comisión Nacional para el Conocimiento y Uso de la Biodiversidad).

El sensor MODIS, por sus características espaciales y espectrales es uno de los más importantes para el monitoreo de los procesos de cambio en la tierra. Cuenta con una alta sensibilidad radiométrica (12 bits y 16 bits) en 36 bandas espectrales, en un rango de longitud de onda que va de los  $0.4\mu m$  a los  $14.4\mu m$ . Las primeras 19 bandas están posicionadas en la región del espectro electromagnético situado entre  $0.000405\mu m$  y  $0.002144\mu m$ . Las bandas de la 1 a la 7 son útiles para las aplicaciones terrestres; las bandas 8 a la 16 para las observaciones oceánicas y las bandas 17 a la 19 para las mediciones atmosféricas. Las bandas 20 a la 36 cubren la porción del infrarrojo térmico del espectro (de  $0.00366 - 0.014385\mu m$ ). Dos bandas son tomadas a una resolución nominal de 250m *at nadir*, cinco bandas a 500m al nadir<sup>1</sup> y las 29 restantes a 1km. MODIS posee una alta calidad geométrica que permite el monitoreo preciso de las alteraciones de la superficie terrestre.

---

<sup>1</sup>A la intersección entre la vertical del observador y la esfera celeste.

El equipo científico de MODIS Land (MODLAND) desarrolla métodos y algoritmos para generar productos sobre cubiertas terrestres.

Las experiencias derivadas de los trabajos con el sensor MODIS han permitido explorar una gran cantidad de aplicaciones potenciales que, al alcanzar el fin de sus operaciones, buscan ser aplicadas a la siguiente etapa de observación de la tierra a través del sensor VIIRS.

El sensor VIIRS, por sus siglas en inglés *Visible Infrared Imager Radiometer Suite*, programado para volar en los satélites del proyecto *National Polar-orbiting Operational Environmental Satellite System* (NPOESS), fue lanzado por primera vez el 28 de Octubre del 2011 a bordo del Satélite Soumi-NPP. Su principal objetivo es reemplazar y combinar las misiones de los sensores *Advanced Very High Resolution Radiometer* (AVHRR) y *Operational Linescan System* (OLS). Actualmente, imágenes de ambos sensores (AVHRR y OLS) tienen una serie de características que limitan su uso en las operaciones de pronóstico, comparadas con imágenes provenientes de satélites geostacionarios.

El sensor VIIRS posee una cobertura espectral que va de los  $0.412\mu m$  a los  $12\mu m$ . Proporciona mediciones parecidas a las del MODIS. Cuenta con 22 bandas. Las imágenes tienen una resolución de  $375m$  al nadir en 5 bandas y  $750m$  al nadir en las bandas restantes. Cuenta con una cobertura global completa: productos de nubes, propiedades de aerosoles, tierra y océanos, temperatura de la superficie, del océano, tierra y hielo; además de detección de incendios. En este sensor se reduce el alargamiento del pixel con los límites de escaneo, la cobertura espectral es ligeramente menor, las bandas infrarrojas son más largas para  $CO_2$ , el sistema de telescopio rotatorio permite un mejor control de la luz, la órbita es más alta, lo que permite la cobertura global completa en un día (de  $705km$  a  $830km$ ). La cantidad radiométrica y la calidad espectral es similar a MODIS con datos de 12 bit, las calibraciones a bordo del satélite también son parecidas a las de MODIS, el impacto en el índice de vegetación es mínimo. La reducción de 36 bandas en MODIS a 22 en VIIRS se compensa en parte debido a las 7 bandas de amplificación dual en VIIRS que pueden ser usadas tanto para obtener datos de cobertura terrestre como de superficies oceánicas, ya que cuenta con un sistema que permite ajustar las mediciones del sensor de acuerdo a la superficie a observar. Los algoritmos de VIIRS surgen del aprovechamiento y mejora de la herencia de MODIS, esperando dar continuidad a estos datos

con coherencia [22].

En la figura 1.1 se muestra una imagen VIIRS, la cual corresponde al día 20 de enero del 2015 y fue recibida, procesada y proporcionada por la CO-NABIO para realización de este trabajo.

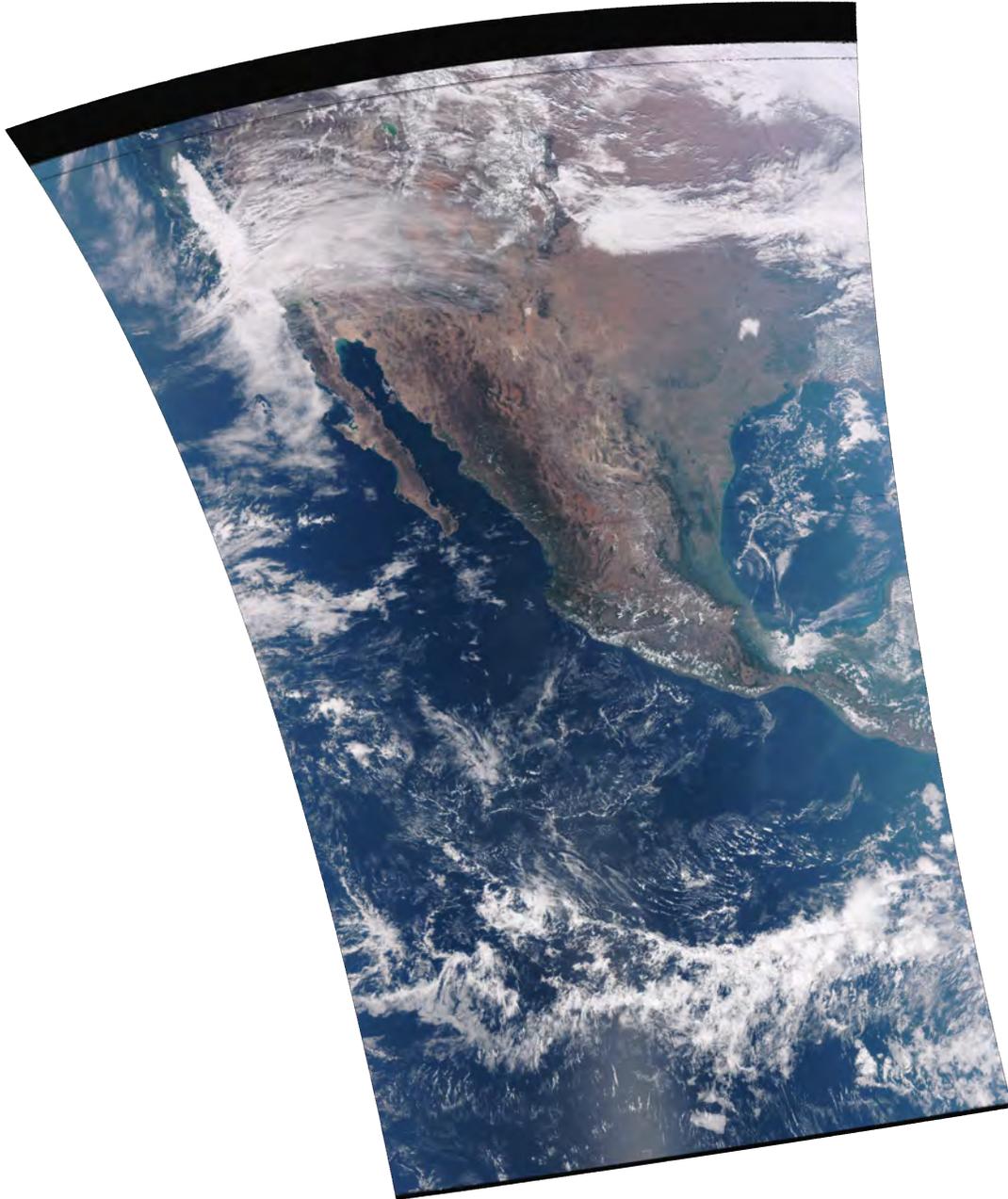
### 1.1.1. Características de las imágenes del sensor VIIRS

El sensor VIIRS cuenta con dos resoluciones especiales: **Imaging(I)** a 375 metros *at nadir* y **Moderate(M)** a 750 metros al nadir. Cuenta con 22 bandas, 5 de las cuales son **Imaging** y las otras 16 bandas son **Moderate**. En la tabla 1.1 se muestra la descripción de las 8 bandas del sensor VIIRS de las cuales se hará uso a lo largo del desarrollo de este trabajo.

Banda	$\lambda$ (nm)	Tamaño del pixel al nadir (m)	Principal(es) Aplicación(es)
I1	600 - 800	371 x 387	Imágenes, Vegetación.
I2	846 - 885	371 x 387	Vegetación.
I3	1580 - 1640	371 x 387	Capa de nieve.
M3	478 - 498	742 x 259	Color del océano, Aerosoles.
M4	545 - 565	742 x 259	Color del océano, Aerosoles.
M8	1230 - 1250	742 x 776	Tamaño de partícula de las nubes.
M10	1580 - 1640	742 x 776	Mapa binario de nieve.
M11	2225 - 2275	742 x 776	Nubes.

**Tabla 1.1:** Descripción de las bandas del sensor VIIRS.

Las bandas del sensor VIIRS que se muestran en la tabla 1.1 contienen la información de la Reflectancia de la Superficie (SR *Reflectance Product*) y los datos están representados en enteros de 12 bits. La información adicional incluye el tipo de proyección, sistema de coordenadas, elipsoide, datum y todo lo necesario para que la imagen pueda ser automáticamente posicionada en un sistema de referencia espacial. Los datos básicos del satélite son transmitidos a la tierra como archivos RDR (Raw Data Records), estos son procesados y convertidos a archivos SDR (*Sensor Data Records*). Un número limitado de archivos SDR son procesados y convertidos a un archivo EDR



**Figura 1.1:** Ejemplo de una imagen VIIRS.

(*Environmental Data Record*). El producto SR pertenece a este grupo de datos EDR y cada archivo VIIRS se encuentra en formato HDF-5.

Los valores de la reflectancia son escalados a una representación de 16 bits. Los datos ocupan 15 bits de un entero de 16 bits, el bit más significativo indica valores inválidos. Los valores de reflectancia sobre el rango dinámico  $[D_{min}, D_{max}]$  es escalado al rango  $[0, 32767]$  y representando en una variable entera.

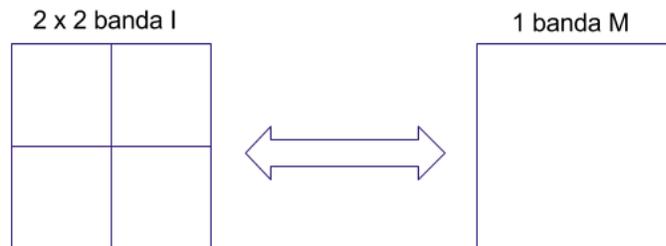
Nombre y significado	16-bit arreglos de enteros sin signos	32 bit arreglos de números con puntos decimales
N/A - No Aplicable.	65535	-999.9
MISS - Valor requerido faltante al momento de procesar.	65534	-999.8
OBPT - Borde de pixeles integrado (pixel eliminado en el procesamiento SDR).	65533	-99.7
OGPT - Recorte de pixeles en el suelo (Superposición / empate Pixel eliminado durante el proceso EDR).	65532	-99.6
ERR - Ocurrió un error durante el procesamiento / no existe convergencia.	65531	-99.5
ELINT - La intersección del elipsoide falló / la línea del instrumento de visión no intercepta la superficie terrestre.	65530	-999.3
VDNE - El valor no existe / el algoritmo de procesamiento no se ejecutó.	65529	-999.3
SOUB - Escalonada fuera de límites / solución no dentro del rango permitido.	65528	-999.2

**Tabla 1.2:** Valores inválidos en imágenes VIIRS.

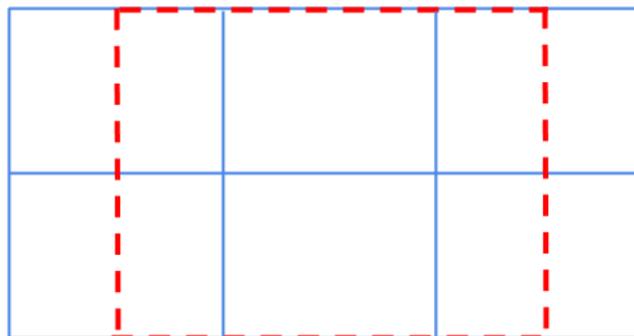
En la tabla 1.2 se muestran los valores utilizados en la generación de máscaras que definen las áreas que no son de interés durante el procesamiento de

las bandas. Para propósito de este trabajo, los valores de la tabla mencionada caen fuera del rango de datos válido y por lo tanto no son utilizados.

En el sensor VIIRS, el registro de las bandas se produce de forma simple debido a que las bandas **I** y **M** están alineadas, lo que significa que los marcos de resolución *moderate* cubren casi la misma cobertura de superficie terrestre que dos marcos de resolución *Imaging*. En conclusión VIIRS fue diseñado para colocar 2x2 píxeles de banda I dentro de un píxel de banda M, tal como se muestra en la figura 1.2. Esto no ocurría en los píxeles del sensor predecesor (MODIS), en el cual, un píxel de baja resolución (banda **M**) se sobrelapa sobre una región de 2x3 píxeles de la banda de alta resolución (banda **I**) como se muestra en la figura 1.3.



**Figura 1.2:** Relación entre bandas del sensor VIIRS.



**Figura 1.3:** Relación entre bandas del sensor MODIS.

### 1.1.2. ¿Quién hace uso de las imágenes VIIRS en México?

Desde 1992, la Comisión Nacional para el Conocimiento y Uso de la Biodiversidad (CONABIO) promueve, coordina, apoya y realiza las actividades

dirigidas al conocimiento de la diversidad biológica del país, así como para su conservación y uso sustentable. La CONABIO recibe en tiempo real imágenes tomadas por el sensor VIIRS a bordo del Satélite Soumi-NPP.

Para varias aplicaciones, es deseable tener bandas de resolución espacial grande, ya que se ha demostrado que la resolución óptima debería ser del orden de unos pocos de cientos de metros para operaciones de monitoreo y control. A raíz de ello, la CONABIO tiene interés en obtener una alta resolución especial entre el grupo de imágenes VIIRS, que permitirá el monitoreo de distintos tipos de cobertura terrestre, teniendo así la posibilidad de hacer evaluaciones más precisas de las propiedades en la superficie y en los biomas [22].

La CONABIO obtiene imágenes de satélites del territorio nacional en alta resolución gracias a un programa de algoritmos diseñado por el Centro de Investigación en Matemáticas (CIMAT). Ambas instancias trabajaron en conjunto en el programa Fusión y Análisis de Imágenes Satelitales de Sensores MODIS y VIIRS para mejorar el procesamiento de las fotografías que muestran a detalle la biodiversidad del país, a fin de detectar el deterioro o mejoramiento de los ecosistemas y mares.

Por medio de la antena que se ubica en las instalaciones de la CONABIO se captan las señales de los satélites Aqua, Terra y Suomi NPP de la NASA, a través de los sensores MODIS y VIIRS. Una vez que se obtienen, son procesadas en el software creado en los laboratorios de CIMAT (el cual pertenece al Sistema de Centros Públicos de Investigación del Consejo Nacional de Ciencia y Tecnología (CONACYT)), a fin de mejorar la resolución de las bandas de baja resolución (**M**).

Entre los proyectos realizados por la CONABIO se encuentra el proyecto “Sistema de alerta temprana de incendios forestales con datos satelitales”, el cual es un sistema que utiliza imágenes de tipo MODIS y VIIRS para generar información en forma operacional para la atención de incendios forestales. Su principal objetivo es detectar incendios forestales a partir de imágenes de satélites diarias. La información generada se publica en la página web de la CONABIO. [1].

## 1.2. Algoritmo CIMAT-CONABIO

A continuación describiremos el método de fusión desarrollado entre el CIMAT y la CONABIO para imágenes MODIS [22], dicho método es una mejora a la propuesta original de Trishchenko [19].

En este proyecto se trabajó con imágenes del sensor MODIS. Aunque como se verá más adelante, su uso se extendió a imágenes VIIRS. Lo apropiado de este procedimiento (diseñado para imágenes MODIS) para imágenes VIIRS se discutirá al final de esta sección.

Se sabe que existe alta correlación entre las bandas de alta y baja resolución del sensor MODIS. Las bandas  $B_3$ ,  $B_4$ ,  $B_6$  y  $B_7$  tienen un alto grado de correlación lineal con la banda  $B_1$ , mientras que la banda  $B_6$  tiene un grado menor de correlación con  $B_2$ . Entonces las bandas  $B_2$ ,  $B_2$  y NDVI (Índice de vegetación normalizado) contribuyen como componentes básicos en la construcción de los otros canales. Los canales  $B_3$ ,  $B_4$ ,  $B_6$  y  $B_7$  muestran correlación negativa con NDVI y el canal  $B_7$  muestra una relación no lineal con NDVI.

Trishchenko propone el procesamiento de las imágenes MODIS haciendo una división en pequeños bloques y generando una regresión no lineal de cada una de las bandas  $B_3$  a  $B_7$  a partir de las bandas  $B_1$ ,  $B_2$  y NVDI.

El modelo de regresión propuesto por Trishchenko (1.1) tiene la forma:

$$G(B_1(x), B_2(x), a) = B_n(x) = a_0 + (a_1 B_1(x) + a_2 B_2(x))(1 + a_3 N(x) + a_4 N(x)^2) \quad (1.1)$$

donde

$$N(x) = \frac{B_2(x) - B_1(x)}{B_2(x) + B_1(x)} \quad (1.2)$$

define el índice vegetación normalizado o NDVI.

El propósito es encontrar los parámetros del modelo de regresión  $B_n(x) = G(B_1(x), B_2(x), a)$  para estimar  $B_1(x)$  a  $B_7(x)$ , donde  $G$  es una función paramétrica,  $x$  un punto en  $\mathbb{R}^2$  y  $a$  un vector de parámetros que pretende minimizar el siguiente problema de mínimos cuadrados no lineales:

$$\underset{a}{\text{mín}} F(a) = \sum_{i=1}^m [y_i - G(B_1(x_i), B_2(x_i), a)]^2. \quad (1.3)$$

Para resolver este problema, Trischenko propone usar el método de solución Levenger-Marquardt y así calcular los coeficientes de la regresión.

La regresión se calcula para todos los datos de una imagen, lo cual tendría una baja precisión porque las características espaciales van cambiando conforme cambian las coberturas, por lo que, el modelo se implementa por ventanas, es decir en cada ventana se calcula una regresión. Esto permite que el modelo se comporte como si se adaptara espacialmente. Si las ventanas son muy pequeñas la regresiones pueden capturar mejor las características locales de las imágenes. Sin embargo, dado a que se trabaja con imágenes de gran tamaño, se deben resolver cientos de miles de sistemas, por lo que una de las formas propuestas para reducir el tiempo computacional es transformar el problema de mínimos cuadrados no lineales a uno de mínimos cuadrados lineales y resolverlo mediante una descomposición de Cholesky. Tomando la ecuación 1.1 y desarrollando los múltiplos de dicha ecuación tenemos que:

$$\begin{aligned} B_n(x) &= a_0 + (a_1 B_1(x) + a_2 B_2(x))(1 + a_3 NVDI(x) + a_4 NVDI(x)^2) \\ B_n(x) &= a_0 + a_1 B_1(x) + a_2 B_2(x) + a_1 a_3 B_1(x) N(x) + a_2 a_3 B_2(x) N(x) \\ &\quad + a_1 a_4 B_1(x) N^2(x) + a_2 a_4 B_2(x) N^2(x) \\ \mathbf{B}_n(\mathbf{x}) &= \theta_0 + \theta_1 \mathbf{B}_1(\mathbf{x}) + \theta_2 \mathbf{B}_2(\mathbf{x}) + \theta_3 \mathbf{B}_1(\mathbf{x}) \mathbf{N}(\mathbf{x}) + \theta_4 \mathbf{B}_2(\mathbf{x}) \mathbf{N}(\mathbf{x}) \\ &\quad + \theta_5 \mathbf{B}_1(\mathbf{x}) \mathbf{N}^2(\mathbf{x}) + \theta_6 \mathbf{B}_2(\mathbf{x}) \mathbf{N}^2(\mathbf{x}) \end{aligned} \quad (1.4)$$

Ahora tenemos un modelo de regresión con parámetros lineales y en lugar de estimar cinco parámetros en el modelo no lineal, estimaremos siete parámetros en el modelo lineal.

Podemos entonces transformar el problema de regresión a un problema de mínimos cuadrados lineales:

$$\min_{\theta} F(\theta) = \sum_{i=1}^m (y_i - B_n(\theta, x_i))^2 = \frac{1}{2} \|A\theta - b\|^2 \quad (1.5)$$

Donde:

$$A = \begin{bmatrix} 1 & B_1(x_0) & B_2(x_0) & B_1(x_0)N(x_0) \\ 1 & B_1(x_1) & B_2(x_1) & B_1(x_1)N(x_1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & B_1(x_{n-1}) & B_2(x_{n-1}) & B_1(x_{n-1})N(x_{n-1}) \end{bmatrix}$$

$$\begin{bmatrix} B_2(x_0)N(x_0) & B_1(x_0)N^2(x_0) & B_2(x_0)N^2(x_0) \\ B_2(x_1)N(x_1) & B_1(x_1)N^2(x_1) & B_2(x_1)N^2(x_1) \\ \vdots & \vdots & \vdots \\ B_2(x_{n-1})N(x_{n-1}) & B_1(x_{n-1})N^2(x_{n-1}) & B_2(x_{n-1})N^2(x_{n-1}) \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_6 \end{bmatrix} \quad b = \begin{bmatrix} B_n(x_0) \\ B_n(x_1) \\ \vdots \\ B_n(x_6) \end{bmatrix}$$

La solución de mínimos cuadrados se obtiene resolviendo el sistema  $(AA^T)\theta = A^Tb$ . La matriz  $A^T A$  puede tener columnas linealmente dependientes por pérdida de datos o porque algunas de las componentes sean 0 o por estar mal condicionada, entonces la matriz  $A^T A$  no será de rango completo. Por lo que se agrega un potencial cuadrático que penaliza la norma del vector de parámetros. A esto se le conoce como regresión Ridge (ver referencia [10]). Con esto se garantiza la existencia de una solución única. La función objetivo en forma matricial se puede escribir como:

$$F(\theta) = \frac{1}{2} \|A\theta - b\|^2 + \lambda \|\theta\|^2. \quad (1.6)$$

El mínimo está dado por:

$$\begin{aligned} (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})\theta - \mathbf{A}^T \mathbf{b} &= \mathbf{0} \\ \mathbb{A}\theta - b &= 0 \end{aligned} \quad (1.7)$$

donde:  $\mathbb{A} \in \mathbb{R}^{7 \times 7}$ ,  $\theta \in \mathbb{R}^{7 \times 1}$ ,  $b \in \mathbb{R}^{7 \times 1}$

$$\mathbf{A}^T \mathbf{b} = b = \begin{bmatrix} \sum_x B_3 \\ \sum_x B_1(x)B_1(x) \\ \sum_x B_2(x)B_3(x) \\ \sum_x B_1(x)N(x)B_3(x) \\ \sum_x B_2(x)N(x)B_3(x) \\ \sum_x B_1(x)N^2(x)B_3(x) \\ \sum_x B_2(x)N^2(x)B_3(x) \end{bmatrix}$$

$$\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I} = \mathbb{A} =$$

$$\begin{bmatrix} N + \lambda & \sum_i B_1(x_i) & \sum_i B_2(x_i) & \sum_i B_1(x_i)N(x_i) \\ \sum_i B_2(x_i) & \sum_i B_1^2(x_i) + \lambda & \sum_i B_1(x_i)B_2(x_i) & \sum_i B_1^2(x_i)N(x_i) \\ \sum_i B_2(x_i) & \sum_i B_1(x_i)B_2(x_i) & \sum_i B_2^2(x_i) + \lambda & \sum_i B_1(x_i)B_2(x_i)N(x_i) \\ \sum_i B_1(x_i)N(x_i) & \sum_i B_1^2(x_i)N(x_i) & \sum_i B_1(x_i)B_2(x_i)N(x_i) & \sum_i B_1^2(x_i)N^2(x_i) + \lambda \\ \sum_i B_2(x_i)N(x_i) & \sum_i B_1(x_i)B_2(x_i)N(x_i) & \sum_i B_2^2(x_i)N(x_i) & \sum_i B_1(x_i)B_2(x_i)N^2(x_i) \\ \sum_i B_1(x_i)N^2(x_i) & \sum_i B_1^2(x_i)N^2(x_i) & \sum_i B_1(x_i)B_2(x_i)N^2(x_i) & \sum_i B_1^2(x_i)N^3(x_i) \\ \sum_i B_2(x_i)N^2(x_i) & \sum_i B_1(x_i)B_2(x_i)N^2(x_i) & \sum_i B_2^2(x_i)N^2(x_i) & \sum_i B_1(x_i)B_2(x_i)N^3(x_i) \end{bmatrix}$$

$$\begin{bmatrix} \sum_i B_2(x_i)N(x_i) & \sum_i B_1(x_i)N^2(x_i) & \sum_i B_2(x_i)N^2(x_i) \\ \sum_i B_1(x_i)B_2(x_i)N(x_i) & \sum_i B_1^2(x_i)N^2(x_i) & \sum_i B_1(x_i)B_2(x_i)N^2(x_i) \\ \sum_i B_2^2(x_i)N(x_i) & \sum_i B_1(x_i)B_2(x_i)N^2(x_i) & \sum_i B_2^2(x_i)N^2(x_i) \\ \sum_i B_1(x_i)B_2(x_i)N^2(x_i) & \sum_i B_1^2(x_i)N^3(x_i) & \sum_i B_1(x_i)B_2(x_i)N^3(x_i) \\ \sum_i B_2^2(x_i)N^2(x_i) + \lambda & \sum_i B_1(x_i)B_2(x_i)N^3(x_i) & \sum_i B_2^2(x_i)N^3(x_i) \\ \sum_i B_1(x_i)B_2(x_i)N^3(x_i) & \sum_i B_1^2(x_i)N^4(x_i) + \lambda & \sum_i B_1(x_i)B_2(x_i)N^4(x_i) \\ \sum_i B_2^2(x_i)N^3(x_i) & \sum_i B_1(x_i)B_2(x_i)N^4(x_i) & \sum_i B_2^2(x_i)N^4(x_i) + \lambda \end{bmatrix}$$

Recordemos que tendremos una matriz por cada ventana y por cada canal, a la cual se ajusta un modelo de regresión. El número de condición de las matrices calculadas en varias secciones es grande, por lo que para un manejo adecuado de la precisión se escalan todos estos en el rango de  $[0,1]$  y se realiza

una suma a los elementos de la diagonal principal con los valores de  $\lambda$  para obtener un mejor resultado numérico en la solución. La matriz  $\mathbb{A}$  es simétrica y positiva definida, por lo que uno de los métodos ampliamente usados para resolver estos sistemas lineales es mediante la factorización de *Cholesky* (ver referencia [8]); muy conocido por su eficiencia y la estabilidad numérica.

Este modelo de regresión es aplicado a cada bloque de  $n \times m$  pixeles en las imágenes de  $500m$ . A todos los pixeles del bloque se les asignan los mismos parámetros. La forma de avance de los bloques puede implicar un traslape de modelos en cada zonas, así que dichos parámetros en la zona de traslapes pueden ser elegidos a través de un estimador como la media.

Cada ventana es independiente, lo que permite ser paralelizado, lográndose con ello una gran eficiencia.

En el método de CIMAT-CONABIO se observó que debido a la existencia de datos atípicos los parámetros de las regresiones podrían variar notablemente entre una ventana y otra, provocando un efecto de mosaico (píxeleo) en los pixeles de las fronteras, esto se debe a que no hay una transición suave entre los parámetros calculados en la regresión de una ventana con la contigua. Otra estrategia (en vez de ventanas traslapadas) es asignar los parámetros calculados sólo a los pixeles centrales de cada ventana y calcular los demás parámetros mediante interpolación bi-cúbica [22].

A pesar de la interpolación de los parámetros en los datos, existen datos que por su naturaleza están fuera de los rangos establecidos en las bandas, esto se debe a que fueron estimados incorrectamente durante el proceso de regresión o incluso fueron erróneamente adquiridos por el sensor. Por ello se aplica el proceso de eliminación de datos atípicos, que serán los datos que se encuentren fuera de un rango previamente establecido. Para corregir aquellos datos fuera del intervalo aplico la máscara (M ocho conectado) a los datos atípicos para sustituir dichos valores por un promedio pesado de los vecinos:

$$M = \frac{1}{2} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$x_{new} = \sum_{k=-1}^1 \sum_{l=-1}^1 M_{k+1l+1} * x_{i+k,j+l}$$

En el caso particular de MODIS, para preservar la consistencia radiométrica, el paso final involucra un proceso de normalización para asegurarse de que el resultado es consistente en comparación con la banda de baja resolución original MODIS a 500m al nadir y VIIRS a 375m *at nadir*. Se debe realizar con un promedio ponderado de un grupo de  $3 \times 2$  píxeles de 250m con pesos de 0.25 para los dos píxeles de los bordes y 0.5 para el píxel central [0.25 0.5 0.5]. Se observó que el proceso originalmente propuesto por Trishchenko era inconsistente y tenía múltiples soluciones, por lo que el algoritmo CIMAT-CONABIO propone realizar la normalización mediante la minimización del siguiente problema de optimización:

$$\begin{aligned} \min_a F(a) &= \frac{1}{2} \|Da - m\|^2 + \frac{1}{2} \lambda \|a - 1\|^2 \\ \min_a \frac{1}{2} [2\rho_1 a_1 + \rho_1 a_2 - 3r_1]^2 &+ \frac{1}{2} \sum_{k=2}^n [\rho_{2k-2} a_{2k-2} + 2\rho_{2k-1} a_{2k-1} + \rho_{2k} a_{2k} - 2r_k]^2 \end{aligned}$$

Donde  $\rho_i$  es el promedio de la reflectancia de dos píxeles de 250m localizados en la misma columna de dos líneas consecutivas que contribuyen a un píxel de 500m,  $r_i$  es el valor de la reflectancia del píxel a 500m,  $a_i$  es el factor desconocido de la normalización y  $n$  es el número de elementos en la fila de la imagen de baja resolución.

Es importante remarcar que la normalización arriba descrita se basa en la geometría del sensor MODIS, de como los píxeles de baja resolución se sobrelapan con los de alta, el usar esa normalización para el caso de VIIRS, ver las figuras 1.3 y 1.2, a pesar de que mejora la correlación de las bandas no está sustentada en ningún modelo reportado del sensor.

## Capítulo 2

# Regresión sobre componentes

### 2.1. Mínimos Cuadrados

En esta sección haré una revisión del método de mínimos cuadrados. El contenido aquí expuesto se basa en el material correspondiente a las referencias [3] y [5].

El término mínimos cuadrados describe un enfoque frecuentemente usado para resolver sistemas de ecuaciones sobredeterminados. En lugar de resolver las ecuaciones exactamente, se busca minimizar la suma de los cuadrados de los residuales.

Sea  $x$  una variable independiente y sea  $y(x)$  una función desconocida de  $x$  que se desea aproximar, luego suponiendo que se tienen  $m$  observaciones  $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ , donde  $y_i \equiv y(x_i)$ , con  $i = 1, 2, \dots, m$ . La idea es modelar  $y(x)$  por medio de una combinación de  $n$  funciones base  $\phi_1(x), \phi_2(x), \dots, \phi_n(x)$ . En el caso lineal suponemos que la función que se ajusta a los datos es una *combinación lineal* de la forma

$$y(x) = c_1\phi_1(x_i) + c_2\phi_2(x_i) + \dots + c_n\phi_n(x_i) \quad i = 1, 2, \dots, m \quad (2.1)$$

La expresión 2.1 constituye un sistema de  $m$  ecuaciones con  $n$  incógnitas  $(c_1, c_2, \dots, c_n)$ . En el ajuste de las curvas el número de funciones base  $n$  es generalmente menor que el número de datos  $m$ , es decir,  $m > n$ . En forma matricial la condición 2.1 puede verse como:

$$\begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_n(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_n(x_2) \\ \vdots & & \ddots & \vdots \\ \phi_1(x_m) & \phi_2(x_m) & \cdots & \phi_n(x_m) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (2.2)$$

A la matriz de este sistema  $A = (a_{ij})$  con  $a_{ij} = \phi_j(x_i)$  se le denomina *matriz de diseño*. Las funciones base  $\phi_i(x)$  con  $i = 1, \dots, n$  pueden ser funciones no lineales de  $x$ , pero los coeficientes y parámetros  $c_j$  aparecen en el modelo en la forma lineal cuando se trata de un ajuste lineal. Dependiendo del problema particular y el objetivo de estudio, las funciones base  $\phi_i(x)$  pueden escogerse de muchas maneras, e incluso pueden depender de ciertos parámetros. Algunas elecciones comunes pueden ser, entre otras: polinomios,  $\phi_j(x) = x^{j-1}$ , funciones racionales,  $\phi_j(x) = x_{j-1}/(\alpha_0 + \alpha_1 x + \dots + \alpha_{n-1} x^n - 1)$ , con  $\alpha_0, \dots, \alpha_{n-1}$  parámetros dados; exponenciales,  $\phi_j(x) = e^{-\lambda_j x}$ , con parámetros de decaimiento  $\lambda_i$ .

Dado que  $m > n$ , el sistema  $Ac = y$  dado por 2.2 es sobredeterminado, y por lo tanto, tiene una solución solo si el vector de datos  $y$  se encuentra en el espacio de la imagen de  $A$ , denotado por  $Im(A)$  y por lo tanto en general, no es posible encontrar una solución  $c$  del sistema 2.2. Entonces el problema es buscar los coeficientes de la función 2.1 que mejor ajusten los datos. El enfoque de mínimos cuadrados consiste en buscar aquel vector de coeficientes  $c$  que minimice el residual  $r = y - Ac$ . Si denotamos la norma Euclidiana en  $\mathbb{R}^m$  por  $\|\cdot\|$ , entonces el problema consiste en resolver

$$\min_{c \in \mathbb{R}^n} \|Ac - y\|^2. \quad (2.3)$$

Es decir, para encontrar el ajuste de mínimos cuadrados debemos encontrar el vector de coeficientes  $c = (c_1, \dots, c_n)^T$  que minimiza la suma de cuadrados:

$$\min_{c \in \mathbb{R}^n} \sum_{i=1}^m (c_1 \phi_1(x_i) + c_2 \phi_2(x_i) + \cdots + c_n \phi_n(x_i) - y_i)^2 \quad (2.4)$$

Como vimos en la sección 1.2 la solución de mínimos cuadrados se obtiene resolviendo el sistema  $(AA^T)\theta = A^T b$ .

En el Apéndice .1 se muestra el código en Python para generar un conjunto de datos aleatorios y realizar con estos un ajuste de mínimos cuadrados. El ejemplo siguiente se realizó haciendo uso de este código.

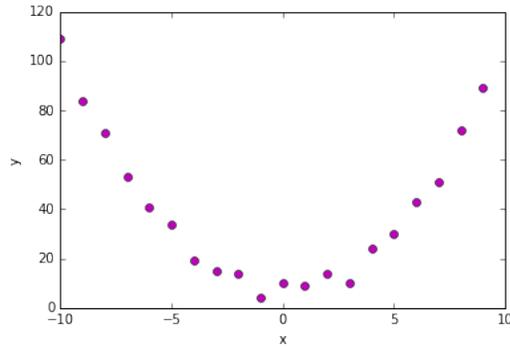
Ahora presento un ejemplo de un ajuste haciendo uso del método descrito. Para empezar se generan dos conjuntos de datos, el primer conjunto ( $X$ ) corresponde al conjunto de variables independientes y son los números del -10 al 9. El segundo conjunto ( $Y$ ) corresponde al conjunto de variables dependientes y es una función cuadrática de  $X$  más ruido, es decir  $y_i = x_i^2 + \epsilon$ , donde  $i = 1, 2, \dots, 20$  y  $\epsilon$  es un número aleatorio entre 1 y 10. Obteniendo así, los conjuntos de datos :

x	-20	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4
y	109	84	71	53	41	34	19	15	14	4	10	9	14	10	24

5	6	7	8	9
30	43	51	72	89

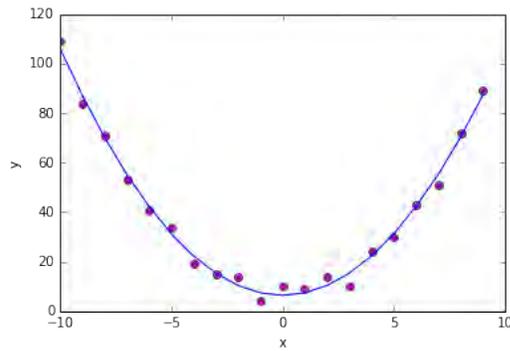
En la figura 2.1 se observa la gráfica de los datos anteriores los cuales, como se puede ver, se aproximan a una función cuadrática.

Ahora se quiere ajustar una función del tipo  $y = \theta_1 + \theta_2 x + \theta_3 x^2$  así que la matriz de diseño  $A$  queda de la siguiente manera:



**Figura 2.1:** Gráfica de los puntos generados.

$$A = \begin{bmatrix} 1 & -10 & 100 \\ 1 & -9 & 81 \\ 1 & -8 & 64 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 9 & 81 \end{bmatrix}$$



**Figura 2.2:** Curva de ajuste calculada con mínimos cuadrados.

Posteriormente a partir del del sistema  $(AA^T)\theta = A^Tb$ , con  $b = Y$  y  $\theta = (\theta_1, \theta_2, \theta_3)^T$ . El vector  $\theta$  o vector de soluciones se obtiene resolviendo el sistema de la siguiente manera:

$$\theta = (A^T A)^{-1}(A^T b)$$

El vector de soluciones que calculé en Python es  $\theta = (6.4, 0.01, 0.99)^T$  y en la gráfica 2.2 se observa la curva que se ajustó con mínimos cuadrados.

## 2.2. Análisis de Componentes Principales

El material de esta sección se revisará el Análisis de Componentes principales y se presenta una versión resumida de las fuentes [9] y [12].

Cuando se colecta información de una muestra de datos, lo más común es tomar en cuenta el mayor número posible de variables con el fin de tener una buena descripción de nuestros datos. Sin embargo, si tomamos demasiadas variables tendremos que considerar un número elevado de coeficientes de regresión, dificultando la visualización de las relaciones entre las variables. Otro problema que existe cuando tenemos demasiadas variables es la fuerte correlación que podría existir entre las variables. Es decir, si una variable se explica como la combinación lineal de otras, podríamos prescindir de ella. Con lo anterior vemos que es necesario y factible reducir el número de variables.

El análisis de componentes principales (PCA por sus siglas en inglés) es un método que permite reducir las dimensiones de un conjunto de datos, conservando, en la medida de lo posible, la variación presente en las variables que lo componen, perdiendo así, la menor cantidad posible de información útil.

La idea principal es la siguiente: dado un conjunto de datos  $\mathbf{X}$  de dimensión  $n \times m$  ( $n$  observaciones de  $m$  variables cada una) se analiza si es posible representar adecuadamente esta información con un número menor de variables llamadas componentes principales, los cuales tienen las características siguientes:

- Están contruidos como combinaciones lineales de las variables originales.
- Están ordenados de tal manera que unos pocos de los primeros retienen la mayor variación presente en las variables originales.

- El número de componentes principales es inferior a la menor de las dimensiones de  $\mathbf{X}$ .

Se considera una serie de variables  $(x_1, x_2, \dots, x_n)$  sobre un grupo de objetos o individuos y se trata de calcular, a partir de ellas, un nuevo conjunto de variables  $(y_1, y_2, \dots, y_n)$ , no correlacionadas (ortogonales:  $y_i^T y_j = 0, i \neq j$ ) entre sí, cuyas varianzas vayan decreciendo progresivamente.

Cada  $y_j$ , con  $j = 1, 2, \dots, n$  es una combinación lineal de las  $x_1, x_2, \dots, x_n$  originales, es decir:

$$y_j = a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jn}x_n = \mathbf{a}_j^T \mathbf{x}$$

donde  $\mathbf{a}_j^T = (a_{1j}, a_{2j}, \dots, a_{nj})$  es un vector de constantes y

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Si lo que se desea es maximizar la varianza de  $Y$ , una forma simple de lograrlo sería aumentar los coeficientes  $a_{ij}$ . Por ello, para mantener la ortogonalidad de la transformación se impone que el módulo del vector  $\mathbf{a}_j^T = (a_{1j}, a_{2j}, \dots, a_{nj})$  sea 1, es decir

$$\mathbf{a}_j^T \mathbf{a}_j = \sum_{k=1}^p a_{kj}^2 = 1 \quad (2.5)$$

El primer componente se calcula eligiendo  $\mathbf{a}_1$  de modo que  $y_1$  tenga la mayor varianza posible, sujeta a la restricción de que  $\mathbf{a}_1^T \mathbf{a}_1 = 1$ . El segundo componente principal se calcula obteniendo  $\mathbf{a}_2$  con las mismas restricciones de  $\mathbf{a}_1$  y de modo que la variable obtenida,  $y_2$  no este correlacionada con  $y_1$ . Del mismo modo que se eligen  $y_3, y_4, \dots, y_n$ , incorreladas entre sí, de manera que las variables aleatorias obtenidas vayan teniendo cada vez menor varianza.

Existen diferentes maneras de realizar PCA, a continuación explicaré una de las más utilizadas.

Una manera de realizar PCA es haciendo uso de la **Descomposición en valores singulares (SVD)**, por sus siglas en inglés) de la matriz  $\mathbf{X}$ ; los detalles de este método se pueden revisar en las referencias [8] y [21]. El primer paso es verificar que la matriz  $\mathbf{X}$  este centrada a media cero, de no ser así, se realizan las operaciones necesarias para lograr lo anterior. Una vez que la matriz  $\mathbf{X}$  esta centrada a media cero se realiza la descomposición en valores singulares de la misma, como se muestra a continuación:

$$\mathbf{SVD}(\mathbf{X}) = U\Delta V^T$$

donde:

$U$  : Eigen-vectores de la matriz  $XX^T$ .  $U \in \mathbb{R}^{n \times n}$

$\Delta$  : Matriz asociada a los eigen-valores de las matrices  $X^T X$  y  $XX^T$ .  $\Delta \in \mathbb{R}^{n \times m}$  y es una matriz diagonal.

$V^T$  : Eigen-vectores de la matriz  $X^T X$ .  $V^T \in \mathbb{R}^{m \times m}$ .

El siguiente paso es analizar la matriz  $\Delta$  para decidir el número de componentes principales que se usarán para describir a la matriz original. Además de que esta matriz es diagonal tiene la propiedad de que sus valores están en orden descendente, lo cual indica el grado de variación que conserva cada uno de los componentes principales. Con el conocimiento de lo anterior decidimos cuantos de estos números ayudan a retener la mayor variación posible y definimos **cp** (número de componentes principales). Una vez que se define **cp**, se debe redefinir  $V^T$  de tal manera que la nueva  $V^T$  ( $NV^T$ ) tenga únicamente **cp** renglones, es decir  $NV^T$  corresponde a los **cp** primeros renglones de  $V^T$ . Una vez que se tiene  $NV^T$ , calculamos la matriz de componentes principales, a la cual llamamos  $Y$ , y la calculamos como sigue:

$$Y = X * NV$$

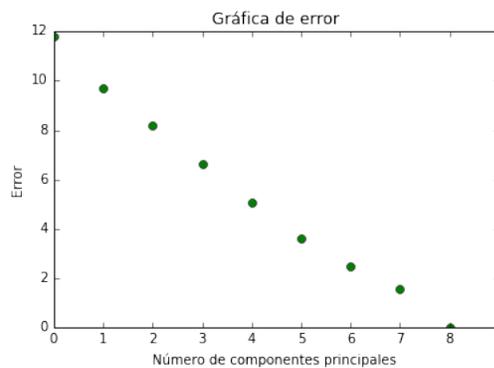
donde  $NV = (V^T)^T$

Si se quiere hacer la reconstrucción de la matriz original (regresarla a su dimensión original haciendo uso de los componentes principales), calculamos la reconstrucción de  $X$  ( $X'$ ) como sigue:

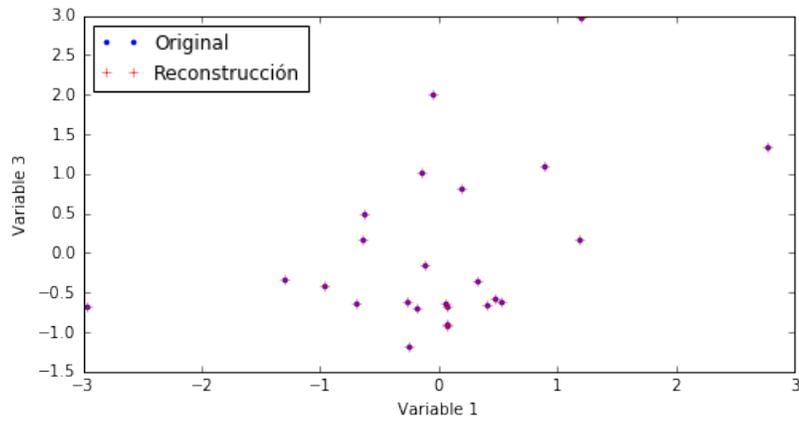
$$X' = Y * NV^T$$

El código en Python para realizar PCA sobre un conjunto de datos  $X$  de dimensiones  $(n \times m)$ , es decir  $n$  muestras de  $m$  variables cada una se puede consultar en el Apéndice .2.

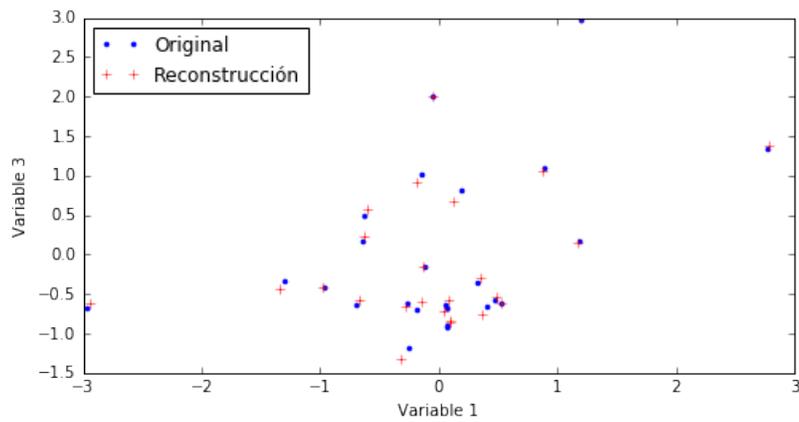
En la figura 2.4 se muestra un ejemplo realizado con el código en python del Apéndice .2 y tomando las primeras 25 muestras de la base de datos “Places Rated data”, la cual se puede consultar en el Apéndice .3. Para visualizar los resultados de las pruebas se construyó una gráfica tomando en cuenta las variables 1 (para graficar en el eje x) y 3 (para graficar en el eje y) de tal manera que se observen los datos originales, contra la reconstrucción de los mismos haciendo PCA. En las figuras 2.4 y 2.5 se pueden observar cuatro reconstrucciones distintas, en las cuales variamos en número de los componentes principales para que se pueda apreciar como es que el error entre los datos originales y la reconstrucción de los mismos varía de acuerdo al número de componentes utilizados en la reconstrucción. Dicho error se puede observar en la gráfica 2.3 y corresponde a la norma de la diferencia entre los datos originales y la reconstrucción de los mismos.



**Figura 2.3:** Error de acuerdo al número de componentes principales.

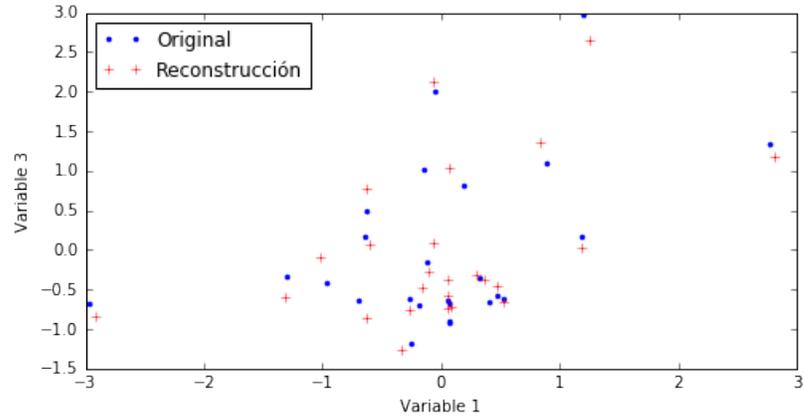


(a) Usando 9 componentes principales.

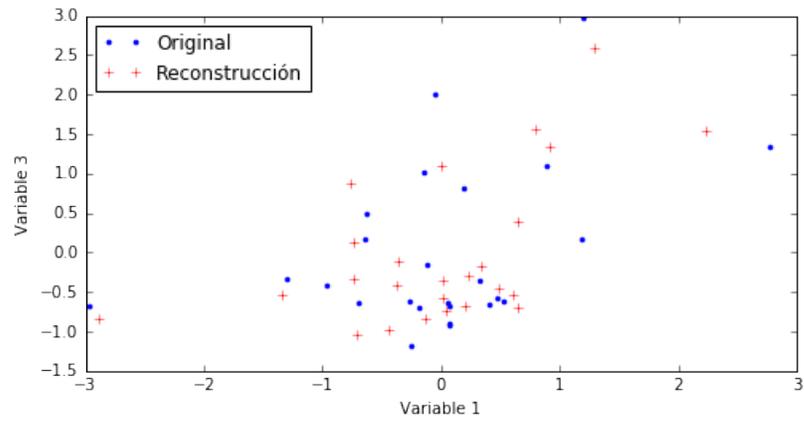


(b) Usando 8 componentes principales.

**Figura 2.4:** Ejemplo del método de Componentes Principales.



(a) Usando 7 componentes principales.



(b) Usando 5 componentes principales.

**Figura 2.5:** Ejemplo del método de Componentes Principales.

### 2.3. Regresión con Componentes Principales

Ahora se verá como combinar la reducción de componentes con el método de mínimos cuadrados con la finalidad de hacer más eficiente y robusto el cálculo de los coeficientes de regresión.

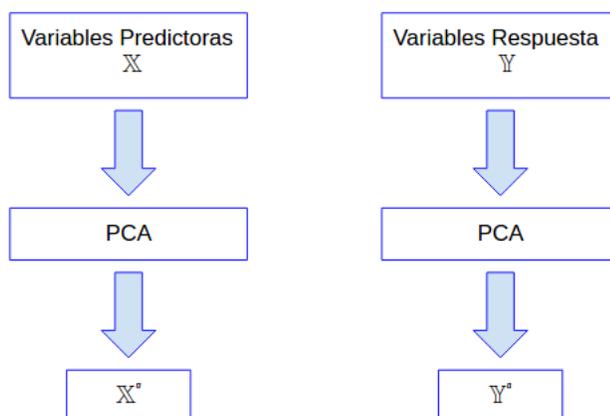
Se asume  $\mathbb{X}$  al conjunto de variables predictoras y  $\mathbb{Y}$  el conjunto de variables respuesta. De tal forma que para cada  $x$  se tiene una  $y$  correspondiente y se considera que tanto  $x \in \mathbb{X}$  como  $y \in \mathbb{Y}$  son de dimensión grande, digamos  $n$  y  $k$  respectivamente. Luego las  $m$  observaciones de  $x$  y  $y$  se podrían escribir como las siguientes matrices:

$$\mathbb{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad \mathbb{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1k} \\ y_{21} & y_{22} & \cdots & y_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mk} \end{bmatrix}$$

Ahora, supongamos que se desea realizar un ajuste por mínimos cuadrados de  $\mathbb{Y}$  como combinación lineal de las variables contenidas en  $\mathbb{X}$ . Un problema que enfrentaremos será la gran dimensión de  $\mathbb{X}$  y  $\mathbb{Y}$  y las correlaciones que entre las variables existan, es decir, si algunas columnas de  $\mathbb{X}$  se pudieran explicar en términos algunas otras columnas de  $\mathbb{X}$ , entonces podríamos y deberíamos prescindir de algunas columnas de  $\mathbb{X}$ , igualmente las columnas de  $\mathbb{Y}$  que sean combinaciones lineales de otras no tendríamos que predecirlas. Por lo que se debe reducir las dimensiones de ambas variables previamente a calcular la regresión. Este razonamiento, aunque lógico, tiene una debilidad, la cual analizaremos más adelante, por lo pronto continuaremos desarrollando esta idea. El desarrollo del método anteriormente descrito se hará en forma visual y se irá introduciendo al lector paso por paso.

Lo primero es realizar una reducción de las dimensiones de  $\mathbb{X}$  y  $\mathbb{Y}$  haciendo uso de PCA como se muestra en la figura 2.6, obteniéndose:

$$\mathbb{X}' = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n'} \\ x_{21} & x_{22} & \cdots & x_{2n'} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn'} \end{bmatrix} \quad \mathbb{Y}' = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1k'} \\ y_{21} & y_{22} & \cdots & y_{2k'} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mk'} \end{bmatrix}$$



**Figura 2.6:** Etapa de entrenamiento.

sabiendo que  $k' < k$  y que  $n' < n$ .

Una vez que tenemos  $\mathbb{X}'$  y  $\mathbb{Y}'$  debemos realizar un ajuste por mínimos cuadrados para calcular  $\mathbb{Y}'$  a partir de  $\mathbb{X}'$ , es decir debemos encontrar el valor de  $\theta$  resolviendo  $\min_{\theta \in \mathbb{R}^n} \|\mathbb{A}'\theta - \mathbb{Y}'\|^2$ .

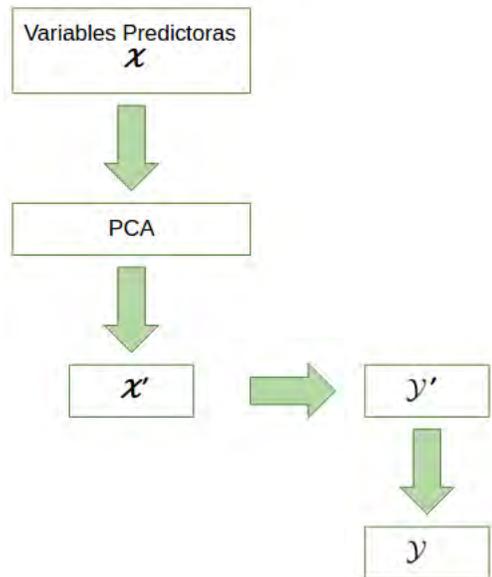
A la etapa anterior se le conoce como etapa de entrenamiento o aprendizaje. Si posteriormente tengo un conjunto de datos no observados ( $\mathcal{Y}$ ) a predecir, es decir, tengo nuevos valores de  $\mathbb{X}$  ( $\mathcal{X}$ ), para realizar la predicción de las  $y$ 's correspondientes a  $\mathcal{X}$  tendríamos que reducir la dimensión de  $\mathcal{X}$  al tamaño de  $\mathbb{X}'$  obteniendo así  $\mathcal{X}'$  y posteriormente aplicar el ajuste de mínimos cuadrados obtenido en la etapa de entrenamiento para predecir  $\mathcal{Y}'$ . Por último, una vez que obtuvimos  $\mathcal{Y}'$  sabemos que esta está en la dimensión del espacio de los PCA's, es decir es de dimensión  $m \times n'$  y queremos regresarla a su espacio original de  $\mathbb{Y}$ , es decir, que sea de dimensión  $m \times n$ . Lo anterior se observa en la figura 2.7.

El procedimiento anterior se describe matemáticamente en los siguientes pasos:

1. Cálculo de  $n'$  y  $k'$  (componentes principales)

$$U_{\mathbb{X}} \delta_{\mathbb{X}} V_{\mathbb{X}}^T = SVD(\mathbb{X})$$

$$U_{\mathbb{Y}} \delta_{\mathbb{Y}} V_{\mathbb{Y}}^T = SVD(\mathbb{Y})$$



**Figura 2.7:** Predicción de variables no observadas usando PCA.

2. Reducción de dimensionalidad

$$\mathbb{X}' = U_{\mathbb{X}}\mathbb{X}$$

$$\mathbb{Y}' = U_{\mathbb{Y}}\mathbb{Y}$$

3. Cálculo de parámetros de regresión

$\min_{\theta \in \mathbb{R}^n} \|\mathbb{A}'\theta - \mathbb{Y}'\|^2$  donde  $\mathbb{A}'$  es la matriz de diseño (de mínimos cuadrados) generada con  $\mathbb{X}'$ .

4. Predicción de datos no observados

$$\mathcal{X}' = U_{\mathcal{X}}\mathcal{X}$$

$$\mathcal{Y}' = \mathcal{A}'\theta$$

donde  $\mathcal{A}'$  es la matriz de diseño correspondiente a  $\mathcal{X}'$

5. Regreso a dimensión original

$$\mathcal{Y} = U_x^T \mathcal{Y}'$$

Al realizar el procedimiento podemos encontrarnos con el siguiente problema: cabe la posibilidad de que al reducir las dimensiones tanto de  $\mathbb{X}$  como de  $\mathbb{Y}$  (haciendo el análisis de componentes principales en cada caso) se dejen fuera los elementos que permitan hacer una buena predicción de  $\mathbb{Y}$  a partir

de  $\mathbb{X}$ . Por lo que necesitamos un método que nos ayude a conservar no los componentes principales, si no los más correlacionados entre las variables  $X$  y  $Y$ . Este método ya existe y se llama Regresión de Mínimos Cuadrados Parciales (PLS, por sus siglas en inglés) y se verá en la siguiente sección.

## 2.4. Regresión de Mínimos Cuadrados Parciales

La regresión de mínimos cuadrado parciales (PLS, por sus siglas en inglés) es una técnica reciente que combina las características de análisis de componentes principales y la regresión múltiple. Su principal objetivo es predecir o analizar un conjunto de variables dependientes ( $Y$ ) a partir de un conjunto de variables independientes o predictores ( $X$ ). Esta predicción se logra extrayendo de los predictores un conjunto de factores ortogonales llamados variables latentes que tiene el mayor valor de predicción. Es un método estadístico que tiene relación con la regresión de componentes principales. Este método se utiliza para encontrar las relaciones fundamentales entre las matrices  $X$  y  $Y$ . La regresión de mínimos cuadrados parciales es particularmente útil cuando necesitamos predecir un conjunto de variables a partir de un conjunto grande de variables independientes (predictores).

La idea principal de este método es primero realizar un análisis de componentes principales (PCA) en la matriz de diseño y después usar solamente los primeros  $k$  componentes para realizar la regresión, pero dando preferencia a los que están mayormente correlacionados con las variables dependientes.

En PLS se asume que las matrices de datos  $X$  y  $Y$  están dadas por los siguientes modelos lineales:

$$X = TP^T + N_1, \quad Y = UQ^T + N_2 \quad (2.6)$$

con

$$U = T\beta, \quad P^T P = I, \quad Q^T Q = I \quad (2.7)$$

es decir, podemos ver a matriz  $P$  y la matriz  $Q$  son matrices base ortonormales y existe una transformación lineal entre las matrices de coeficientes  $U$  y  $T$ . Para resolver este problema es necesario imponer algunas restricciones, en particular en PLS usa el siguiente enfoque.

Sea la matriz  $P = P[i]$  con  $i = 1, 2, \dots, n$  donde los  $P_i$  son vectores columna, el método de regresión de mínimos cuadrados parciales puede verse como la implementación a la solución del problema expuesto a continuación:

$$\arg \max_p ||\text{corr}(Y, PX)||^2 \text{var}(PX) \quad (2.8)$$

$$\begin{aligned} & \text{sujeto a} \\ & p_i^T p_j = 0 \text{ y } p_i^T p_i = 1 \\ & \text{donde } p_i \text{ es la columna } i\text{-ésima de } P \end{aligned}$$

De lo anterior puede verse que se busca una matriz ortogonal  $P$  que cumpla lo siguiente:

- a) Maximice la correlación entre la variable  $Y$  y la rotación  $PX$ .
- b) Los vectores columnas (ortonormales de  $P$ ) maximicen la varianza de  $X$ .

Como podemos ver, PLS difiere de PCA en que, además de pedir maximizar la varianza, también pide maximizar la correlación de la proyección con la variable dependiente.

Un algoritmo que implementa una solución al problema 2.8 es el llamado NIPALS [17], es el siguiente:

1. Inicialice  $\mathbf{u}_1$ , por ejemplo, con la primera de la matriz  $\mathbf{Y}$ .
2.  $\mathbf{w}_1 = \frac{\mathbf{X}^T \mathbf{u}_1}{\mathbf{u}_1^T \mathbf{u}_1}$ .
3.  $\mathbf{w}_1 = \frac{\mathbf{w}_1}{\|\mathbf{w}_1\|}$ .
4.  $\mathbf{t}_1 = \mathbf{X} \mathbf{w}_1$ .

$$5. \mathbf{c}_1 = \frac{\mathbf{Y}^T \mathbf{t}_1}{\mathbf{t}_1^T \mathbf{t}_1}.$$

$$6. \mathbf{c}_1 = \frac{\mathbf{c}_1}{\|\mathbf{c}_1\|}.$$

$$7. \mathbf{u}_1^* = \mathbf{Y} \mathbf{c}_1.$$

$$8. \mathbf{u}_\Delta = \mathbf{u}_1^* - \mathbf{u}_1.$$

$$9. \Delta \mathbf{u} = \mathbf{u}_\Delta^T \mathbf{u}_\Delta.$$

10. Si  $\Delta \mathbf{u} < \varepsilon$ , entonces pare; si no  $\mathbf{u}_1 = \mathbf{u}^*$  y vuelva al paso 2.

Aunque la derivación del algoritmo NIPALS está fuera del alcance de esta tesis, daremos a continuación una breve justificación a su sustento teórico. Recomendamos al lector revisar el material de las referencias [6] y [9] para tener mayor información sobre este algoritmo.

Si en el paso 2 del algoritmo se está ejecutando la iteración  $j + 1$ , entonces:

$$\mathbf{w}_1^{j+1} = \frac{\mathbf{X}^T \mathbf{u}_1^j}{(\mathbf{u}_1^j)^T \mathbf{u}_1^j}.$$

Ahora, considerando el paso 7. Con  $\mathbf{u}_1^j$ , reemplazando  $\mathbf{c}_1^j$  por los pasos 6 y 5, reemplazando  $\mathbf{t}_1^j$  por el paso 4 y  $\mathbf{w}_1^j$  por el paso 3 se obtiene:

$$\mathbf{w}_1^{j+1} = \mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \mathbf{w}_1^j k$$

donde la constante  $k$  depende de las normas de los diferentes vectores. Esta última ecuación es en efecto el método de las potencias para el cálculo de eigen valores, donde  $\mathbf{w}_1$  es el eigenvector asociado al eigenvalor mas grande de la matriz  $\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}$ . De manera similar puede mostrarse que:

$$\mathbf{c}_1^{j+1} = \mathbf{Y}^T \mathbf{X} \mathbf{X}^T \mathbf{Y} \mathbf{c}_1^j k$$

converge al eigenvector asociado al eigenvalor más grande de la matriz  $\mathbf{Y}^T \mathbf{X} \mathbf{X}^T \mathbf{Y}$ .

Regresando al algoritmo NIPALS. Una vez concluido el paso 10, el método continúa con los siguientes pasos:

$$11. \mathbf{p}_1 = \frac{\mathbf{X}^T \mathbf{t}_1}{\mathbf{t}_1^T \mathbf{t}_1}.$$

$$12. \mathbf{q}_1 = \frac{\mathbf{Y}^T \mathbf{u}_1}{\mathbf{u}_1^T \mathbf{u}_1}.$$

$$13. \mathbf{d}_1 = \frac{\mathbf{u}_1^T \mathbf{t}_1}{\mathbf{t}_1^T \mathbf{t}_1}.$$

$$14. \mathbf{X}_1 = \mathbf{X} - \mathbf{t}_1 \mathbf{p}_1^T \text{ y } \mathbf{Y} = \mathbf{Y} - \mathbf{d}_1 \mathbf{t}_1 \mathbf{q}_1^T.$$

La línea 14 indica como se actualizan los datos  $X$  y  $Y$  substrayendo el componente recién calculado. Ahora definamos las matrices  $P$ ,  $Q$ ,  $T$  y  $U$  cuyas columnas corresponden a los vectores  $p, q, t$  y  $u$  calculados mediante el algoritmo NIPALS. Luego, dado que PLS calculó  $T$  y  $U$  que cumplen con el modelo lineal 2.6  $\beta$  se calcula mediante:

$$\min_{\beta} \|U - T\beta\|^2 \quad (2.9)$$

Sustituyendo  $U$  en el modelo 2.7 tenemos:

$$Y = T\beta Q^T = X P \beta Q^T.$$

Esta fórmula nos permite hacer la predicción para cualquier  $x'$ :

$$y' = x' P \beta Q^T. \quad (2.10)$$

Una implementación propia en Python de NIPALS se puede ver en el Apéndice .4

## 2.5. Algoritmo de fusión de imágenes propuesto

En esta sección explico en que consiste la metodología del algoritmo de fusión de imágenes satelitales VIIRS que propongo y desarrollo a lo largo de la elaboración de este trabajo de tesis. El objetivo de dicho algoritmo es me-

jorar la resolución espacial de las imágenes satelitales de las bandas de baja resolución  $M_3, M_4, M_8, M_{10}, M_{11}$  del sensor VIIRS. Con el fin anterior utilizo las bandas de alta resolución ( $I_1, I_2, I_3$ ) como bandas predictoras para inferir información acerca de las bandas de baja resolución ( $M_3, M_4, M_8, M_{10}, M_{11}$ ).

A continuación se muestra un listado de las etapas en las que consiste el método del algoritmo propuesto y en las secciones siguientes se desarrollarán con detalle cada una de ellas.

1. Creación de máscaras.
2. Reducción de escala.
3. Predicción de bandas de baja resolución.
4. Normalización.

### 2.5.1. Creación de máscaras

Una vez que se realizó la lectura de los datos, se utiliza la información de la tabla 1.2 para obtener máscaras de procesamiento de las imágenes que consisten en imágenes de tipo de datos binario, las cuales son creadas con el fin de diferenciar las áreas que se desea procesar de la imagen de áreas que consisten en píxeles muertos o que carecen de información en el escaneo de datos. La máscara se genera validando la información en cada una de las bandas que participan en el proceso; llamaremos “información válida” a la información que se desea procesar. Este es un paso muy importante ya que estas máscaras de procesamiento se tomarán en cuenta para algunos de los pasos siguientes con el fin de obtener los mejores resultados posibles al procesar solamente información válida.

### 2.5.2. Reducción de escala

Como se muestra en la figura 1.2 sabemos que en las imágenes VIIRS, un pixel en una imagen de baja resolución corresponde a 4 píxeles en una imagen de alta resolución. Por tanto se debe reducir la escala(resolución) de los

pixeles de las bandas de alta resolución a la escala de los pixeles de baja resolución. Lo anterior para tener tanto las bandas predictoras como las bandas respuesta en la misma escala y poder así crear un modelo para mejorar la resolución de las bandas de baja resolución.

Para realizar lo anterior se toman en cuenta las máscaras de procesamiento creadas en el paso anterior y realizo la reducción de dimensión haciendo uso de tres métodos diferentes con el fin de observar cual es el que me otorga los mejores resultados. Los métodos anteriores se realizan pixel a pixel, es decir, toman un grupo de 4 pixeles de alta resolución y los reducen a un pixel de baja resolución obteniendo como producto final una imagen de baja resolución por cada una de las imágenes de alta resolución.

Cabe mencionar que para cada imagen de baja resolución (por cada banda), se crea una máscara de procesamiento, ya que se puede dar el caso en el que ninguno de los 4 pixeles correspondientes tenga información válida y por lo tanto el pixel de baja resolución tampoco proporcionará información válida para el procesamiento de datos.

Se entiende que al finalizar este proceso obtendremos las bandas  $I_1$ ,  $I_2$ ,  $I_3$  en baja resolución, para fines de explicar las etapas siguientes llamaremos a estas bandas en baja resolución  $i_1$ ,  $i_2$ ,  $i_3$ ; respectivamente. A continuación se describen cada uno de los tres métodos que fueron utilizados para realizar la reducción de escala.

### **Media aritmética**

En este método se considera que todos los pixeles que tienen información válida nos brindan información confiable así que con ayuda de las máscaras de procesamiento verificamos cuántos y cuáles de los 4 pixeles correspondientes tienen información válida y se calcula el promedio de los mismos para obtener el pixel de baja resolución correspondiente. En el caso de que ninguno de los 4 pixeles cuente con información válida se indica en la máscara de procesamiento correspondiente.

### Mediana

En este método se considera que pueden existir píxeles que a pesar de tener información válida no contienen información confiable, por tanto se ordena el valor de los píxeles que contienen información válida y se obtiene el valor del píxel de baja resolución correspondiente calculando la mediana de dichos valores, tratando de descartar así valores muy alejados del valor real. Al igual que el método anterior, si ninguno de los cuatro píxeles contiene información válida se especifica en el máscara correspondiente.

### Selección aleatoria

En este método, al igual que el primero, se considera que los cuatro píxeles contiene información confiable, así que se elige aleatoriamente el valor de uno de los cuatro píxeles para asignarlo al valor de baja resolución correspondiente. Si el píxel que elegimos no cuenta con información válida se especifica en la máscara de procesamiento correspondiente.

### 2.5.3. Predicción de las bandas de alta resolución

A continuación se enlistan y describen cada una de las subetapas correspondientes a esta etapa del método.

#### Preprocesamiento de datos

En esta subetapa del método se preparan los datos que servirán como entrenamiento para el algoritmo de PLS que se usará en una de las subetapas siguientes; para ello se realiza lo siguiente:

- I) **Selección de la ventana:** Se crea una subimagen a partir de la imagen original, la cual será procesada en la etapa siguiente. Para ello se considera las coordenadas del píxel superior izquierdo de la subimagen y su dimensión, además de los límites de la imagen original para ver si es posible crear la subimagen. Además de crear la subimagen

también se crea una submáscara a partir de la máscara de la imagen original.

- II) **Construcción de la matriz  $\mathbb{X}$  (matriz predictora de entrenamiento):** Se realiza la construcción de la matriz  $\mathbb{X}$ , que sirve como matriz predictora de entrenamiento del algoritmo PLS. El número de renglones de  $\mathbb{X}$  corresponde al número de píxeles de la ventana, es decir, si la ventana es de 200 píxeles por 200 píxeles el número de renglones de  $\mathbb{X}$  será 40,000; el número de columnas es tres e indica el valor de las bandas  $i_1, i_2, i_3$  (las bandas I's en baja resolución); respectivamente. Se puede decir entonces que el valor  $\mathbb{X}_{ij}$  de la matriz  $\mathbb{X}$  corresponde al valor de la banda  $j$  del pixel  $i$ , con  $i = 1, 2, 3$  y  $j = 1, 2, \dots, p$ , donde  $p$  es el número de píxeles de la ventana.
- III) **Construcción de la matriz  $\mathbb{Y}$  (matriz respuesta de entrenamiento):** Se realiza la construcción de la matriz  $\mathbb{Y}$  que sirve como matriz respuesta de entrenamiento para el algoritmo PLS. Al igual que la matriz  $\mathbb{X}$  el número de renglones corresponde al número de píxeles de la ventana y el número de columnas es cinco e indica el valor original de las bandas  $m_3, m_4, m_8, m_{10}, m_{11}$ . Se hace referencia a las bandas con m minúscula para indicar que están en baja resolución (resolución original).
- IV) **Construcción del vector indicador:** Una vez que se tiene  $\mathbb{X}$  y  $\mathbb{Y}$  se debe crear un vector que indique cuáles de los píxeles que forman las matrices  $\mathbb{X}$  y  $\mathbb{Y}$  son los que tienen información válida, y por tanto son los que se deben procesar. Para ello se hace uso de las submáscaras de procesamiento correspondientes a la ventana con la que estamos trabajando; para cada pixel se verifica si este contiene información válida en cada uno de sus valores de las bandas i's y de las bandas m's; de ser así, se asigna un uno a la posición correspondiente de este pixel en el vector indicador, de lo contrario se coloca un cero para hacer referencia a que este pixel no tiene que ser procesado. En resumen, el vector indicador es un vector binario de tamaño igual al número de píxeles de la ventana.
- V) **Construcción de la matriz  $\mathcal{X}$  (matriz predictora):** Una vez que se tienen los datos de entrenamiento se procede a construir la matriz  $\mathcal{X}$  que es la matriz que sirve como matriz predictora para calcular los valo-

res de las bandas  $M_3, M_4, M_8, M_{10}, M_{11}$ ; se denotan con M mayúscula para indicar que han sido mejoradas o están en alta resolución. Los renglones de la matriz  $\mathcal{X}$  corresponden a los pixeles de la ventana para las bandas I's antes de ser reducidas de dimensión. Es decir, equivale a cuatro veces el número de renglones de la matriz  $\mathbb{X}$ ; el número de columnas es seguirá siendo 3 y cada columna indica el valor de las bandas  $I_1, I_2, I_3$  (valores originales).

### **Etapa de aprendizaje o entrenamiento**

En esta etapa se utilizan  $\mathbb{X}$  y  $\mathbb{Y}$  para construir un modelo de la forma

$$\mathbb{Y}' = \mathbb{X}'P\beta Q^T.$$

El cálculo de las matrices  $P$  y  $Q$  y del vector  $\beta$  se realizan mediante PLS. Ver la sección 2.4 para mayor detalles.

Para construir este modelo se hace uso del vector indicador para procesar únicamente los pixeles que tengan información válida.

### **Regresión o Predicción**

En esta subetapa se aplica el modelo de la etapa anterior para calcular  $\mathcal{Y}$ , la cual tiene el mismo número de renglones que  $\mathcal{X}$ , tiene 5 columnas y cada una de estas representa el valor de las bandas  $M_3, M_4, M_8, M_{10}, M_{11}$  en alta resolución.

$$\mathcal{Y}' = \mathcal{X}'P\beta Q^T.$$

#### **2.5.4. Normalización**

Ya que se ha mejorado la resolución de las bandas de baja resolución, se llama  $m$  a la banda de baja resolución y  $M$  a la banda de alta resolución que calculamos, se realiza una normalización con el proceso matemática

siguiente:

1. Calculamos la media de  $m$  y  $M$ :

$$\mu = \frac{1}{k} \sum_k m_k \quad y \quad \eta = \frac{1}{l} \sum_l M_l, \text{ respectivamente.}$$

2. Calculamos la desviación estándar de  $m$  y  $M$ :

$$s = \sqrt{\frac{(\sum_k m_k) - \mu)^2}{k}} \quad y \quad \mathcal{S} = \sqrt{\frac{(\sum_l M_l) - \eta)^2}{l}}, \text{ respectivamente.}$$

3. Hacemos que  $M$  tenga media cero y varianza unitaria

$$\hat{M} = \left( \frac{M - \eta}{\mathcal{S}} \right)$$

4. Hacemos que la media y la desviación estándar de  $M$  sean iguales a las de  $m$

$$\hat{M} \leftarrow \hat{M}s + \mu$$

Con lo anterior se logra que tanto el promedio como la varianza de  $m$  y  $M$  sean iguales.

### 2.5.5. Variantes

El algoritmo descrito anteriormente es el resultado de una serie de pruebas y variaciones que se hicieron al mismo, siendo la versión presentada en este trabajo la que otorga mejores resultados. Dichas variaciones no serán descritas a detalle por fines prácticos, pero a continuación menciono algunas de las más relevantes.

La motivación de este algoritmo parte del hecho de que se pensaba que en el sensor VIIRS existían 5 bandas de alta resolución para inferir 5 bandas de baja resolución, así que el objetivo inicial era utilizar el algoritmo PLS para hacer una reducción de las bandas predictoras (de alta resolución) para inferir 5 bandas a partir de solamente 3; pero en el proceso descubrí que únicamente contaba con 3 bandas predictoras ya que las dos bandas restantes con las que se pensaba trabajar miden emisividad en el rango en

vez de reflectancia. Así que para resolver el problema anterior partí de la idea de no hacer reducción de las bandas y predecir 5 bandas a partir de 3, pero después de una serie de experimentos concluimos en predecir 5 bandas a partir de 2.

Con el fin de aumentar el número de variables predictoras decidí incluir el NDVI (Índice de vegetación normalizada), el cual incluí de diferentes maneras (el índice sólo, el índice al cuadrado, etc.) pero no aportó más información al proceso, así que decidí no utilizarlo. Esto se explica ya que, como podemos ver en la tabla 2.1 las bandas desde un principio (antes de ser procesadas) ya cuentan con una alta correlación entre ellas. Para construir está matriz se tomaron los valores correspondientes a una de las subimágenes de la sección de Resultados; tomando en cuenta los valores originales de las  $m$ 's y la reducción de las  $I$ 's para que todas las bandas estuvieran en la misma dimensión y poder crear así esta matriz.

	$i_1$	$i_2$	$i_3$	$m_3$	$m_4$	$m_8$	$m_{10}$	$m_{11}$
$i_1$	<b>1</b>	-0.28	0.09	0.88	0.93	-0.29	0.09	0.46
$i_2$	-0.28	<b>1</b>	0.60	-0.49	-0.38	0.90	0.60	0.32
$i_3$	0.09	0.60	<b>1</b>	-0.33	-0.20	0.84	0.99	0.90
$m_3$	0.88	-0.49	-0.33	<b>1</b>	0.98	-0.61	-0.33	0.03
$m_4$	0.93	-0.38	-0.20	0.98	<b>1</b>	-0.49	-0.20	0.16
$m_8$	-0.29	0.90	0.84	-0.61	-0.49	<b>1</b>	0.84	0.57
$m_{10}$	0.09	0.60	0.99	-0.33	-0.20	0.84	<b>1</b>	0.90
$m_{11}$	0.46	0.32	0.90	0.03	0.16	0.57	0.90	<b>1</b>

**Tabla 2.1:** Correlación entre bandas originales.

## 2.6. Implementación en Python

Con el fin de probar la eficiencia del algoritmo que propongo y mostrar los resultados obtenidos con este, hice la implementación de cada una de las etapas descritas con anterioridad en el lenguaje de programación Python, el ambiente que utilicé es Jupyter. La elección de este lenguaje se hizo tomando en cuenta que este lenguaje de programación ya cuenta con la implementación de muchos métodos matemáticos, entre ellos se encuentra el algoritmo de PLS, del cual hice uso en las subetapas de aprendizaje y regresión. Entre los paquetes de Python de los cuales hice uso para la implementación de

mi algoritmo destacan numpy, con el cual se calcularon algunas medidas de tendencia central como lo son la media aritmética y la mediana; h5py, con el cual leí e hice la extracción de la información de los archivos hdf5 en los que estaban las imágenes; sklearn, el cual contiene el algoritmo de PLS; matplotlib, con el cual creé todas las imágenes que se muestran en la sección siguiente; entre otros.



## Capítulo 3

# Resultados

Como se menciona al final de la sección anterior, las siguientes pruebas se obtuvieron mediante la programación y aplicación del algoritmo propuesto. Todas las gráficas, imágenes y tablas que se muestran a continuación son resultado de aplicar mi algoritmo y se construyeron en el lenguaje de programación Python.

### 3.1. Descripción de los datos

A continuación muestro las pruebas realizadas en imágenes facilitadas por la CONABIO que corresponden a diferentes partes de la república mexicana y que fueron tomadas en diferentes días, en cada una de las pruebas definiré la fecha y el área a la que corresponden. Las imágenes que procesé fueron previamente georeferenciadas, así que únicamente apliqué cada una de las etapas de mi algoritmo, obteniendo como resultado las bandas  $M_3, M_4, M_8, M_{10}$  y  $M_{11}$  en alta resolución. Todas las subimágenes (ventanas) originales y reconstrucciones que se muestran a continuación son de 200 píxeles por 200 píxeles, por lo tanto, las imágenes creadas a partir de mi algoritmo (mejoradas o de alta resolución) son de 400 por 400 píxeles. Dichas subimágenes se crearon a partir de las bandas  $M_3, M_4, M_8$  para poder mostrar de una manera gráfica los resultados obtenidos a través de nuestro algoritmo. A pesar de que tuve el cuidado de normalizar las imágenes

fusionadas de manera que tuvieran la misma media y varianza que las imágenes originales, al momento de desplegarse podemos ver una variación en los contrastes, esto se debe al escalamiento automático que hace la función de python que las despliega, sobre la cual no tengo control.

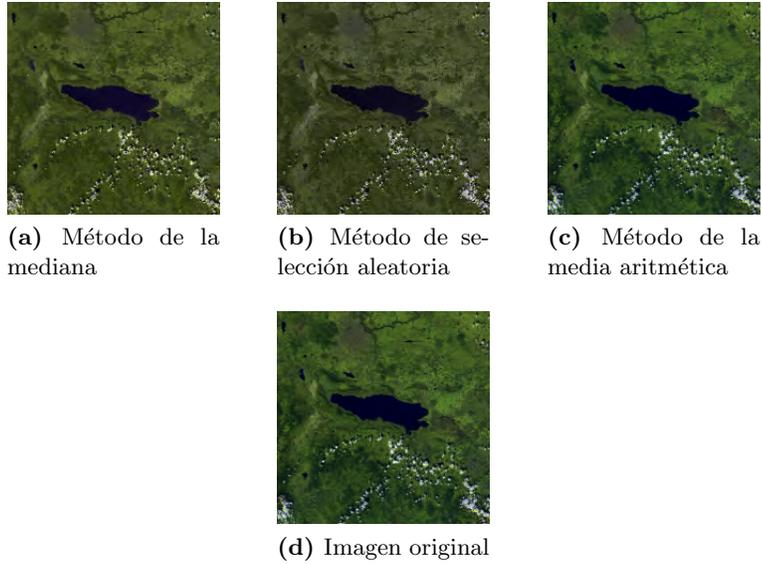
Para cada una de las pruebas siguientes muestro la subimagen creada a partir de los datos originales, tres subimágenes creadas a partir de la predicción de los datos originales (reconstrucción) y tres subimágenes (cuatro veces más grandes que las anteriores) creadas a partir de los datos mejorados; los datos de las últimas seis imágenes mencionadas los obtuve mediante la aplicación de las tres variaciones de mi algoritmo. Además de una matriz de covarianza cuyos detalles se especifican más adelante, que me ayuda a ver las relaciones entre las bandas y el grado de reconstrucción de las mismas. Con el fin de especificar más detalles del funcionamiento de mi algoritmo, en algunas pruebas presento información adicional a la antes mencionada.

## 3.2. Imagen 1

Los datos de esta sección corresponden al día 20 de enero del año 2015 y a la región del lago de Chapala, Jalisco.

En la figura 3.1 se observa la imagen original y las reconstrucciones de la misma. En la figura 3.2 se muestra la imagen original y las imágenes mejoradas o en alta resolución.

Con el fin de mostrar las relaciones que existen entre las bandas y que el proceso de normalización no afecta en la matriz de covarianza, para esta sección en particular se muestran la matrices de covarianza entre las bandas  $m$ 's originales ( $m$ ) y la reconstrucción de las mismas ( $m'$ ). Para las secciones siguiente se mostrará únicamente la matriz de covarianza correspondiente a la variación del algoritmo con la que mejores resultados obtuve.



**Figura 3.1:** Reconstrucción de la imagen 1 haciendo uso de diferentes métodos de reducción de dimensión.

	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	0.80432	<b>0.81302</b>	0.58870	0.64569	0.71148
$m_4$	0.81302	<b>0.82193</b>	0.60473	0.66140	0.72623
$m_8$	0.58870	0.60473	0.88047	<b>0.88267</b>	0.85423
$m_{10}$	0.64569	0.66140	0.88267	<b>0.89225</b>	0.87521
$m_{11}$	0.71148	0.72623	0.85423	0.87521	<b>0.87657</b>

**Tabla 3.1:** Método de reducción por selección aleatoria antes de la normalización.

	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	0.80432	<b>0.81302</b>	0.58870	0.64569	0.71148
$m_4$	0.81302	<b>0.82193</b>	0.60473	0.66140	0.72623
$m_8$	0.58870	0.60473	0.88047	<b>0.88267</b>	0.85423
$m_{10}$	0.64569	0.66140	0.88267	<b>0.89225</b>	0.87521
$m_{11}$	0.71148	0.72623	0.85423	0.87521	<b>0.87657</b>

**Tabla 3.2:** Método de reducción por selección aleatoria después de la normalización.



(a) Método de la mediana



(b) Método de selección aleatoria



(c) Método de la media aritmética



(d) Imagen original

**Figura 3.2:** Alta resolución de la imagen 1 haciendo uso de diferentes métodos de reducción de dimensión

	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	0.91789	<b>0.92292</b>	0.50120	60635	0.71608
$m_4$	0.92292	<b>0.92849</b>	0.52282	0.626	0.73374
$m_8$	0.50120	0.52282	<b>0.94552</b>	0.92893	0.80784
$m_{10}$	0.6063	0.62647	0.92893	<b>0.93252</b>	0.90676
$m_{11}$	0.71609	0.73374	0.87849	0.90676	<b>0.91218</b>

**Tabla 3.3:** Método de reducción por mediana antes de la normalización.

	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	0.91789	<b>0.92292</b>	0.50120	60635	0.71608
$m_4$	0.92292	<b>0.92849</b>	0.52282	0.62647	0.73374
$m_8$	0.50120	0.52282	<b>0.94552</b>	0.92893	0.80784
$m_{10}$	0.60635	0.62647	0.92893	<b>0.93252</b>	0.90676
$m_{11}$	0.71609	0.73374	0.87849	0.90676	<b>0.91218</b>

**Tabla 3.4:** Método de reducción por mediana después de la normalización.

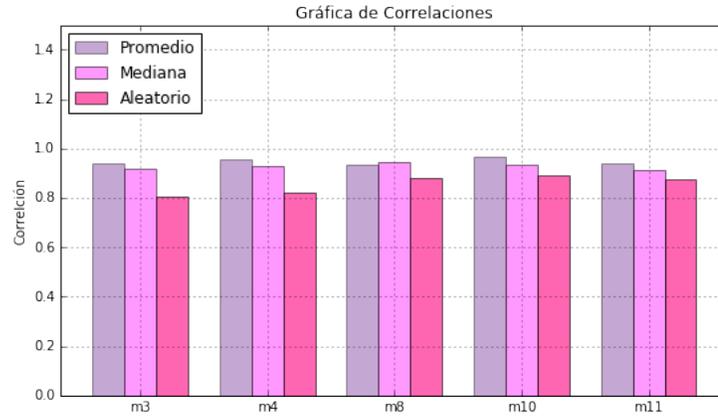
	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	0.94078	<b>0.94741</b>	0.49838	0.61300	0.71766
$m_4$	0.94741	<b>0.95444</b>	0.51853	0.63254	0.73547
$m_8$	0.49838	0.51853	0.93513	<b>0.94124</b>	0.89500
$m_{10}$	0.61300	0.63254	0.94124	<b>0.96552</b>	0.94050
$m_{11}$	0.71766	0.73547	0.89500	0.94050	<b>0.94237</b>

**Tabla 3.5:** Método de reducción por media aritmética antes de la normalización.

	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	0.94078	<b>0.94741</b>	0.49838	0.61300	0.71766
$m_4$	0.94741	<b>0.95444</b>	0.51853	0.63254	0.73547
$m_8$	0.49838	0.51853	0.93513	<b>0.94124</b>	0.89500
$m_{10}$	0.61300	0.63254	0.94124	<b>0.96552</b>	0.94050
$m_{11}$	0.71766	0.73547	0.89500	0.94050	<b>0.94237</b>

**Tabla 3.6:** Método de reducción por media aritmética después de la normalización.

En la figura 3.3 se muestra la gráfica de barras construida a partir de los valores correspondientes a las diagonales de cada una de las matrices de covarianza 3.3, 3.5 y 3.1.

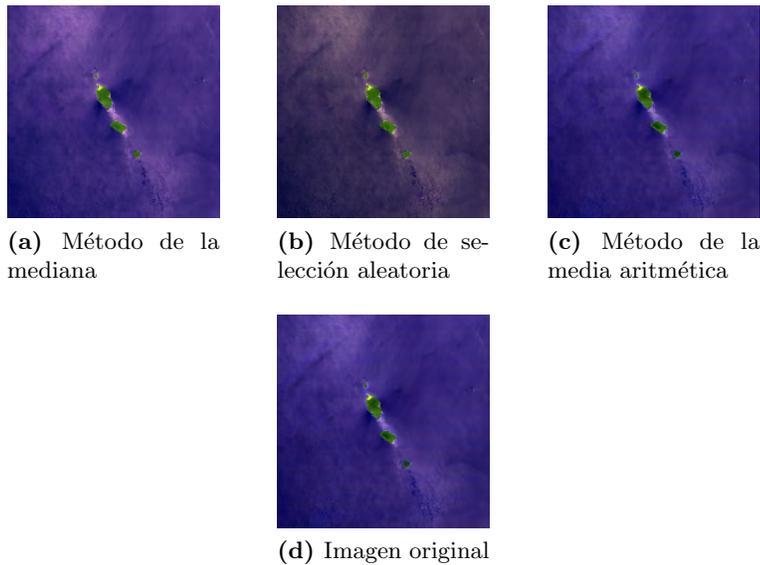


**Figura 3.3:** Gráfica de la imagen 1.

De la gráfica 3.3 se puede decir que la variación de mi algoritmo con la cual se obtuvo una mejor reconstrucción es la de la reducción de dimensión por media aritmética (promedio), siendo la variación de reducción por selección aleatoria la que realiza la reconstrucción menos favorable. Para este caso en particular, la banda  $m_{10}$  es la mejor reconstruida y la banda  $m_3$  es la que no se reconstruyó tan favorablemente como las otras. De hecho, si se observan las matrices 3.3, 3.5 y 3.1 se puede ver que la banda  $m_3$  es la mejor reconstruida si se usa  $m'_4$ .

### 3.3. Imagen 2

Los datos de esta sección corresponden al día 30 de mayo del año 2016 y a la región de las Islas Marías, Nayarit. En la figura 3.4 se observa la imagen original y las reconstrucciones de la misma. En la figura 3.5 se muestra la imagen original y las imágenes mejoradas a partir de mi algoritmo.

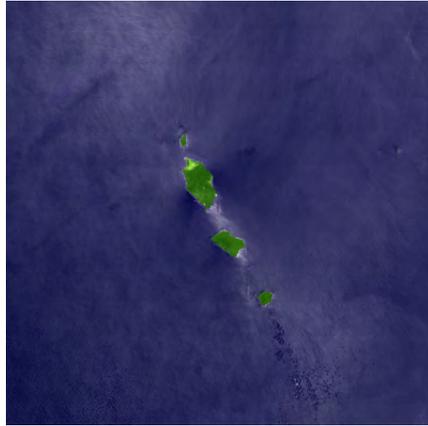


**Figura 3.4:** Reconstrucción de la imagen 2 haciendo uso de diferentes métodos de reducción de dimensión.

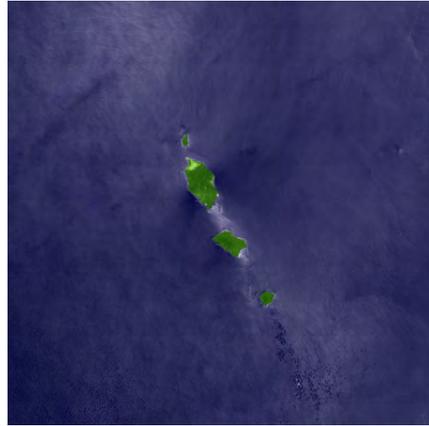
	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	<b>0.98985</b>	0.98644	0.36839	0.56304	0.82279
$m_4$	0.98644	<b>0.99722</b>	0.47772	0.65913	0.88602
$m_8$	0.36839	0.47772	<b>0.99834</b>	0.97258	0.82075
$m_{10}$	0.56304	0.65913	0.97258	<b>0.99632</b>	0.92391
$m_{11}$	0.82279	0.88602	0.82075	0.92391	<b>0.99132</b>

**Tabla 3.7:** Método de reducción por promedio para imagen 2.

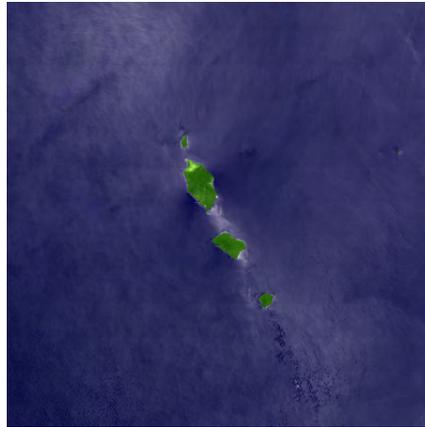
La gráfica correspondiente a esta sección se puede observar en la figura 3.6. De la gráfica 3.6 se puede decir que la variación de mi algoritmo con la cual se obtuvo una mejor reconstrucción es la de la reducción de escala por media aritmética, siendo la variación de reducción por selección aleatoria la



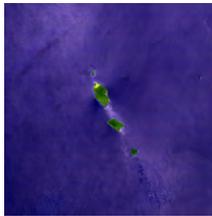
(a) Método de la mediana



(b) Método de selección aleatoria



(c) Método de la media aritmética



(d) Imagen original

**Figura 3.5:** Alta resolución de la imagen 2 haciendo uso de diferentes métodos de reducción de dimensión.

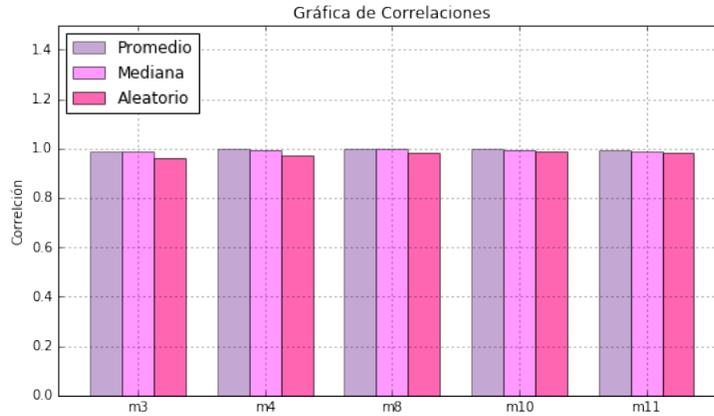


Figura 3.6: Gráfica de la imagen 2.

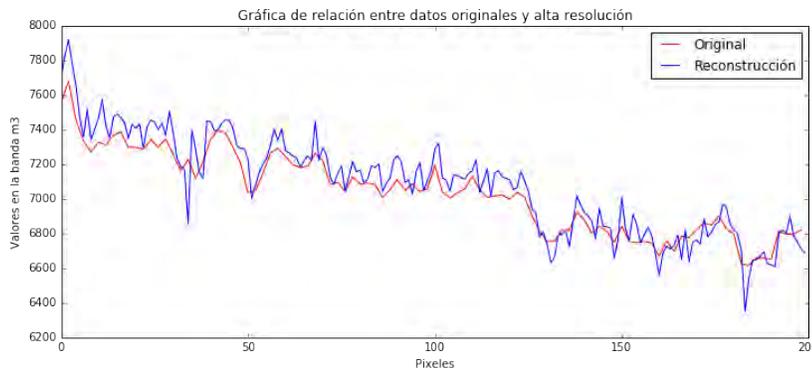


Figura 3.7: Originas vs Alta resolución.

que realiza una reconstrucción no tan favorable. Para este caso en particular, la banda mejor reconstruida es la  $m_8$  y la que no fue reconstruida de una manera favorable es la  $m_3$ . De la matriz 3.7 se puede ver que la banda  $m_8$  es la mejor reconstruida si usamos  $m'_8$ .

Con la finalidad de mostrar la relación que existe entre los valores originales de una banda y los valores en alta resolución (creados a partir de nuestro algoritmo) de la misma, se presenta la figura 3.7, en la cual la idea principal es que el lector pueda observar como es que existe una fuerte relación entre los valores originales y los valores en alta resolución de una banda, presentándose mayor detalle en estos últimos. Para su construcción se eligieron 200 píxeles al azar de esta subimagen, tomando en cuenta sus valores

en la banda  $m_3$ .

### 3.4. Imagen 3

Los datos de esta sección corresponden al día 28 de abril del año 2016 y a la región de Puerto Vallarta, Jalisco. En la figura 3.8 se observa la imagen original y las reconstrucciones de la misma. En la figura 3.9 se muestra la imagen original y las imágenes mejoradas a partir de mi algoritmo.



(a) Método de la mediana



(b) Método de selección aleatoria



(c) Método de la media aritmética



(d) Imagen original.

**Figura 3.8:** Reconstrucción de la imagen 3 haciendo uso de diferentes métodos de reducción de dimensión.

	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	<b>0.93641</b>	0.92199	-0.64753	-0.26071	0.14937
$m_4$	0.92199	<b>0.93129</b>	-0.52407	-0.11272	0.29379
$m_8$	-0.64753	-0.52407	<b>0.99521</b>	0.875469	0.60533
$m_{10}$	-0.26071	-0.11272	0.87546	<b>0.95446</b>	0.85713
$m_{11}$	0.14937	0.29379	0.60533	0.85713	<b>0.93966</b>

**Tabla 3.8:** Método de reducción por promedio para imagen 3.



(a) Método de la mediana



(b) Método de selección aleatoria

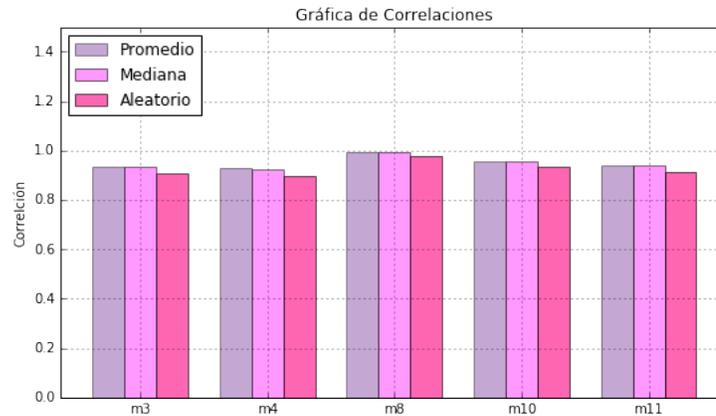


(c) Método de la media aritmética



(d) Imagen original

**Figura 3.9:** Alta resolución de la imagen 3 haciendo uso de diferentes métodos de reducción de dimensión.



**Figura 3.10:** Gráfica de la imagen 3.

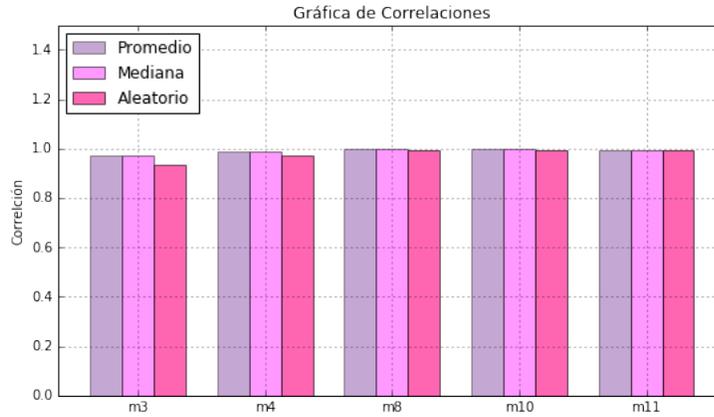
La figura 3.10 nos muestra la gráfica de barras correspondiente a esta imagen. De la gráfica 3.10 se puede decir que la variación de mi algoritmo con la cual se obtuvo una mejor reconstrucción es la de la reducción de escala por media aritmética, siendo la variación de reducción por selección aleatoria la que realiza la reconstrucción menos favorable. Para esta subfigura, la banda mejor reconstruida es la banda  $m_8$  y la que no fue tan bien reconstruida como las otras es la banda  $m_4$ . De la matriz 3.8 se puede ver que la banda  $m_8$  es la mejor reconstruida si usamos  $m'_8$ .

### 3.5. Imagen 4

Los datos de esta sección corresponden al día 3 de mayo del año 2016 y a la región de la Isla Ángel de la Guarda, Baja California y la Isla Tiburón, Sonora.

En la figura 3.13 se observa la imagen original y las reconstrucciones de la misma. En la figura 3.12 se muestra la imagen original y las imágenes mejoradas a partir de mi algoritmo.

La figura 3.11 nos muestra la gráfica de barras correspondiente a esta imagen. De la gráfica 3.11 se puede decir que la variación de nuestro algoritmo con la cual se obtuvo una mejor reconstrucción es la de la reducción de escala



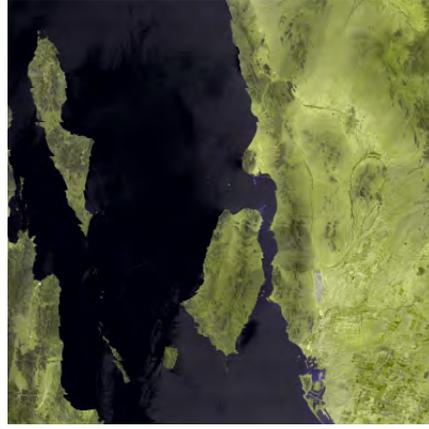
**Figura 3.11:** Gráfica de la imagen 4.

	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	<b>0.97210</b>	0.96834	0.85249	0.85938	0.87691
$m_4$	0.96834	<b>0.98915</b>	0.92782	0.93247	0.94405
$m_8$	0.85249	0.92782	<b>0.99921</b>	0.99818	0.99469
$m_{10}$	0.85938	0.93247	<b>0.99818</b>	0.99739	0.99455
$m_{11}$	0.87691	0.94405	<b>0.99469</b>	0.99455	0.99345

**Tabla 3.9:** Método de reducción por promedio para imagen 4.



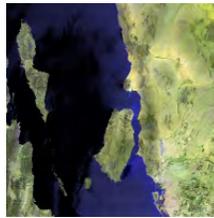
(a) Método de la mediana



(b) Método de selección aleatoria

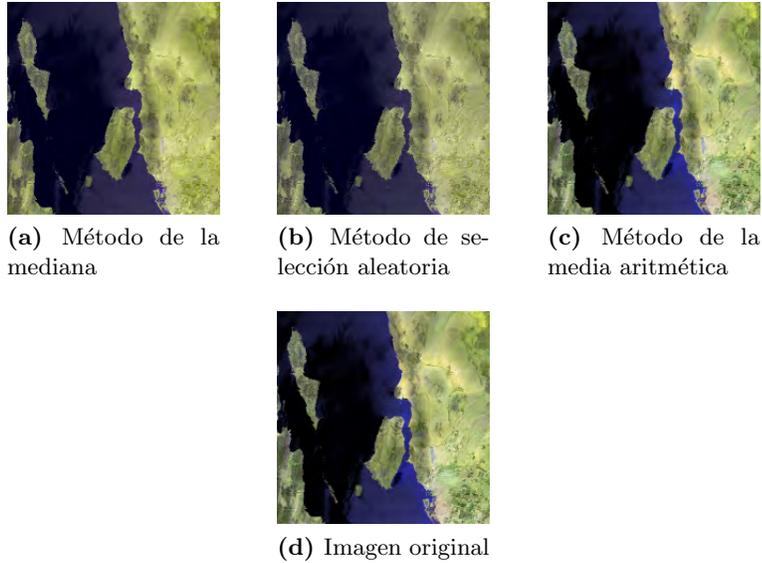


(c) Método de la media aritmética



(d) Imagen original.

**Figura 3.12:** Alta resolución de la imagen 4 haciendo uso de diferentes métodos de reducción de dimensión.



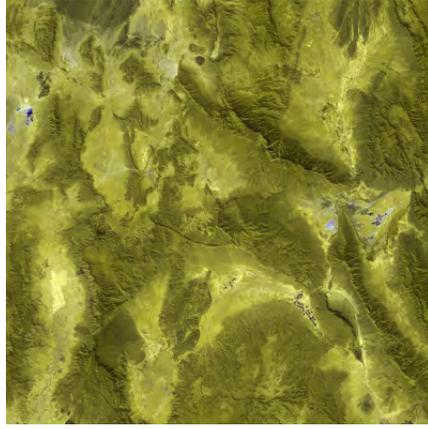
**Figura 3.13:** Reconstrucción de la imagen 4 haciendo uso de diferentes métodos de reducción de dimensión.

por promedio, siendo la variación de reducción por selección aleatoria la que realiza la reconstrucción menos favorable. Para este caso se puede decir que la banda mejor reconstruida es la banda  $m_8$  y la que no fue tan bien reconstruida como las otras es la  $m_3$ . De la matriz 3.9 se puede ver que la banda  $m_8$  es la mejor reconstruida si usamos  $m'_8$ .

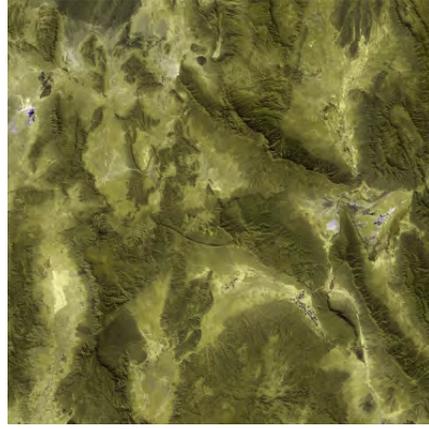
### 3.6. Imagen 5

Los datos de esta sección corresponden al día 19 de febrero del año 2017 y a la región de la Reserva de la Biosfera Mipimi, Durango. En la figura 3.15 se observa la imagen original y las reconstrucciones de la misma haciendo. En la figura 3.9 se muestra la imagen original y las imágenes mejoradas a partir de mi algoritmo. La figura 3.16 nos muestra la gráfica de barras correspondiente a esta imagen.

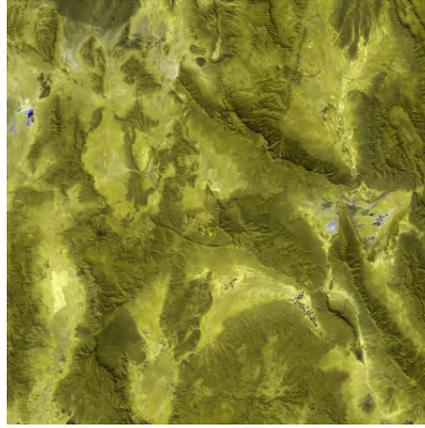
De la gráfica 3.16 podemos decir que la variación de mi algoritmo con la cual se obtuvo una mejor reconstrucción es la de la reducción de escala



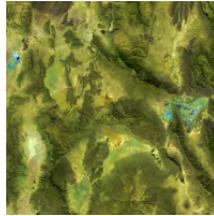
(a) Método de la mediana



(b) Método de selección aleatoria

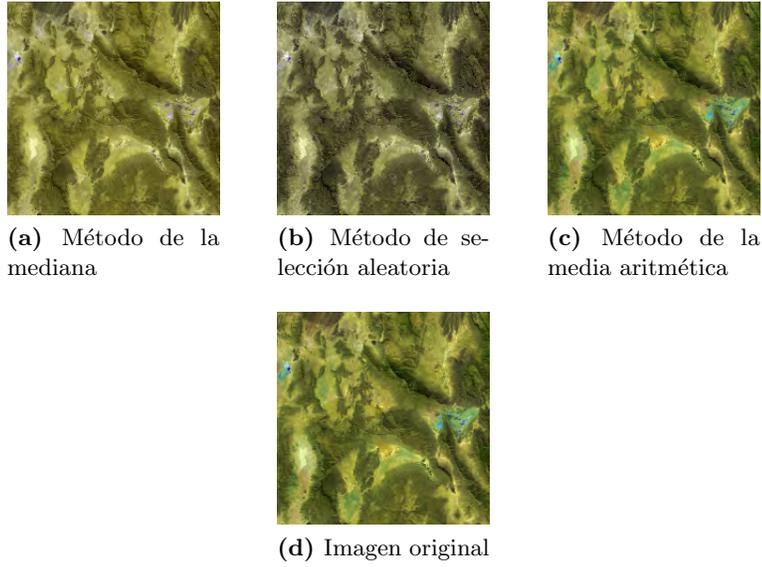


(c) Método de la media aritmética

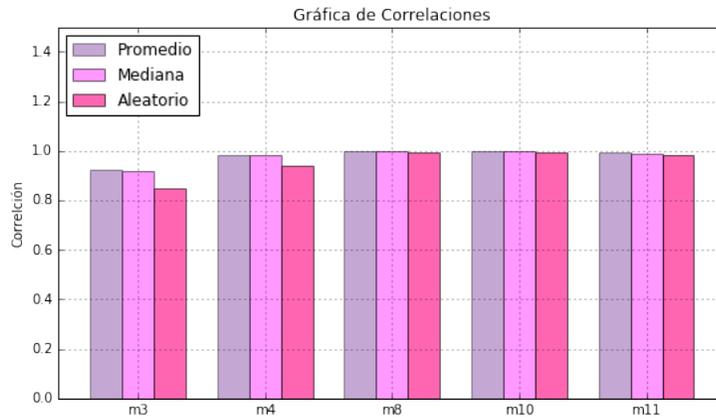


(d) Imagen original.

**Figura 3.14:** Alta resolución de la imagen 5 haciendo uso de diferentes métodos de reducción de dimensión.



**Figura 3.15:** Reconstrucción de la imagen 5 haciendo uso de diferentes métodos de reducción de dimensión.



**Figura 3.16:** Gráfica de la imagen 5.

	$m'_3$	$m'_4$	$m'_8$	$m'_{10}$	$m'_{11}$
$m_3$	0.97423	<b>0.98124</b>	0.93399	0.89033	0.87330
$m_4$	0.98124	<b>0.98832</b>	0.94298	0.89988	0.88286
$m_8$	0.93399	0.94298	<b>0.99331</b>	0.98851	0.97431
$m_{10}$	0.89033	0.89988	0.98851	<b>0.99968</b>	0.98705
$m_{11}$	0.87330	0.88286	0.97431	<b>0.98705</b>	0.97460

**Tabla 3.10:** Método de reducción por promedio para imagen 5.

por promedio, siendo la variación de reducción por selección aleatoria la que realiza la reconstrucción menos favorable; la banda  $m_{10}$  es la mejor reconstruida y la banda  $m_3$  es la que no reconstruida de una manera tan favorable en comparación de las otras. De la matriz 3.10 se puede ver que la banda  $m_{10}$  es la mejor reconstruida si usamos  $m'_{10}$ .

# Conclusiones y trabajo futuro

Como mencionamos al principio del presente trabajo, el objetivo principal era mostrar una alternativa a los algoritmos de fusión de imágenes satelitales que existen en la actualidad. Consideramos que dicho objetivo se ha conseguido, obteniendo las siguientes conclusiones:

- El algoritmo presente es el resultado de una serie de pruebas y variaciones que toman como base el algoritmo de mínimos cuadrados parciales, siendo el que se reporta el que mejor resultados nos otorga.
- De las variaciones mencionadas para reducción de escala, la que mejor trabaja es la de reducción de escala por media aritmética.
- Con el algoritmo propuesto se obtiene resultados con correlaciones entre las imágenes procesadas hasta de un .99.
- A pesar de los problemas que se presentaron al no contar con 5 bandas predictoras, se logró adaptar el algoritmo al número de bandas disponibles.
- Obtuvimos una mejora con respecto al algoritmo CIMAT-CONABIO de acuerdo a las correlaciones entre las imágenes, ya que dicho algoritmo alcanzaba un .98 y el nuestro alcanza una correlación hasta de

.99. Cabe destacar que el algoritmo CIMAT-CONABIO fue evaluado para imágenes MODIS.

## Trabajo Futuro

- Comparar el algoritmo propuesto con los algoritmos ya existentes de fusión de imágenes.
- Mejorar la etapa de normalización de tal forma que sea variable espacialmente, aunque suave.
- Aplicar el algoritmo propuesto a otro tipo de imágenes, por ejemplo imágenes Landsat.
- Aplicar el algoritmo propuesto específicamente a imágenes MODIS, ya que se sabe que estas imágenes si cuentan con cinco bandas predictoras; para ello se modificarían los procesos normalización y de reducción de escala, ya que para este tipo de imágenes ya están bien definidos.
- Mejorar la eficiencia de nuestro algoritmo programándolo en un lenguaje de más bajo nivel que Python.
- Procesar imágenes completas y no sólo subimágenes.
- Reportar los resultados obtenidos en un artículo científico.

# Apéndices



## .1. Mínimos Cuadrados

```
#Programa para ajustar un polinomio de grado 2
#Haciendo uso de minimos cuadrados

import numpy as np
import random as ran
import numpy.linalg

x = np.zeros(20)
y = np.zeros(20)
for i in range(0,10) : #generamos X
    x[i+10] = i
    x[i] = i-10
for i in range(0,20) :
    y[i] = pow(x[i],2) + ran.randint(1,10) #generamos Y

A = np.zeros((20,3))
for i in range(0,20) : #generamos A
    A[i,0] = 1
    A[i,1] = x[i]
    A[i,2] = pow(x[i],2)

#Calculamos el vector solucion
mc = np.dot(numpy.linalg.inv(np.dot(A.T,A)),np.dot(A.T,y))

#Calculamos los nuevos valores de Y a partir del polinomio
ajustado
ny = np.zeros(20)
for i in range(0,20):
    ny[i] = mc[0] + mc[1]*x[i] + mc[2]*pow(x[i],2)
```

## .2. Análisis de Componentes Principales

```
#Programa para realizar PCA sobre un conjunto de datos X
import numpy as np

x = base_de_datos

#Centrando a media 0 y desviacion estandar 1
x = x - np.mean(x, axis = 0)
x = x / np.std(x, axis = 0)

#Calculando SVD de X
u,delta , vt = np.linalg.svd(x)

#Numero de componentes principales
cp = numero de componentes deseado

#Tomando el numero de componentes principales correspondiente
nvt [:cp, :]

#Calculando la matriz de componentes principales
Y = x * nvt.T

#Reconstruyendo x
xcal = Y * nvt
```

### .3. Base de datos “Places Rated”

A continuación se exponen los detalles de la base de datos “Places Rated data”, la cual puede ser fácilmente consultada en línea y para este trabajo se obtuvo de la referencia [2].

Esta base de datos, como su nombre lo dice, fue creada para clasificar lugares, en específico, ciudades de Estados Unidos, la base de datos cuenta con nueve atributos (variables), todos numéricos y son los siguientes:

- Clima y terreno (CT).
- Viviendas (V).
- Salud y medio ambiente (SMA).
- Crimen (C).
- Transporte (T).
- Educación (E).
- Arte (A).
- Recreación (R).
- Economía (EC).

La base de datos cuenta con 329 muestras o datos, pero para fines de este trabajo mostramos a continuación los veinticinco primeros únicamente:

CT	V	SMA	C	T	E	A	R	EC
521	6200	237	923	4031	2757	996	1405	7633
575	8138	1656	886	4883	2438	5564	2632	4350
468	7339	618	970	2531	2560	237	859	5250
476	7908	1431	610	6883	3399	4655	1617	5864
659	8393	1853	1483	6558	3026	4496	2612	5727

520	5819	640	727	2444	2972	334	1018	5254
559	8288	621	514	2881	3144	2333	1117	5097
537	6487	965	706	4975	2945	1487	1280	5795
561	6191	432	399	4246	2778	256	1210	4230
609	6546	669	1073	4902	2852	1235	1109	6241
885	16047	2025	983	3954	2843	5632	3156	6220
195	12175	601	1223	5091	2414	2346	3000	7668
530	5704	580	878	2865	2469	430	838	3370
591	5725	820	975	2707	2772	169	613	4262
546	11014	2508	1067	3433	3346	7559	2288	4579
560	5530	598	1125	3051	2189	268	1165	4730
396	7877	833	525	3298	2844	1166	2315	5275
694	6722	1204	566	5086	2990	1391	1542	5196
601	6691	605	933	1866	2646	3546	1001	5193
696	8316	3195	1308	8409	3057	7559	1362	6315
615	11074	637	1878	3556	2929	621	2711	8107
534	6292	1798	872	2523	2915	1047	913	5431
474	10384	1203	821	3943	2208	3857	1800	5097
435	8831	782	1049	3670	3063	5355	1063	7439
560	8068	420	1561	3725	2564	1222	1568	6056

**.4. NIPALS**

```

#Algoritmo NIPALS

import numpy as np
import math

data = np.loadtxt('base_de_datos')

nd = numero de datos
ni = numero de variables que desea conservar
nvx = numero de variables de X
nvy = numero de variables de Y

#Construimos X y Y de acuerdo a nuestro datos
X = data[:,0:nvx]
Y = data[:,nvx:nvx+nvy]

#Para no perder los valores originales creamos copias
xc = np.copy(X)
yc = np.copy(Y)

#centramos X a media 0
xprom = np.matrix(np.mean(xc,1))
xprom = xprom.T
xc = xc - xprom

#centramos Y a media 0
yprom = np.matrix(np.mean(yc,1))
yprom = yprom.T
yc = yc - xprom

u = np.matrix(Y[:,0])
u = u.T
U = np.zeros((nd,1))
T = np.zeros((nd,1))
P = np.zeros((nvx,1))
Q = np.zeros((nvy,1))

```

```

for i in range(0, ni):
    tant = np.matrix(Y[:,1])
    tant = tant.T
    t = np.matrix(Y[:,0])
    t = t.T
    while np.linalg.norm(tant - t) > 0.000001:
        tant = t
        temp = x.T*u
        p = temp / (math.sqrt(temp.T * temp))
        t = x * p
        temp2 = y.T*t
        q = temp2 / (math.sqrt(temp2.T * temp2))
        u = y*q

    U = np.concatenate((U, u), axis=1)
    T = np.concatenate((T, t), axis=1)
    P = np.concatenate((P, p), axis=1)
    Q = np.concatenate((Q, q), axis=1)
    x = x - t*p.T
    y = y - u*q.T

T = T[:,1:]
U = U[:,1:]
P = P[:,1:]
Q = Q[:,1:]

beta = (U/T).mean()

#Reconstruimos Y
nuevay = xc*(P*Q.T)

```

# Bibliografía

- [1] *Comisión nacional para el conocimiento y uso de la biodiversidad*, <https://www.gob.mx/conabio>, 2016.
- [2] Richard Boyer, David Savageau, and Rand McNally, *Places rated data*, <http://lib.stat.cmu.edu/datasets/places.data>, 2017.
- [3] Richard L. Burden and J.Douglas Faires, *Numerical analysis*, ninth ed., Amana images, Richard Stratton, 2011.
- [4] Campbell, J.B, and R.H Wynne, *Introduction to remote sensing*, fifth ed., The Guilford Press, 2011.
- [5] Jeffrey R. Chasnov, *Introduction to numerical methods*, The Hong Kong University of Science and Technology, 2012.
- [6] Ildiko E.Frank and Jerome H. Friedman, *A statistical view of some chemometricsa regression tools (with discussion)*, 1993.
- [7] C. Frey, C. Kuenzer, and S. Dech, *Quantitative comparison of the operational noaa avhrr lst product of dlr and the modis lst product v005*.
- [8] Golub Gene H. and Charles F Van Loan, *Matrix computations*, third ed., The Johns Hopkins University Press, 1996.
- [9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning*, Springer, 2001.

- [10] Friedman Jerome, Trevor Hastie, and Robert Tibshirani, *The elements of statistical learning*, Springer, Berlin: Springer series in statistics.
- [11] Jones, H.G., and R.A. Vaughan, *Remote sensing of vegetation : Principles, techniques, and applications*, USA, Oxford University Press, 2010.
- [12] David Lay, *Linear algebra and it's applications*, second ed., 2000.
- [13] Lillesand, T., Kiefer, R.W, and J. Chipman, *Remote sensing and image interpretation*, seventh ed., Wiley, 2015.
- [14] L. LU, C. Kuenzer, C. Wang, H. Gio, and Q. Li, *Evaluation of th-ree modis-derived vegetation index time series for dry land vegetation dynamics monitoring. remote sensing*, 2015.
- [15] Jean-François Mas, *Aplicaciones del sensor modis para el monitoreo del territorio*, first ed., Secretaría de Medio Ambiente y Recursos Naturales, 2011.
- [16] Horning N., Robinson, J.A, Sterling, E.J, Turner, W., and Spector S., *Remote sensing for ecology and conservation: A handbook of techniques*, USA, Oxford University Press, 2010.
- [17] Carlos Gaviria Peña, *Regresión por mínimos cuadrados parciales pls aplicada a datos variedad valuados*, Universidad Nacional de Colombia, Facultad de Ciencias, Escuela de Estadística, Medellín, Colombia, 2016.
- [18] Tenenhaus, M., Esposito, V., Chatelic, Y-M., and Lauro C., *Pls plath modeling. computational statistics and sata analysis*, 2005.
- [19] Alexander P. Trishchenko, Yi Luo, and Konstantin V. Khlopenkova, *A method for downscaling modis land channels to 250 spatial resolution using adaptive regression and normalization*, Canada Center for Remote Sensing, Earth Sciences Sector (2006).
- [20] Vinzi, Chin, W.W, and Henseler J., *Handbook of partial least squares; concepts, methods and applications*, Springer-Verlag Berlin Heidelberg, 2010.

- [21] Todd Will, *Introduction to the singular value decomposition*, Davison Collage (1999).
- [22] Mariano Rivera y Judith Esquivel, *Fusión y análisis de imágenes satelitales de sensores modis y viirs*, Centro de Investigación en Matemáticas A.C (2015).