

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO

---



OPTIMIZACIÓN DEL COCIENTE DE LA TRAZA PARA  
MÁQUINAS DE APRENDIZAJE

TESIS

QUE PARA OBTENER EL TÍTULO DE  
LICENCIADO EN ACTUARÍA

PRESENTA

SALVADOR GARCÍA GONZÁLEZ

ASESOR

DR. ZEFERINO PARADA GARCÍA

MÉXICO, D.F.

2016

“Con fundamento en los artículos 21 y 27 de la Ley Federal de Derecho de Autor y como titular de los derechos moral y patrimonial de la obra titulada **“OPTIMIZACIÓN DEL COCIENTE DE LA TRAZA PARA MÁQUINAS DE APRENDIZAJE”**, otorgo de manera gratuita y permanente al Instituto Tecnológico Autónomo de México y a la biblioteca Raúl Baillères Jr., autorización para que fijen la obra en cualquier medio, incluido el electrónico, y la divulguen entre sus usuarios, profesores, estudiantes o terceras personas, sin que pueda percibir por tal divulgación una contraprestación”

SALVADOR GARCÍA GONZÁLEZ

---

Fecha

---

Firma

# Índice general

<b>Prólogo</b>	<b>I</b>
<b>1. Discriminante Lineal de Fisher</b>	<b>5</b>
1.1. Aprendizaje de máquina . . . . .	6
1.1.1. Procesos de aprendizaje . . . . .	6
1.2. Matrices de dispersión . . . . .	8
1.3. Problema del cociente de trazas . . . . .	12
1.3.1. Solución cuando $p = 1$ . . . . .	13
1.3.2. Generalización a $p$ dimensiones . . . . .	16
1.3.3. Existencia de la solución . . . . .	19
1.3.4. Equivalencia con un problema escalar . . . . .	22
1.3.5. Localización del óptimo . . . . .	29
<b>2. El Método Newton-Lanczos</b>	<b>37</b>
2.1. Métodos de Lanczos . . . . .	38
2.1.1. Algoritmo de Lanczos . . . . .	39
2.2. Derivada de $f(\rho)$ . . . . .	41
2.3. Método Newton-Lanczos . . . . .	44
2.3.1. Condiciones necesarias de optimalidad . . . . .	45
<b>3. Experimentos numéricos</b>	<b>49</b>

## ÍNDICE GENERAL

3.1. Comparación de tiempo entre el método de Lanczos y la factorización SVD . . . . .	51
3.2. Modelos comparados y preprocesamiento de las bases . . . . .	53
3.3. Base de datos State Farm . . . . .	54
3.3.1. Proyección sobre un espacio de dimensión 20 . . . . .	56
3.3.2. Comparación con otros métodos . . . . .	59
3.4. Base de datos Otto Group . . . . .	62
3.4.1. Proyección sobre un espacio de dimensión 20 . . . . .	63
3.4.2. Comparación con otros métodos . . . . .	66
<b>4. Conclusiones</b>	<b>68</b>
<b>A. Apéndice A: Optimización de <math>Tr(V^T AV)</math></b>	<b>70</b>
A.1. Problema relajado . . . . .	70
A.2. Observaciones finales . . . . .	74
<b>Bibliografía</b>	<b>75</b>

# Índice de figuras

1.1. Enfoques para resolver problemas de clasificación. . . . .	7
1.2. Distancias en las matrices de dispersión. . . . .	10
1.3. Mejores proyecciones en $\mathbb{R}^2$ y $\mathbb{R}$ . . . . .	14
1.4. Comportamiento de $f(\rho)$ . . . . .	27
1.5. Gráfica de los eigenvalores en función de $\rho$ . . . . .	28
1.6. $f(\rho)$ para proyectores de 1,2 y 3 dimensiones. . . . .	34
1.7. Intervalos para $\rho^*$ . (Primera cota) . . . . .	35
1.8. Intervalos para $\rho^*$ . (Segunda cota) . . . . .	36
3.1. Desempeño de Lanczos . . . . .	52
3.2. Ejemplo de fotografías de la base de datos (State Farm). . . . .	55
3.3. Valores de $\rho$ y $f(\rho)$ para distintas iteraciones (State Farm). . . . .	57
3.4. Ejemplo de proyección en 20 dimensiones (State Farm) . . . . .	58
3.5. Tasa de reconocimiento de los 3 métodos (State Farm) . . . . .	60
3.6. Tiempo de cómputo (State Farm) . . . . .	61
3.7. Ejemplo de la base de datos (Otto Group) . . . . .	62
3.8. Valores de $\rho$ y $f(\rho)$ para las iteraciones (Otto Group). . . . .	64
3.9. Ejemplo de proyección en 20 dimensiones (Otto Group) . . . . .	65
3.10. Tasa de reconocimiento de los 3 métodos (Otto Group) . . . . .	66
3.11. Tiempo de cómputo (Otto Group) . . . . .	67

*A mis padres, Michelle y Alexis,  
quienes siempre me brindan su apoyo incondicional.*

# Agradecimientos

Quiero agradecer en primer lugar a mis padres, por todo el esfuerzo que realizan día a día para que salgamos adelante como una familia unida. Les agradezco por siempre comprenderme y por darme buenos consejos, sin los cuales, no hubiera podido llegar a este lugar. También a Alexis, quien me brinda los mejores momentos de compañía.

También quiero agradecer al Dr. Zeferino Parada, por que siempre me comparte sus mejores consejos y por la invaluable aportación que me brindó para que lograra este trabajo. Es una persona muy brillante y con la mejor disposición para ayudarme a realizar un buen trabajo.

Agradezco a Michelle, quien a lo largo de 5 años me ha apoyado incondicionalmente. Agradezco sus consejos y sus regaños, así como sus cálidos abrazos en momentos difíciles. Me haces crecer como persona y estoy seguro, que tu estarás a mi lado en un futuro.

Uno de las más grandes satisfacciones que me llevé de la universidad son los grandes amigos que conocí ya que cada uno de ustedes aportó algo especial a lo que soy. En especial agradezco a Fernanda por la grandiosa persona que es y por motivarme a llegar cada vez más lejos. A mis roomies y amigos desde primer semestre Alberto, Miriam, Jorge, Sofí, JM, Darío, Yedam, Gabriel, Gilberto y Fernando por haber estado estos 6 años acompañándome y compartiendo muy buenas experiencias conmigo. A David, José Luis y Alfredo por inspirarme con su ejemplo diario. A los MVs: Patiño, Kike, Hugo, Ernesto, Pepe, Rico, Magos, Esteban, Chucho, Roomie y Juan por compartir el amor a las matemáticas y por las jornadas de estudio que pasamos juntos.

También agradezco a mis dos más grandes ejemplos de vida profesional que

## ÍNDICE DE FIGURAS

conozco. El Dr. Guillermo Zárate y el Dr. Fabián Hernández. Les agradezco sus consejos diarios y las enseñanzas que me han brindado a lo largo de este último año y medio. También agradezco a Rodrigo y Mariana, por todo su apoyo en esta tesis, ya que sin su ayuda, hubiera sido casi imposible terminar esta tesis a tiempo.

Finalmente, y no menos importante, quiero agradecer a mis sinodales, al Dr. José Luis Morales y al Dr. José Luis Farah por haberme dedicado su tiempo y ayudarme a mejorar este trabajo.



# Prólogo

El Análisis Discriminante Lineal de Fisher (ADLF) es un método estadístico utilizado comúnmente con el fin de reducir la dimensionalidad de un conjunto de datos como paso preliminar para resolver un problema de clasificación. La reducción dimensional realizada por este método es especial, ya que su objetivo final es separar dos o más clases de individuos en un nuevo espacio. El método para dos clases fue desarrollado por Ronald A. Fisher (1936), aunque después fue generalizado por C. R. Rao (1948).

El ADLF es empleado como un método de aprendizaje de máquina, en particular en problemas de reconocimiento de patrones. Puede ser aplicado a diversas áreas; desde la medicina, en reconocimiento de tumores cancerígenos, hasta en finanzas, para la identificación de individuos con alta probabilidad de caer en incumplimiento de pago.

El planteamiento del problema de optimización del ADLF consiste en lo siguiente: dado un conjunto de individuos que tienen una clase asignada, se busca encontrar la mejor matriz  $V^{**}$  que proyecte a los individuos en un espacio  $k$  – *dimensional*, con  $k$  menor a la dimensión del espacio original. Al encontrar la mejor matriz se logra representar a los individuos, en un espacio de menor dimensión, de tal manera que se encuentren agrupados de acuerdo con la clase a la que pertenecen.

El ADLF encuentra la solución a dicho problema por medio de la maximización del cociente formado por la matriz de dispersión entre-clases (*Between-class scatter matrix*) y la matriz de dispersión intra-clase (*Within-class scatter matrix*), ambas de los individuos proyectados. Maximizar este cociente era considerado computacionalmente costoso, por lo que solía ser remplazado por versiones

## ÍNDICE DE FIGURAS

simplificadas.

El objetivo de este trabajo es proponer el método de Newton-Lanczos como una solución computacionalmente accesible. Este método resuelve iterativamente el problema de maximización aprovechando la velocidad de convergencia del método de Newton. Para medir el desempeño de esta propuesta se compara con la precisión y el tiempo de cómputo de otros métodos de clasificación lineal.

El primer capítulo de esta tesis profundiza en el problema de optimización planteado por el ADLF. Para esto se definen los conceptos indispensables para el desarrollo de la teoría. Luego, se demuestra que el problema de optimización es equivalente a un problema escalar, por lo que se puede expresar en términos de una función objetivo  $f(\rho)$  y un argumento unidimensional  $\rho$ . Enseguida, se dan las condiciones de existencia de la solución.

En el segundo capítulo se profundiza en el método de Newton-Lanczos, el cual resuelve el problema de encontrar la  $\rho$  óptima del problema escalar. Se proporciona una breve presentación de la teoría que sustenta a los métodos de Lanczos, su costo computacional y las ventajas que tienen sobre los métodos tradicionales. Más adelante se combina con el método de Newton, por lo que se requiere formular la derivada de  $f(\rho)$ . Al final, se proporcionan las condiciones necesarias de optimalidad.

En el tercer capítulo se presenta los experimentos numéricos realizados sobre las bases de datos de las empresas *Otto Group* y *State Farm*. Se eligen estas bases ya que cuentan con una dimensionalidad alta que resulta difícil de manejar en métodos de clasificación tradicionales. Además, se da una breve introducción del preprocesamiento y posteriormente se compara el desempeño del ADLF contra dos métodos de clasificación lineal: el Análisis Discriminante Lineal (ADL) y la Regresión Lineal Múltiple (RLM).

Para el desarrollo de esta tesis se utilizó el lenguaje de programación estadístico R y la paquetería *ProjectTemplate*, que sirve para gestionar proyectos de análisis. Para asegurar la portabilidad y reproducibilidad del proyecto se utilizó la paquetería *Pacman*. Finalmente, para los cálculos se utilizó la biblioteca de *LA-PACK* (*Fortran*) en su versión para OS X 10.11.4 (*liblapack.dylib*). Los códigos de esta tesis se encuentran para su libre distribución en:

<https://github.com/savrgg/TraceRatio.Optimization>.

## Capítulo 1

# Discriminante Lineal de Fisher

En este capítulo se hablará del Análisis Discriminante Lineal de Fisher (ADLF), el cual busca optimizar un cociente de la forma  $\text{Tr}(V^T AV) / \text{Tr}(V^T BV)$  sobre el conjunto de matrices ortogonales  $V$  con  $A, B$  matrices positivas definidas. Para resolver el ADLF, se han presentado en libros de aprendizaje estadístico y clasificación de patrones formulaciones alternas al problema original, ya que este era considerado computacionalmente muy costoso [9] [8]. Planteamientos como: maximizar la traza de la matriz de dispersión entre clases sujeto a una restricción sobre la matriz de dispersión interna, maximizar la traza del cociente de matrices; o bien, maximizar el cociente de determinantes, han sido formulaciones presentadas en distintos textos [2] [5] [7] [3]. Al final, todas estas propuestas resultan ser versiones simplificadas del problema.

En este capítulo se resolverá el problema original del ADLF a través del método de Newton-Lanczos, el cual resulta ser computacionalmente eficiente. En la primera sección del capítulo se contextualizará al problema dentro del área de Aprendizaje de Máquina. En la segunda parte, se proporcionará la teoría correspondiente para seguir con facilidad el texto. Por último, en la tercera parte, se plantea la solución al problema de ADLF para una dimensión, se generaliza a  $p$  dimensiones y se proporcionan las condiciones para la existencia de la solución.

## 1.1. Aprendizaje de máquina

El Aprendizaje de Máquina toma como base dos áreas de investigación: la Ciencia de la Computación y la Estadística. De la primera, retoma las preguntas: ¿Cómo se pueden construir máquinas que resuelvan problemas? Y ¿Qué problemas son tratables o intratables? De la segunda, toma las preguntas: ¿Qué puede ser inferido de los datos? ¿Bajo que supuestos del modelo? Y ¿Con qué confiabilidad? [7]. El esfuerzo por resolver estas preguntas da como resultado una disciplina enfocada a construir teorías estadístico-computacional de los procesos de aprendizaje.

### 1.1.1. Procesos de aprendizaje

Se dice que una máquina “aprende” dada una tarea (T), una medición del rendimiento (R) y un tipo de experiencia (E) si el sistema mejora confiablemente su rendimiento (R) en la tarea (T) dada la experiencia (E) [7]. Es decir, se modela una estructura con los datos proporcionados de manera que el rendimiento en la tarea mejora conforme más información recibe. La diversidad de las tareas, así como el campo de aplicaciones es muy diverso, por ejemplo:

- *Clasificación de spam/no-spam*, en el que (E) son los correos, (T) el clasificar correctamente el *spam* y (R) el porcentaje de correos correctamente clasificados.
- *Reconocimiento/Clasificación facial*, en el que (E) son los rostros de distintas personas, (T) el reconocimiento o clasificación de los rostros y (R) el porcentaje de rostros correctamente reconocidos o clasificados.

Los procesos de aprendizaje tienen diversas aplicaciones y distintos supuestos, por lo que han surgido clasificaciones para analizarlos en conjunto. La utilizada en este texto es la propuesta por T. Hastie [5], la cual divide a los métodos en dos grupos: aprendizaje supervisado y aprendizaje no supervisado. El primero, supone la presencia de una variable de salida que actúa como guía en la construcción de la estructura. Ejemplos de éste es la regresión lineal, los árboles de decisión y las máquinas de soporte vectorial. Por otra parte, el aprendizaje no supervisado solo cuenta con la información de las variables independientes; por

ejemplo, análisis de conglomerados, reglas de asociación y algunoas algoritmos de reducción dimensional.

Después de esta primer clasificación, se subclasifica a los métodos de aprendizaje supervisado de acuerdo al tipo de variable de salida (Figura 1.1). <sup>1</sup> Cuando se trata de una variable cuantitativa recibe el nombre de regresión, mientras que en el caso de cualitativas se le llama clasificación. Por otra parte, el aprendizaje no supervisado tiene dos ramas en las que el texto hace énfasis [5]: Segmentación en el caso que se desee asignar un grupo a cada individuo, de manera que los grupos sean homogéneos entre sí; o bien, reducción dimensional cuando solamente se desea proyectar a los individuos en un espacio de menor dimensión, de manera que se cumplan características especiales en dicho subespacio.

El problema de ADLF pertenece a la rama de aprendizaje supervisado, en particular a los métodos clasificación. Alternativas para esta finalidad, y que sigan supuesto linealidad en la frontera, son la Regresión Logística, el Análisis Discriminante Lineal y las Máquinas de Soporte Vectorial.

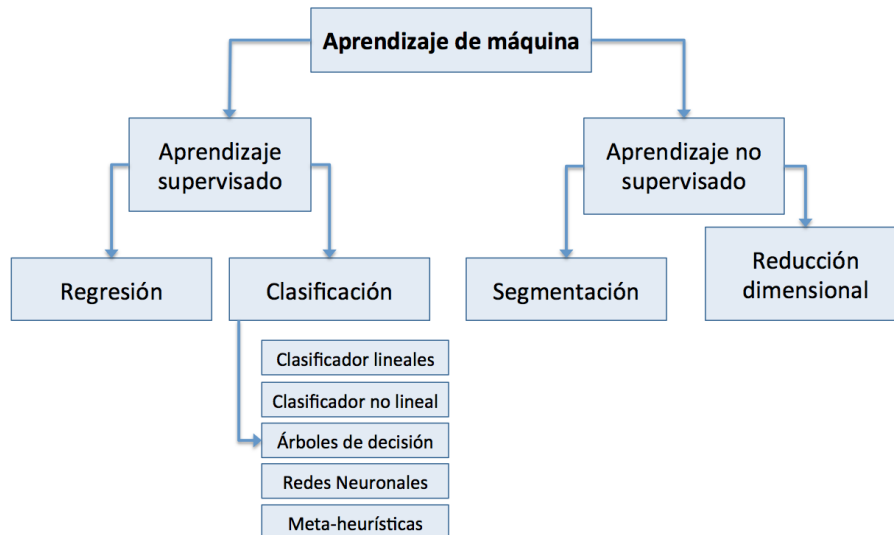


Figura 1.1: Distintos enfoques para resolver problemas de clasificación en el área de Aprendizaje de Máquina.

<sup>1</sup>A lo largo del texto se usará indiferentemente variable de entrada como “input” o variable independiente y variable de salida como “output” o variable dependiente

## 1.2. Matrices de dispersión

Se comenzará definiendo la nomenclatura necesaria para el texto. Sea  $\mathbb{C} = k; k = 1, 2, 3, \dots, K$  el conjunto de  $K$  clases distintas a las que puede ser asignado un individuo  $x_i$ , con  $i = 1, \dots, N$  y  $N$  el número total de individuos. Entonces sea  $C_k$  un subconjunto de los  $1, 2, 3, \dots, N$  individuos, de manera que el subconjunto  $C_k$  contiene a los índices de los individuos que pertenecen a la clase  $k$ . De esta manera, se define como  $N_k$  al número de individuos asignados en la clase  $k$ . Por último sea  $w_i = V^T x_i$ ; es decir, los datos multiplicados por la matriz  $V$ . Entonces, se definen las medias del grupo  $k$  como  $\mu_k$  y la media de todos los datos  $x_i$  como  $\mu$ :

$$\mu_k = \frac{1}{N_k} \sum_{i \in C_k} x_i \quad (1.1)$$

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (1.2)$$

Por otro lado, se definen las medias correspondientes a los datos proyectados  $w_i$ :

$$\widetilde{\mu}_k = \frac{1}{N_k} \sum_{i \in C_k} w_i \quad (1.3)$$

$$\widetilde{\mu} = \frac{1}{N} \sum_{i=1}^N w_i \quad (1.4)$$

El ADLF hace amplio uso de las matrices de dispersión, en específico de la matriz de covarianza, la matriz de dispersión de todos los individuos, la matriz de dispersión interna y la matriz de dispersión entre clases. Es importante analizar a profundidad la terminología y las fórmulas que se usarán a lo largo de la tesis para entender la lógica detrás de la formulación del problema de optimización.

Sea  $\Sigma$  la matriz de covarianza (*Covariance Matrix*) de todos los individuos. Se define como  $\widehat{\Sigma}$  al estimador insesgado de  $\Sigma$  el cual está escalada entre  $N - 1$ :

$$\widehat{\Sigma} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T \quad (1.5)$$

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

Si esta matriz no está escalada por  $N - 1$  entonces se le conoce como matriz de dispersión (*Scatter Matrix*). En esta tesis se representará como  $S_T$ , con el subíndice  $T$  que significa que está tomando en cuenta a todos los individuos:

$$S_T = \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T \quad (1.6)$$

Cuando solo se toman a los individuos de una clase particular  $k$ , se puede encontrar su correspondiente matriz de dispersión. Esta matriz será representada como  $S_k$ , con el subíndice  $k$  indicando que está tomando en cuenta solo a los individuos de la clase  $k$ :

$$S_k = \sum_{i \in C_k} (x_i - \mu)(x_i - \mu)^T$$

De esta misma forma se define la matriz de dispersión interna (*Within-class scatter matrix*) como la suma sobre  $k$  de todas las matrices de dispersión de cada clase:

$$S_I = \sum_{k=1}^K \sum_{i \in C_k} (x_i - \mu_k)(x_i - \mu_k)^T \quad (1.7)$$

Ahora solo falta definir la matriz de dispersión entre clases (*Between-class scatter matrix*) como la suma de diferencias al cuadrado de las medias de clase contra la media de todos los datos multiplicada por el número de individuos en cada clase  $N_k$ :

$$S_E = \sum_{k=1}^K N_k (\mu_k - \mu)(\mu_k - \mu)^T \quad (1.8)$$

Entre la matriz de dispersión interna  $S_I$ , la matriz de dispersión entre clases  $S_E$  y la matriz de dispersión total  $S_T$  existe una relación importante. Se cumple que  $S_T = S_I + S_E$ ; es decir, la dispersión de las medias de grupos con la media global más la dispersión de cada clase individual es igual a la dispersión de los datos sin la información de las clases. Los datos de la figura 1.2 representan las distancias que toman en cuenta cada una de estas matrices.

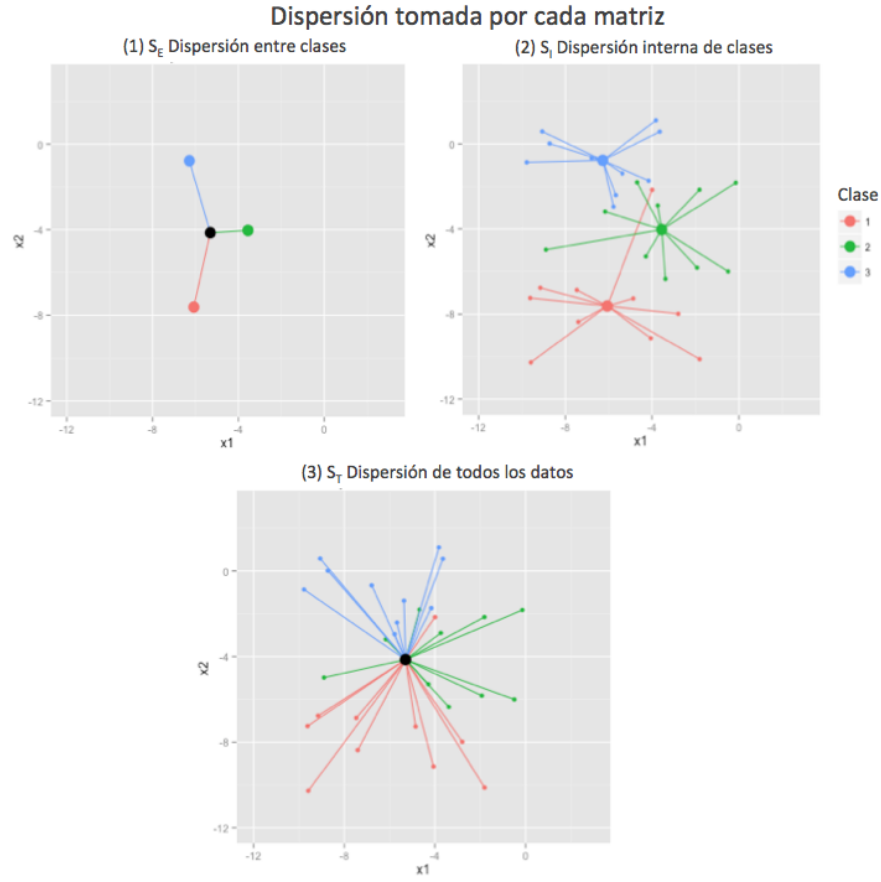


Figura 1.2: En la gráfica (1) se representa la  $S_E$ , es decir las distancias al cuadrado entre la media de todos los datos (Punto negro) y las medias de cada clase (Puntos de color gruesos). La gráfica (2) representa  $S_I$ ; es decir, la distancia al cuadrado de los individuos a la media de su clase. La gráfica (3) representa  $S_T$ , la dispersión de los datos con respecto a la media de todos.

Para ejemplificar la relación  $S_T = S_I + S_E$ , se generaron 10 datos por clase suponiendo distribuciones normales (El coeficiente de correlación de los datos generados es  $-0.005$ ). Calculando las matrices de dispersión de acuerdo a las fórmulas (1.6), (1.7) y (1.8):



## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

Clase	Distribución x1	Distribución x2
1	N(-5, 2.5)	N(-8, 2)
2	N(-3, 2.5)	N(-4, 2)
3	N(-7, 2.5)	N(-1, 2)

$S_I$	$S_E$	$S_T$
$\begin{bmatrix} 186.05 & 2.78 \\ 2.78 & 94.58 \end{bmatrix}$	$\begin{bmatrix} 46.13 & -4.15 \\ -4.15 & 234.57 \end{bmatrix}$	$\begin{bmatrix} 232.18 & -1.36 \\ -1.36 & 329.16 \end{bmatrix}$

De este ejemplo numérico se puede ver que al sumar la dispersión interna  $S_I$  y la dispersión entre clases  $S_E$  da como resultado la dispersión de todos los individuos  $S_T$ . En general este resultado se cumple.

Un problema muy común que surge en problemas de aprendizaje estadístico es que el costo computacional puede volverse intratable conforme la dimensionalidad de los individuos crece. En el ADLF se requiere hacer el cómputo de las matrices de dispersión de los individuos constantemente (o bien calcular la inversa de matrices de alta dimensionalidad), cálculos que para grandes dimensiones son sumamente costosos. Existen distintas maneras para hacer frente a este problema, uno de ellos involucra el PCA (Principal Component Analysis) en el preprocesamiento de los datos. Este método es fácil de calcular y solo requiere computar una vez la de matriz de dispersión [8]. Debido a la finalidad de esta tesis no se profundizará en más técnicas para hacer frente a este problema, pero en textos como [5], [2] aparecen distintos métodos para reducción de dimensionalidad.

Retomando el problema de cociente de trazas, lo que se busca es encontrar la proyección que mantenga juntos individuos de una clase al mismo tiempo que separa las medias de distintas clases. Una vez obtenida esta proyección se puede encontrar un hiperplano separador de los datos, o bien algún criterio para asignar la clase de pertenencia.

El problema de optimización se plantea como:

$$\max_{\substack{V \in \mathbb{R}^{n \times p} \\ V^T V = I}} \frac{\text{Tr}(V^T S_E V)}{\text{Tr}(V^T S_I V)} \quad (1.9)$$

La solución a este problema no tiene una forma cerrada, por lo que en la literatura se buscan formulaciones alternas para resolverlo de una manera más sencilla [9] [3], algunos ejemplos de estas formulaciones son:

$$\max_{\substack{V \in \mathbb{R}^{n \times p} \\ V^T S_I V = I}} \text{Tr}(V^T S_E V) \quad (1.10)$$

$$\max_{\substack{V \in \mathbb{R}^{n \times p} \\ V^T V = I}} \text{Tr} \left( \frac{V^T S_E V}{V^T S_I V} \right) \quad (1.11)$$

$$\max_{\substack{V \in \mathbb{R}^{n \times p} \\ V^T V = I}} \frac{|V^T S_E V|}{|V^T S_I V|} \quad (1.12)$$

Con  $|\bullet| = \det(\bullet)$  y  $\text{Tr}(\bullet) = \text{Traza}(\bullet)$ .

En la siguiente parte de este capítulo se resolverá el problema original (1.9) para  $p = 1$ , para lo cual se introduce el cociente generalizado de Rayleigh. Para la generalización a  $p$  dimensiones solo se plantea el problema y se proporcionan los casos en que la solución existe y es única. Seguido de esto, se definirá una función  $f(\rho)$  la cual sirve para encontrar el óptimo por métodos iterativos. Por último haciendo uso de los eigenvalores de  $S_I$  y  $S_E$  se darán cotas inferiores y superiores al óptimo.

### 1.3. Problema del cociente de trazas

En este texto se utilizará el término de *proyección* a la transformación ortogonal dada por la matriz  $V = [V_1|V_2|\dots|V_p]$  tal que  $V \in \mathbb{R}^{n \times p}$ .<sup>2</sup> Sea  $X = [X_1|X_2|\dots|X_m]$  la matriz de individuos de tamaño  $\mathbb{R}^{n \times m}$ , entonces al desarrollar el producto  $V^T X$ :

---

<sup>2</sup>Esta matriz, al tener columnas ortogonales, y suponiendo que cada columna tiene norma 1, cumple la característica que  $V^T V = I_p$

$$V^T X = \begin{pmatrix} V_1^T(X_1) & V_1^T(X_2) & \dots & V_1^T(X_m) \\ V_2^T(X_1) & V_2^T(X_2) & \dots & V_2^T(X_m) \\ \vdots & \vdots & \ddots & \vdots \\ V_p^T(X_1) & V_p^T(X_2) & \dots & V_p^T(X_m) \end{pmatrix}$$

Si se observa, cada elemento de la matriz  $V^T X$  tiene la forma  $V_i^T X_j$ , es decir un producto punto entre una columna de  $V$  y un individuo  $X_j$ . Por otra parte, se tiene que  $proj_{V_i} X_j = \frac{X_j \cdot V_i}{\|V_i\|^2} V_i$ . Como  $\|V_i\| = 1$ , entonces la fórmula queda simplificada como  $proj_{V_i} X_j = (X_j \cdot V_i) V_i$ ; es decir, un vector de magnitud  $(X_j \cdot V_i)$  con la dirección  $V_i$ . Por este motivo se llamarán los individuos proyectados, ya que cada individuo está proyectado a cada columna de  $V$ .

El problema del cociente de trazas (Trace ratio problem) es fácil de ver cuando  $V \in \mathbb{R}^{n \times p}$  proyecta a un espacio de pocas dimensiones. Por ejemplo, cuando  $p = 2$ , se desea obtener la mejor proyección sobre un plano y cuando  $p = 1$ , sobre una recta. Para ejemplificar esta situación se creo un conjunto sintético donde cada  $x_i \in \mathbb{R}^3$ . Las distribuciones de los datos son normales y se proyectan en  $\mathbb{R}^2$  y  $\mathbb{R}^1$ . Los datos se pueden observar en la figura 1.3.

### 1.3.1. Solución cuando $p = 1$

El problema (1.9) toma la siguiente forma cuando  $V \in \mathbb{R}^n$ . Se nombrará  $v$  a este proyector de una dimensión ya que resulta ser solo un vector:

$$\max_{v \in \mathbb{R}^n} \frac{v^T S_E v}{v^T S_I v} \quad (1.13)$$

Se tiene que  $x_i \in \mathbb{R}^n$  son los individuos originales con  $i = 1, \dots, N$ . Entonces sean  $w_i \in \mathbb{R}$  los individuos proyectados por el vector  $v$  de manera que  $w_i = v^T x_i$ . De esta manera, es conveniente definir  $\hat{\mu}_k = v^T \mu_k$  y  $\hat{\mu} = v^T \mu$  como la media por clase y la media total de los datos proyectados. Las matrices de dispersión entre clases e intra clase se definen como sigue:

**Matriz de dispersión entre clases  $\Phi_E$  de los individuos proyectados  $w_i$ :**

$$\Phi_E = \sum_{k=1}^K N_k (\hat{\mu}_k - \hat{\mu})^2$$

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

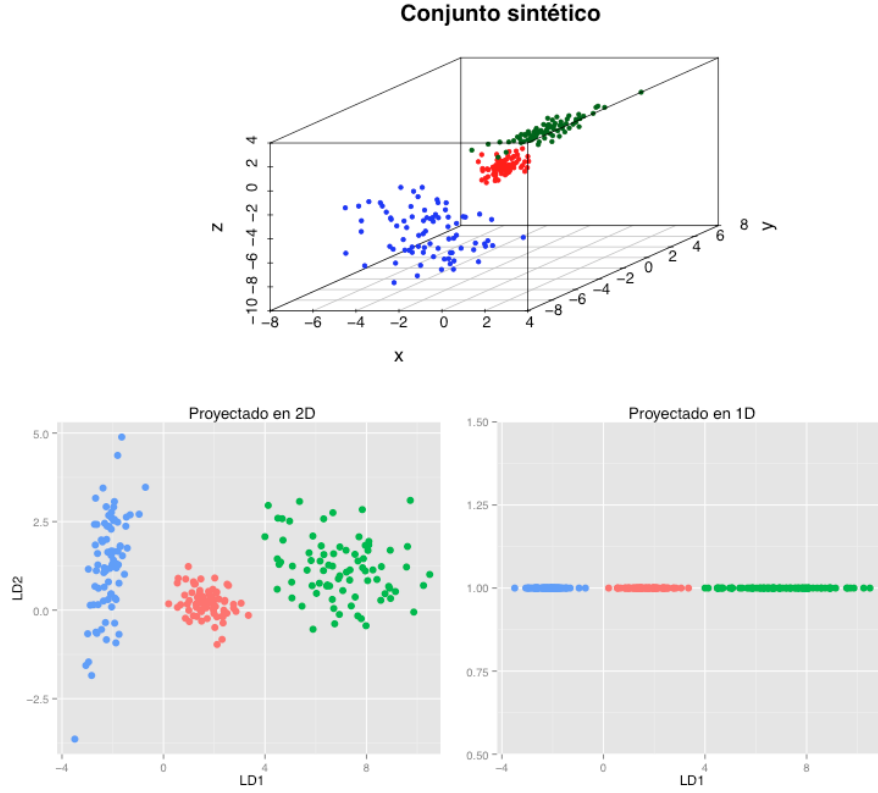


Figura 1.3: En la gráfica de arriba se muestran los datos originales en  $\mathbb{R}^3$  los cuales fueron generados a través de distribuciones normales con distintas medias y varianzas. En la gráfica de abajo a la izquierda se muestra la mejor proyección en  $\mathbb{R}^2$  y abajo a la derecha la mejor proyección en  $\mathbb{R}$ .

$$\Phi_E = \sum_{k=1}^K N_k (v^T \mu_k - v^T \mu)^2$$

$$\Phi_E = \sum_{k=1}^K N_k v^T (\mu_k - \mu) (\mu_k - \mu)^T v$$

Por distributividad en matrices se cumple que  $vAv + vBv = v(A+B)v$ , entonces:

$$\Phi_E = v^T \left[ \sum_{k=1}^K N_k (\mu_k - \mu) (\mu_k - \mu)^T \right] v \quad (1.14)$$

**Matriz de dispersión intra clase  $\Phi_I$  de los individuos proyectados  $w_i$ :**

$$\begin{aligned}\Phi_I &= \sum_{k=1}^K \sum_{\substack{i=1 \\ y_i=k}}^N (w_i - \hat{\mu}_k)^2 \\ \Phi_I &= \sum_{k=1}^K \sum_{\substack{i=1 \\ y_i=k}}^N (v_i^T x_i - v_i^T \mu_k)^2 \\ \Phi_I &= \sum_{k=1}^K \sum_{\substack{i=1 \\ y_i=k}}^N v^T (x_i - \mu_k)(x_i - \mu_k)^T v\end{aligned}$$

Usando de nuevo la distributividad de matrices:

$$\Phi_I = v^T \left[ \sum_{k=1}^K \sum_{\substack{i=1 \\ y_i=k}}^N (x_i - \mu_k)(x_i - \mu_k)^T \right] v \quad (1.15)$$

Las fórmulas de  $\Phi_I$  y  $\Phi_E$  de los individuos  $w_i$  se pueden expresar en función de las matrices de dispersión intra clase y entre clases  $S_I$  y  $S_E$  de los individuos originales  $x_i$ . De esta manera:

$$\begin{aligned}\Phi_E &= f(S_E) = v^T S_E v \\ \Phi_I &= f(S_I) = v^T S_I v\end{aligned}$$

Se tiene que  $\Phi_I, \Phi_E \in \mathbb{R}$ , entonces maximizar el cociente  $\frac{\Phi_E}{\Phi_I}$  con respecto a  $v$  tiene como resultado una proyección que conserva cerca a los individuos pertenecientes a la misma clase, mientras que aleja a los centros de cada clase. Para el caso de una dimensión se puede encontrar una solución cerrada. La teoría asociada a este problema de maximización esta relacionada con el Cociente Generalizado de Rayleigh, el cual bajo las condiciones enunciadas de este caso, se puede transformar a un Cociente de Rayleigh.

**Proposición 1.1.** *La solución a la maximización del Cociente de Rayleigh:*

$$\max_{v \in \mathbb{R}^n} \frac{v^T A v}{v^T v}$$

*cuando  $A$  es simétrica, es obtenida cuando  $v$  es el eigenvector asociado al eigenvalor más grande de la matriz  $A$ .*

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

*Demostración.* Primero se hace la observación que factorizando  $c$  las siguientes dos formulaciones de la maximización son equivalentes cuando  $c \neq 0$  y  $c \in \mathbb{R}$ :

$$\begin{aligned} \max_{v \in \mathbb{R}^n} \frac{v^T A v}{(v^T v)} \\ \max_{v \in \mathbb{R}^n} \frac{(cv)^T A (cv)}{(cv)^T (cv)} \end{aligned}$$

Sin pérdida de generalidad se supone que  $\|v\| = 1$ . Cuando  $v \in \mathbb{R}^{n \times 1}$  el problema es equivalente a la maximización del numerador sujeta a una restricción de igualdad  $v^T v = 1$ :

$$\max_{\substack{v \in \mathbb{R}^n \\ v^T v = 1}} v^T A v$$

Este problema es más sencillo y se puede resolver formulando la función lagrangiana asociado, derivando e igualando a 0:

$$\begin{aligned} \mathcal{L}(v, \lambda) &= v^T A v - \lambda(v^T v - 1) \\ \frac{\partial \mathcal{L}(v, \lambda)}{\partial v} &= (A + A^T)v - 2\lambda v = 0 \end{aligned}$$

Como  $A$  es simétrica, entonces  $A + A^T = 2A$ , por lo que la solución es:

$$Av = \lambda v \tag{1.16}$$

Este es el problema de eigenvalores generalizado. Entonces el máximo es alcanzado cuando  $v$  es el eigenvector asociado al eigenvalor más grande de  $A$ .

□

### 1.3.2. Generalización a $p$ dimensiones

Para dimensiones más grandes de  $v$ , el Cociente Generalizado de Rayleigh no puede ser escrito en general como el Cociente de Rayleigh, por lo que la solución planteada en el capítulo anterior no es de utilidad. Esto genera la dificultad de no tener una solución cerrada, por lo que se han propuesto métodos iterativos y planteamientos alternos a la solución.

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

La generalización a  $p$  dimensiones implica que los individuos  $x_i \in \mathbb{R}^n$  son proyectados ahora por la matriz  $V = (V_1|V_2|\dots|V_p)$ , de manera que  $w_i = V^T x_i$  con  $w_i \in \mathbb{R}^p$  y  $V_j \in \mathbb{R}^n$ . De esta forma, las matrices  $\Phi_I$  y  $\Phi_E$  se definen como sigue:

$$\begin{aligned}\Phi_E &= \sum_{k=1}^K N_k \|\hat{\mu}_k - \hat{\mu}\|_2^2 \\ \Phi_E &= \sum_{k=1}^K N_k \|V^T \mu_k - V^T \mu\|_2^2 \\ \Phi_E &= \sum_{k=1}^K N_k \|V^T (\mu_k - \mu)\|_2^2\end{aligned}$$

$$\Phi_E = \sum_{k=1}^K N_k [(V_1^T (\mu_k - \mu))^2 + (V_2^T (\mu_k - \mu))^2 + \dots + (V_p^T (\mu_k - \mu))^2] \quad (1.17)$$

De esta expresión hay que destacar que  $V_1^T (\mu_k - \mu)$  es un escalar, ya que  $V_1 \in \mathbb{R}^n$  y  $(\mu_k - \mu) \in \mathbb{R}^n$ . Otra fórmula equivalente y que es comúnmente usada por sus propiedades algebraicas consiste en la siguiente expresión:

$$\Phi_E = \sum_{k=1}^K N_k \text{Tr} [V^T (\mu_k - \mu)(\mu_k - \mu)^T V] \quad (1.18)$$

Para mostrar que estas dos formulaciones son iguales, se toma una clase  $k = k_1$ . Al desarrollar  $(\bullet) = V^T (\mu_1 - \mu)(\mu_1 - \mu)^T V$  se tiene una matriz en  $\mathbb{R}^{p \times p}$  igual a:

$$(\bullet) = \begin{pmatrix} V_1^T (\mu_1 - \mu) \\ V_2^T (\mu_1 - \mu) \\ \vdots \\ V_p^T (\mu_1 - \mu) \end{pmatrix} \begin{pmatrix} (\mu_1 - \mu)^T V_1 & (\mu_1 - \mu)^T V_2 & \dots & (\mu_1 - \mu)^T V_p \end{pmatrix}$$

$$(\bullet) = \begin{pmatrix} V_1^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_1 & \dots & V_1^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_p \\ V_2^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_1 & \dots & V_2^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_p \\ \vdots & \ddots & \vdots \\ V_p^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_1 & \dots & V_p^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_p \end{pmatrix}$$

$$(\bullet) = \begin{pmatrix} (V_1^T(\mu_1 - \mu))^2 & \dots & V_1^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_p \\ V_2^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_1 & \dots & V_2^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_p \\ \vdots & \ddots & \vdots \\ V_p^T(\mu_1 - \mu)(\mu_1 - \mu)^T V_1 & \dots & (V_p^T(\mu_1 - \mu))^2 \end{pmatrix}$$

Al tomar la diagonal de la matriz desarrollada arriba se tiene que  $\text{Tr}(V^T(\mu_1 - \mu)(\mu_1 - \mu)^T V)$  es equivalente a:

$$\text{Tr}(\bullet) = (V_1^T(\mu_1 - \mu))^2 + (V_2^T(\mu_1 - \mu))^2 + \dots + (V_p^T(\mu_1 - \mu))^2$$

Al sumar esta expresión sobre las  $K$  clases y usando la propiedad de linealidad en la traza; es decir,  $\text{Tr}(A + B) = \text{Tr}(A) + \text{Tr}(B)$ , entonces se puede escribir de la siguiente manera:

$$\Phi_E = \text{Tr} \sum_{k=1}^K N_k [V^T(\mu_k - \mu)(\mu_k - \mu)^T V]$$

Esta expresión resulta ser idéntica a la expresión (1.16). Como paso final se factoriza  $V^T$  y  $V$  sobre todos los sumandos, lo que nos llevaría a lo siguiente:

$$\Phi_E = \text{Tr}(V^T \sum_{k=1}^K N_k [(\mu_k - \mu)(\mu_k - \mu)^T] V)$$

o, expresada en términos de  $S_E$ :

$$\Phi_E = \text{Tr}(V^T S_E V) \quad (1.19)$$

Similarmente se puede llegar a la formulación de la varianza intra-clase  $\Phi_I$ .

$$\Phi_I = \text{Tr}(V^T S_I V) \quad (1.20)$$



### 1.3.3. Existencia de la solución

Para demostrar la existencia y unicidad de la solución, las matrices  $S_I$  y  $S_E$  deben cumplir ciertas características. Sean  $A = S_E$  y  $B = S_I$ , la primer condición que se les impone es que sean positivas definidas. La razón que apoya la restricción está relacionada con la forma de la función a maximizar, que es un cociente. Como  $B$  se encuentra en el denominador, se tiene que evitar que  $\text{Tr}(V^T B V) = 0$ , ya que con este valor se indetermina la función objetivo [8].

T.T. Ngo propone generalizar el estudio a las matrices positivas semidefinidas. Para esto se deben encontrar los casos en que  $\text{Tr}(V^T B V)$  toma el valor de 0. Si se diagonaliza a la matriz  $B = Q\Lambda_B Q^T$  con  $Q$  ortogonal y  $\Lambda_B$  una matriz diagonal con entradas iguales a los eigenvalores de  $B$ , entonces:

$$\text{Tr}(\Lambda_B) = \lambda_{B_1} + \lambda_{B_2} + \dots + \lambda_{B_n} \quad \text{con} \quad \hat{V} = Q^T V$$

De este modo  $\hat{V} = (\hat{V}_1 | \hat{V}_2 | \dots | \hat{V}_p)$  y cada  $\hat{V}_i^T = (\hat{V}_{i1}, \hat{V}_{i2}, \dots, \hat{V}_{in})$  es un vector renglón. De esta manera la matriz  $\hat{V}^T$ :

$$\hat{V}^T = \begin{pmatrix} \hat{V}_1^T \\ \hat{V}_2^T \\ \vdots \\ \hat{V}_p^T \end{pmatrix} = \begin{pmatrix} \hat{V}_{11} & \hat{V}_{12} & \dots & \hat{V}_{1n} \\ \hat{V}_{21} & \hat{V}_{22} & \dots & \hat{V}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{V}_{p1} & \hat{V}_{p2} & \dots & \hat{V}_{pn} \end{pmatrix}$$

Por lo que la traza que involucra a  $B$  tiene la siguiente forma:

$$\text{Tr}(V^T B V) = \text{Tr}(V^T Q \Lambda_B Q^T V)$$

$$\text{Tr}(V^T B V) = \text{Tr}(\hat{V}^T \Lambda_B \hat{V})$$

$$V^T B V = \begin{pmatrix} \hat{V}_1^T \\ \hat{V}_2^T \\ \vdots \\ \hat{V}_p^T \end{pmatrix} \begin{pmatrix} \lambda_{B_1} & 0 & \dots & 0 \\ 0 & \lambda_{B_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_{B_n} \end{pmatrix} \begin{pmatrix} \hat{V}_1 | \hat{V}_2 | \dots | \hat{V}_p \end{pmatrix}$$

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

Desarrollando la multiplicación de matrices, y calculando la traza resulta en los siguientes sumandos:

$$\begin{aligned} \text{Tr}(V^T BV) = & \lambda_{B_1} \widehat{V}_{11}^2 + \lambda_{B_2} \widehat{V}_{12}^2 + \dots + \lambda_{B_n} \widehat{V}_{1n}^2 + \\ & \lambda_{B_1} \widehat{V}_{21}^2 + \lambda_{B_2} \widehat{V}_{22}^2 + \dots + \lambda_{B_n} \widehat{V}_{2n}^2 + \\ & \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ & \lambda_{B_1} \widehat{V}_{p1}^2 + \lambda_{B_2} \widehat{V}_{p2}^2 + \dots + \lambda_{B_n} \widehat{V}_{pn}^2. \end{aligned}$$

Es fácil de ver que la expresión de arriba tiene  $p \times n$  sumandos, por lo que se puede expresar en términos de dos sumatorias. La primera de  $j = 1, \dots, p$  y la segunda de  $i = 1, \dots, n$ :

$$\begin{aligned} \text{Tr}(V^T BV) &= \sum_{j=1}^p \sum_{i=1}^n \lambda_{B_i} \widehat{V}_{ji}^2 \\ \text{Tr}(V^T BV) &= \sum_{i=1}^n \lambda_{B_i} \sum_{j=1}^p \widehat{V}_{ji}^2 \end{aligned} \tag{1.21}$$

De la última expresión se separa la sumatoria sobre  $i$ . De esta manera, para cada elemento  $i$  se tienen dos factores:

$$(i) \lambda_{B_i} \tag{1.22}$$

$$(ii) \sum_{j=1}^p \widehat{V}_{ji}^2 \tag{1.23}$$

La idea para que  $\text{Tr}(V^T BV)$  sea positivo, es que al menos uno de los sumandos sea positivo. Si (1.21) y (1.22) son ambos distintos de cero para al menos una  $i$ , entonces se cumple esta condición. Esta idea está expresada en el Lema 1.1.

**Lema 1.1.** *Sea  $B$  positiva semidefinida y  $V \in \mathbb{R}^{n \times p}$ . Si  $B$  tiene a lo más  $p-1$  eigenvalores iguales a 0, entonces  $\text{Tr}(V^T BV) = \text{Tr}(\widehat{V}^T \Lambda_B \widehat{V}) \neq 0$  para cualquier matriz ortogonal  $V$ .*

*Demostración.* Sea  $\widehat{V} = [\widehat{V}_1 | \dots | \widehat{V}_p]$  tal que  $\widehat{V}^T \widehat{V} = V^T Q Q^T V = V^T I_n V = I_p$ . De esta manera se puede construir una matriz  $\widehat{V}' \in \mathbb{R}^{p \times p}$  seleccionando  $p$  de

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

los  $n$  renglones de  $\widehat{V}$  tal que  $\widehat{V}'$  sea no singular.  $\widehat{V}'$  tiene la propiedad de no contener eigenvalores iguales a 0; como consecuencia, sus renglones y columnas no contienen al vector  $\widehat{0}$ . Al no contenerlo, se sabe que al menos hay  $p$  renglones de  $\widehat{V}$  tal que  $\sum_{j=1}^p \widehat{V}_{ji}^2 \neq 0$  para cada uno de ellos. Por otra parte en el lema se supone que la matriz  $B$  tiene a lo más  $p-1$  eigenvalores iguales 0 por lo que al menos un elemento de la sumatoria es distinto de cero.  $\square$

Analizando a mayor profundidad el resultado anterior, se sabe que hay  $n-p+1$  eigenvalores de  $B$  positivos ( $\lambda_{B_i} \neq 0$ ) y  $p$  renglones de  $\widehat{V}$  que tienen norma distinta de cero. Al calcular la sumatoria (1.20), se tiene que al menos una combinación de  $\lambda_{B_i}$  y uno de los  $p$  renglones cumplen que su multiplicación tiene signo positivo. Para ejemplificar esta situación sean  $C_i$  con  $i = 1, \dots, n-p+1$  los eigenvalores de  $B$  y  $K_j$  con  $j = 1, \dots, p$  la norma de los renglones de  $\widehat{V}$  que son distintos de 0.

$i$	$\lambda_{B_i}$	$\sum_{j=1}^p \widehat{V}_{ji}^2$	$\lambda_{B_i} \sum_{j=1}^p \widehat{V}_{ji}^2$
1	$C_1$	0	0
2	$C_2$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n-p$	$C_{n-p}$	0	0
$n-p+1$	$C_{n-p+1}$	$K_p$	$C_{n-p+1}K_p$
$n-p+2$	0	$K_{p-1}$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n-1$	0	$K_2$	0
$n$	0	$K_1$	0

Con esta combinación se tiene que al menos hay un sumando de  $\sum_{i=1}^n \lambda_{B_i} \sum_{j=1}^p \widehat{V}_{ji}^2$  distinto de cero, por lo que  $\text{Tr}(V^T B V) \neq 0$ . Bajo estas condiciones se garantiza que el denominador sea mayor a 0, solo falta asegurarse que el numerador sea menor a infinito.

**Lema 1.2.** Sea  $U_p = \{V \in \mathbb{R}^{n \times p} | V^T V = I_p\}$  un conjunto compacto con  $V = (v_1, v_2, \dots, v_p)$

*Demostración.* Se tiene que  $U_p$  es un conjunto cerrado porque contiene a todos sus puntos límite; por otro lado,  $U_p$  también es acotado bajo la norma 2 y la norma de Frobenius:

Tomando la norma-2 y la norma de Frobenius de  $V$ :

$$\begin{aligned} \|V\|_2 &= \text{Max}\{\|V_x\|_2 \mid \|x\|_2 = 1\} \\ &= \|V_x\|_2^2 \\ &= (Vx)^T(Vx) \\ &= x^T V^T V x \\ &= x^T x = 1 \\ \|V\|_F &= \sum_F^p \|v_i\| = p \end{aligned}$$

Entonces se tiene  $U_p$  es cerrado y acotado, por lo que  $U_p$  es compacto.  $\square$

Con este resultado se tiene que  $\text{Tr}(V^T AV)$  toma un valor finito ya que todas sus entradas son finitas.

**Lema 1.3.** Sean  $A$  y  $B$  dos matrices simétricas tales que  $B$  es positiva semi-definida con rango mayor que  $n - p$ ; es decir, que tenga al menos  $n - p + 1$  eigenvalores distintos de cero. Entonces el cociente (1.9) admite un máximo con valor  $\rho^*$  [8].

*Demostración.* Tomando el resultado del lema 1.1 se tiene que  $\text{Tr}(V^T BV) \neq 0$ ; por otra parte,  $V \in U_p$  que es un conjunto compacto. Con estas dos observaciones, el valor de (1.9) es distinto de infinito. Entonces el cociente (1.9) admite un máximo con valor  $\rho^*$  y que tiene como argumento  $V^{**}$ .  $\square$

#### 1.3.4. Equivalencia con un problema escalar

**Valor en el óptimo.** Del lema 1.3 se sabe que existe una matriz  $V^{**} \in U_p$  tal que (1.9) alcanza el valor máximo  $\rho^*$ . Expresando esta idea se tiene que:

$$\frac{\text{Tr}(V^{T**} AV^{**})}{\text{Tr}(V^{T**} BV^{**})} = \rho^* \quad (1.24)$$

Entonces para cualquier otra matriz  $V \in U_p$ :

$$\frac{\text{Tr}(V^T AV)}{\text{Tr}(V^T BV)} \leq \rho^* \quad (1.25)$$

Como la traza es un operador lineal y por la propiedad distributiva de las matrices entonces (1.24) es equivalente a:

$$\text{Tr}(V^T AV) - \rho^* \text{Tr}(V^T BV) \leq 0$$

$$\text{Tr}(V^T AV - \rho^* V^T BV) \leq 0$$

$$\text{Tr}(V^T (A - \rho^* B)V) \leq 0 \quad (1.26)$$

y el resultado equivalente para (1.23):

$$\text{Tr}(V^{T**}(A - \rho^* B)V^{**}) = 0 \quad (1.27)$$

Para facilitar la lectura, de aquí en adelante se define la función  $G(\rho) = A - \rho B$ . Maximizar el lado izquierdo de la desigualdad (1.25) sujeto a  $V^T V = I$  es equivalente a maximizar un problema de eigenvalores generalizado. Usando lo establecido en el apéndice A, se sabe que el valor máximo de este problema dado  $\rho^*$  es igual a:

$$\max_{\substack{V \in \mathbb{R}^{n \times p} \\ V^T V = I}} \text{Tr}(V^T G(\rho^*)V) = \lambda_{G(\rho^*)_1} + \lambda_{G(\rho^*)_2} + \dots + \lambda_{G(\rho^*)_p} \quad (1.28)$$

Con  $\lambda_{G(\rho^*)_1} \geq \lambda_{G(\rho^*)_2} \geq \dots \geq \lambda_{G(\rho^*)_p}$  los  $p$  eigenvalores más grandes de  $G(\rho^*)$ . De esta manera el valor óptimo de (1.27) es simplemente la suma de los  $p$  eigenvalores más grandes de esta matriz, y  $V^{**}$  el conjunto de correspondientes eigenvectores. Para obtener este valor y la matriz, el primer paso es encontrar a  $\rho^*$ , ya que teniéndolo es inmediato calcular  $V^{**}$ . Dada esta premisa, se puede ver que el problema a resolver se reduce a buscar el valor óptimo de  $\rho$ . Para esto se define la función  $f(\rho)$  sobre todo  $\mathbb{R}$ , tal que  $f(\rho)$  es continua sobre su argumento  $\rho$ :

$$f(\rho) = \max_{V^T V = I} \text{Tr}(V^T (G(\rho)) V) \quad (1.29)$$

Es conveniente examinar  $f(\rho)$  con dos objetivos, el primero es estimar la dificultad de calcular el valor de  $f(\rho)$  y el segundo es encontrar la maximización adecuada para obtener  $\rho^*$ . Respecto al primer punto, la manera de calcular  $f(\rho)$  en cada punto es equivalente a (1.27), pero en lugar de usar los eigenvalores de  $G(\rho^*)$  se usan los de  $G(\rho)$ . En particular se llamará  $V(\rho)^*$  al argumento que resuelve (1.28) <sup>3</sup>. Sean  $\lambda_{G(\rho)_1} \geq \lambda_{G(\rho)_2} \geq \dots \geq \lambda_{G(\rho)_n}$  los  $n$  eigenvalores de  $G(\rho)$ . Con esta notación  $f(\rho)$  toma el valor de:

$$f(\rho) = \lambda_{G(\rho)_1} + \lambda_{G(\rho)_2} + \dots + \lambda_{G(\rho)_p} \quad (1.30)$$

Para el segundo punto, la idea es iterar hasta obtener el valor de  $\rho^*$ . Por esto, es conveniente analizar como se comporta la función con respecto a su argumento. A continuación se presentan dos propiedades de  $f(\rho)$ . Para demostrarlas, primero se enuncia el teorema 8.1.5 de [4].

**Teorema 1.1.** *Sean  $X$  y  $X + E$  matrices simétricas  $n \times n$ , y  $\lambda_{X_k}$ ,  $\lambda_{E_k}$  los  $k$ -ésimos eigenvalores más grandes de  $X$  y  $E$  respectivamente. De esta manera  $\lambda_{X_k}$  es el  $k$ -ésimo eigenvalor más grande de  $X$  y  $\lambda_{E_1}$  el más grande de  $E$ . Con estas definiciones se cumple lo siguiente:*

$$\lambda_{X_k} + \lambda_{E_n} \leq \lambda_{(X+E)_k} \leq \lambda_{X_k} + \lambda_{E_1} \quad (1.31)$$

Se procede a enunciar este lema acerca de la función  $f(\rho)$

**Lema 1.4.** *La función  $f(\rho) = \max_{V^T V = I} \text{Tr}(V^T (A - \rho B) V)$  cumple las siguientes dos propiedades:*

- (1)  *$f$  es una función no creciente de  $\rho$*
- (2)  *$f(\rho) = 0$  si y solo si  $\rho = \rho^*$*

Para probar la parte (1) del lema 1.4 se comparan los valores de  $f(\rho)$  para  $\rho_1$  y  $\rho_2$  con  $\rho_2 \geq \rho_1$ . Como se desea demostrar que  $f$  es una función no creciente de

---

<sup>3</sup>Se utilizó la nomenclatura de  $V(\rho^*)$  porque estos eigenvectores dependen del valor de  $\rho$  en la matriz  $A - \rho B$ .

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

$\rho$ , se busca que  $f(\rho_2) \leq f(\rho_1)$ . Se definen las matrices  $Y = X + E$  y  $E = Y - X$ , para después restarles  $\lambda_{X_k}$ , entonces (1.29) se puede escribir como:

$$\lambda_{(X)_k} + \lambda_{(Y-X)_n} \leq \lambda_{(Y)_k} \leq \lambda_{(X)_k} + \lambda_{(Y-X)_1}$$

$$\lambda_{(Y-X)_n} \leq \lambda_{(Y)_k} - \lambda_{(X)_k} \leq \lambda_{(Y-X)_1}$$

La expresión en el centro de estas desigualdades es la resta del eigenvalor  $k$ -ésimo de la matriz  $X$  y  $Y$ , por lo que si se realiza la suma de  $k = 1$  a  $k = p$  se tiene lo siguiente:

$$p\lambda_{(Y-X)_n} \leq \sum_{k=1}^p \lambda_{(Y)_k} - \sum_{k=1}^p \lambda_{(X)_k} \leq p\lambda_{(Y-X)_1} \quad (1.32)$$

Definiendo  $X = A - \rho_2 B$  y  $Y = A - \rho_1 B$ , entonces (1.31) toma la siguiente forma:

$$p\lambda_{(B(\rho_2-\rho_1))_n} \leq \sum_{k=1}^p \lambda_{(A-\rho_1 B)_k} - \sum_{k=1}^p \lambda_{(A-\rho_2 B)_k} \leq p\lambda_{(B(\rho_2-\rho_1))_1} \quad (1.33)$$

Retomando el resultado (1.29), y sustituyéndolo en (1.32) la desigualdad queda de la siguiente forma:

$$p\lambda_{(B(\rho_2-\rho_1))_n} \leq f(\rho_1) - f(\rho_2) \leq p\lambda_{(B(\rho_2-\rho_1))_1} \quad (1.34)$$

En la parte izquierda de la desigualdad anterior, se puede determinar el signo que toman los eigenvalores  $\lambda_{(B(\rho_2-\rho_1))_n}$ . Si  $(\rho_2 - \rho_1) \geq 0$  entonces la matriz  $(\rho_2 - \rho_1)B$  es positiva semidefinida; por lo tanto, todos sus eigenvalores son mayores o iguales a 0:

$$0 \leq p\lambda_{(B(\rho_2-\rho_1))_n} \leq f(\rho_1) - f(\rho_2) \leq p\lambda_{(B(\rho_2-\rho_1))_1}$$

$$0 \leq f(\rho_1) - f(\rho_2) \quad (1.35)$$

De esta manera  $f(\rho_2) \leq f(\rho_1)$  cuando  $\rho_2 \geq \rho_1$

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

Para probar la parte (2) del lema 1.4, se tiene que demostrar la condición suficiente y la condición necesaria. La demostración de la primera es inmediata, ya que cuando  $\rho = \rho^*$ , entonces por (1.26),  $f(\rho) = 0$ . Para la condición necesaria se usará (1.23) y la propiedad que la  $\text{Tr}(V^T BV) > 0 \ \forall \ V \in U_p$ . Se demostrará que, dado  $f(\rho^*) = 0$ , entonces:

$$(i) \quad f(\rho) < 0 \quad \text{si} \quad \rho > \rho^* \quad (1.36)$$

$$(ii) \quad f(\rho) > 0 \quad \text{si} \quad \rho < \rho^* \quad (1.37)$$

### Caso (i) $\rho > \rho^*$

Se tiene que las siguientes desigualdades se cumplen:

$$\frac{\text{Tr}(V^T AV)}{\text{Tr}(V^T BV)} \leq \rho^* < \rho$$

Por lo que es equivalente a:

$$\text{Tr}(V^T (A - \rho B)V) < 0 \quad \forall V \in U_p$$

De esta manera  $f(\rho) < 0$  cuando  $\rho > \rho^*$ .

### (ii) $\rho < \rho^*$

Retomando el resultado (1.24) y suponiendo que  $\rho^* > \rho$  entonces existe una  $V^*$  tal que:

$$\rho < \frac{\text{Tr}(V^{T*} AV^*)}{\text{Tr}(V^{T*} BV^*)} \leq \rho^*$$

$$\text{Tr}(V^{T*} AV^* - \rho V^{T*} BV^*) > 0 \quad \implies \quad \frac{\text{Tr}(V^{T*} AV^*)}{\text{Tr}(V^{T*} BV^*)} > \rho$$

En particular:

$$\max_{V^T V} \frac{\text{Tr}(V^T AV)}{\text{Tr}(V^T BV)} > \rho$$

De esta manera  $f(\rho) > 0$  cuando  $\rho < \rho^*$ .



## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

Las ecuaciones (1.35) (1.36), junto con la continuidad de la función  $f$ , muestran que  $f(\rho) = 0$  implica  $\rho = \rho^*$  [8]. De esta manera el problema puede ser visto como encontrar la raíz de la función  $f(\rho)$ . La figura 1.4 muestra como es esta función.

**Corolario 1.1.** *La función  $f(\rho) = \max_{V^T V = I} \text{Tr}(V^T (A - \rho B) V)$  cumple las siguientes condiciones:*

$$f(\rho) > 0 \quad \forall \quad \rho \in (-\inf, \rho^*)$$

$$f(\rho) < 0 \quad \forall \quad \rho \in (\rho^*, \inf)$$

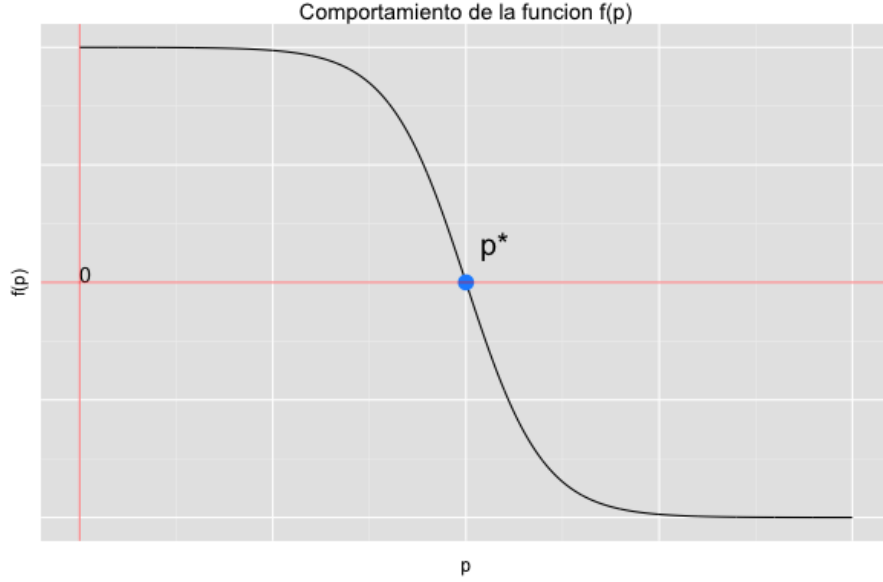


Figura 1.4: La función  $f(\rho)$  es no creciente para toda  $\rho$ . El valor de  $f(\rho) = \lambda_{G(\rho)1} + \lambda_{G(\rho)2} + \dots + \lambda_{G(\rho)p}$ .  $f(\rho^*) = 0$

**Ejemplo 1.1.** *Para ejemplificar el lema 1.4 se muestran las matrices  $A, B \in \mathbb{R}^{3 \times 3}$ . Para el valor de  $f(\rho)$  se utiliza la propiedad (1.29):*

$$f(\rho) = \lambda_{G(\rho)1} + \lambda_{G(\rho)2} + \dots + \lambda_{G(\rho)p}$$

con  $p$  la dimensión a la que se va a proyectar.

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

$$A = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 8 \end{pmatrix}, \quad B = \begin{pmatrix} 1.5 & 0 & 0 \\ 0 & 2.5 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

$$G(\rho) = A - \rho B = \begin{pmatrix} 4 - 1.5\rho & 0 & 0 \\ 0 & 6 - 2.5\rho & 0 \\ 0 & 0 & 8 - 5\rho \end{pmatrix}$$

Los eigenvalores de esta matriz son  $4 - 1.5\rho$ ,  $6 - 2.5\rho$  y  $8 - 5\rho$ . Las funciones graficadas de estos eigenvalores se presentan en la figura 1.5.

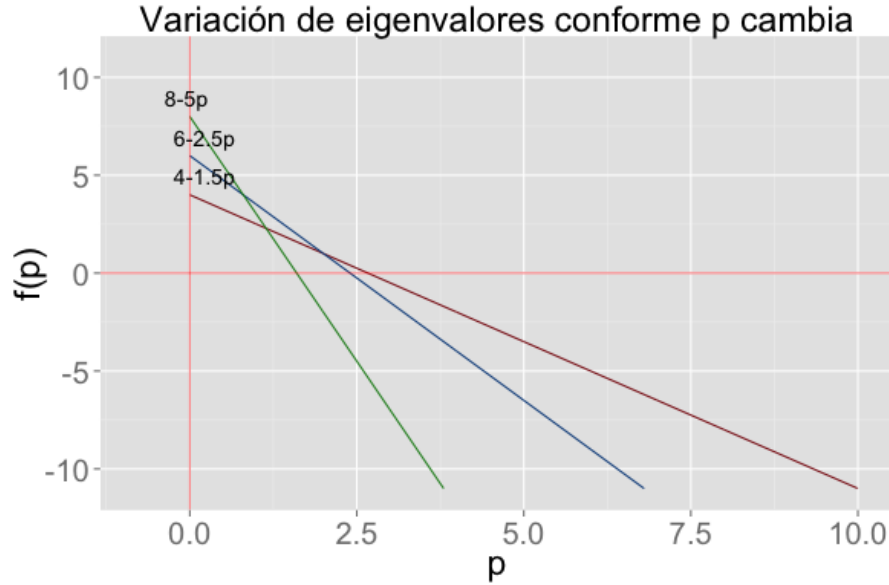


Figura 1.5: Cada línea representa como se comporta cada eigenvalor de  $A - \rho B$  cuando se varía  $\rho$ .

Cuando se desea que el proyector sea de dimensión 1, entonces se tiene que  $f(\rho)$  es el eigenvalor más grande, cuando sea de dimensión 2, la suma de los dos más grandes y así respectivamente. El valor de  $f(\rho)$  para estos tres casos está representado en la figura 1.5.

### 1.3.5. Localización del óptimo

En la sección anterior se encontró que la maximización al cociente de trazas puede ser visto como el problema de encontrar la raíz de la función  $f(\rho) = \max_{V^T V = I} \text{Tr}(V^T(A - \rho B)V)$ . Por esto, encontrar un intervalo  $(\rho_1, \rho_2)$  que contenga al valor óptimo  $\rho^*$  puede reducir el número de iteraciones del método. Usando el lema 1.4 se sabe que  $f$  es una función no creciente de  $\rho$ . Por esta razón si se encuentra una  $\rho_1$  y  $\rho_2$  tal que  $f(\rho_1) \geq 0$  y  $f(\rho_2) \leq 0$  y con la propiedad de continuidad de la función  $f(\rho)$  entonces se encontró un intervalo que contiene a  $\rho^*$ .

En esta tesis se dan cotas para el valor de  $\rho^*$ , la primera en función los eigenvalores de una transformación de  $B - \rho A$  y la segunda en función de los eigenvalores de  $B$  y  $A$  [8]. La demostración de cada una requiere del conocimiento del concepto de inercia y del Teorema de la Inercia de Sylvester. Por este motivo se presentan a continuación: <sup>4</sup>

**Definición 1.1.** *La inercia de una matriz simétrica  $A$  es la tripleta de enteros no negativos  $(m, z, p)$  donde  $m$ ,  $z$  y  $p$  son respectivamente el número de eigenvalores negativos, cero y positivos de  $A$  [4].*

**Teorema 1.2.** *Sea  $A \in \mathbb{R}^{n \times n}$  una matriz simétrica y  $Z \in \mathbb{R}^{n \times n}$  no singular. Entonces  $A$  y  $Z^T A Z$  tienen la misma inercia [4].*

**Proposición 1.2.** *La raíz  $\rho^*$  de  $f(\rho)$  está localizada en el intervalo  $(\lambda_p, \lambda_1)$  donde  $\lambda_p$  es el  $p$ -ésimo eigenvalor más grande de  $Z^T(A - \rho B)Z$ .*

*Demostración.* Sea  $Z$  la matriz que diagonaliza a  $A - \rho B$  de manera que<sup>5</sup>:

$$\begin{aligned} Z^T A Z &= \Lambda \\ Z^T B Z &= I \end{aligned} \tag{1.38}$$

Con  $\Lambda$  una matriz diagonal y  $\mu_1, \dots, \mu_n$  sus respectivos eigenvalores e  $I$  la identidad de tamaño  $n$ . Entonces por el teorema 1.2 se sabe que  $A - \rho B$  y  $Z^T(A - \rho B)Z = \Lambda - \rho I$  tienen el mismo número de eigenvalores positivos, negativos y cero. Por lo tanto, la matriz en cuestión es de la siguiente forma:

<sup>4</sup>La demostración de este teorema puede ser encontrada en [4].

<sup>5</sup>El cálculo de esta matriz puede obtenerse en el algoritmo 8.7.1 de [4]

$$\Lambda - \rho I = \begin{pmatrix} \mu_1 & 0 & \dots & 0 \\ 0 & \mu_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mu_n \end{pmatrix} - \rho \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad (1.39)$$

Como la matriz  $V$  de (1.9) es de tamaño  $n \times p$ , solo nos interesa saber el signo de los  $p$  eigenvalores más grandes. Tomando  $\rho = \mu_p$  entonces los elementos de la diagonal de la matriz  $\Lambda - \rho I$  son de la forma:

$$\begin{aligned} \mu_i - \mu_p &\geq 0 \quad \text{para } i \geq p \\ \mu_i - \mu_p &\leq 0 \quad \text{para } i \leq p \end{aligned} \quad (1.40)$$

Los primeros  $p$  elementos tienen la propiedad de ser no negativos, ya que  $\lambda_{(A-\rho B)_1} \geq \lambda_{(A-\rho B)_2} \geq \dots \geq \lambda_{(A-\rho B)_p}$ . Usando el teorema 1.2 se sabe que los primeros  $p$  eigenvalores de  $A - \rho B$  también son no negativos. Por ende la suma de ellos es mayor o igual que cero.

Por otro lado si se toma  $\rho = \mu_1$ . Entonces los elementos de la diagonal de la matriz (1.38) son de la forma:

$$\mu_i - \mu_1 \leq 0 \quad \forall \quad i \quad (1.41)$$

Con  $i = 1, \dots, p$ , cada uno de los elementos de la diagonal tiene la propiedad de ser no positivo por el mismo argumento que el caso pasado. Por lo tanto los  $p$  eigenvalores más grandes de  $\Lambda - \rho I$  y de  $A - \rho B$  son no positivos, por lo que su suma es menor o igual que cero:

$$\rho = \mu_p \Rightarrow \sum_{i=1}^p (\mu_i - \mu_p) \geq 0 \Rightarrow f(\rho) \geq 0 \quad (1.42)$$

$$\rho = \mu_1 \Rightarrow \sum_{i=1}^p (\mu_i - \mu_1) \leq 0 \Rightarrow f(\rho) \leq 0 \quad (1.43)$$

□

## CAPÍTULO 1: DISCRIMINANTE LINEAL DE FISHER

**Ejemplo 1.2.** Tomando las matrices  $A, B$  iguales que en el ejercicio 1.1, se puede encontrar fácilmente a la matriz  $Z$ :

$$Z = \begin{pmatrix} \sqrt{(\frac{1}{1.5})} & 0 & 0 \\ 0 & \sqrt{(\frac{1}{2.5})} & 0 \\ 0 & 0 & \sqrt{(\frac{1}{5})} \end{pmatrix}$$

Con la matriz  $Z$  definida de esta manera,  $Z^T A Z = \Lambda$  y  $Z^T B Z = I$  toman la siguiente forma:

$$\Lambda - \rho I = \begin{pmatrix} \frac{4}{1.5} & 0 & 0 \\ 0 & \frac{6}{2.5} & 0 \\ 0 & 0 & \frac{8}{5} \end{pmatrix} - \rho \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Entonces se puede encontrar un intervalo tal que  $\rho^* \in [\rho_1, \rho_2]$ :

$$\rho_1 = \mu_p$$

$$\rho_2 = \mu_1$$

Conforme el tamaño de la dimensión a proyectar cambia, las cotas son las siguientes:

$$\begin{aligned} p=1 &\Rightarrow \quad \rho_1 = \frac{4}{1.5} \quad y \quad \rho_2 = \frac{4}{1.5} \\ p=2 &\Rightarrow \quad \rho_1 = \frac{4}{1.5} \quad y \quad \rho_2 = \frac{6}{2.5} \\ p=3 &\Rightarrow \quad \rho_1 = \frac{4}{1.5} \quad y \quad \rho_2 = \frac{8}{5} \end{aligned}$$

Estas cotas se pueden ver más fácil en la figura 1.6, donde se observa  $f(\rho)$  con respecto a  $p$ :

Otro intervalo que se ha desarrollado tiene que ver con directamente con los eigenvalores de  $A$  y  $B$  en lugar de los obtenidos por la matriz  $A - \rho B$ :

**Proposición 1.3.** *Sea  $B$  positiva definida, entonces la raíz  $\rho^*$  de  $f(\rho)$  es tal que [8]:*

$$\frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_i}} \leq \rho^* \leq \frac{\sum_{i=1}^p \lambda_{(A)_i}}{\sum_{i=1}^p \lambda_{(B)_{n-i+1}}}$$

con  $\lambda_{A_i}$  y  $\lambda_{B_i}$  el  $i$ -ésimo eigenvalor más grande de la matriz  $A$  y  $B$  respectivamente [8].

*Demostración.* Se tiene la propiedad que para una  $p$  dada:

$$\max_{V^T V = I} \text{Tr}(V^T A V) = \text{Tr}(V^{T*} A V^*) = \sum_{i=1}^p \lambda_{A_i} \quad (1.44)$$

Con  $\lambda_{A_i}$  los eigenvalores de  $A$ . Como esta  $V^*$  maximiza la traza sobre  $A$ , entonces no necesariamente maximiza la de  $B$ . Al sustituirla en  $\text{Tr}(V^T B V)$  se tiene que:

$$\text{Tr}(V^{T*} B V^*) \leq \sum_{i=1}^p \lambda_{B_i} \quad (1.45)$$

Con  $\lambda_{B_i}$  los eigenvalores de  $B$ . Al hacer el cociente de (1.43) y (1.44) Se puede acotar inferiormente a  $\rho^*$ :

$$\frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_i}} \leq \frac{\text{Tr}(V^{T*} A V^*)}{\text{Tr}(V^{T*} B V^*)} \leq \max_{V^T V = I} \frac{\text{Tr}(V^T A V)}{\text{Tr}(V^T B V)} = \rho^*$$

Ahora falta acotarlo superiormente. Usando las siguientes propiedades que son derivadas de (1.27):

$$\text{Tr}(V^T A V) \leq \sum_{i=1}^p \lambda_{A_i} \quad (1.46)$$

$$\text{Tr}(V^T B V) \geq \sum_{i=1}^p \lambda_{B_{(n-i+1)}} \quad (1.47)$$

La expresión (1.46) es la suma de los  $p$  eigenvalores más chicos de  $B$ . Dividiendo (1.45) entre (1.46) se tiene que para cualquier matriz ortogonal  $V$ :

$$\frac{\text{Tr}(V^T AV)}{\text{Tr}(V^T BV)} \leq \frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_{(n-i+1)}}} \quad (1.48)$$

En particular si se toma  $V = V^{**}$  (La matriz con la que se alcanza  $\rho^*$ ):

$$\rho^* = \frac{\text{Tr}(V^{T**} AV^{**})}{\text{Tr}(V^{T**} BV^{**})} \leq \frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_{(n-i+1)}}} \quad (1.49)$$

□

**Ejemplo 1.3.** Para ejemplificar esta cota se usará las matrices  $A$  y  $B$  de los dos ejemplos anteriores y se muestra en la figura 1.7.

$$A = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 8 \end{pmatrix}, \quad B = \begin{pmatrix} 1.5 & 0 & 0 \\ 0 & 2.5 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

(i) Para  $p = 1$  la cota es la siguiente:

$$\lambda_{A_1} = 8 \quad \lambda_{B_1} = 5 \quad \lambda_{B_3} = 1.5$$

$$\rho_1 = \frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_i}} = \frac{8}{5} \quad \rho_2 = \frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_{n-i+1}}} = \frac{8}{1.5}$$

(ii) Para  $p = 2$  la cota es la siguiente:

$$\sum_{i=1}^2 \lambda_{A_i} = 14 \quad \sum_{i=1}^2 \lambda_{B_i} = 7.5 \quad \sum_{i=1}^2 \lambda_{B_{3-i+1}} = 4$$

$$\rho_1 = \frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_i}} = \frac{14}{7.5} \quad \rho_2 = \frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_{n-i+1}}} = \frac{14}{4}$$

(iii) Para  $p = 3$  la cota es la siguiente:

$$\sum_{i=1}^3 \lambda_{A_i} = 18 \quad \sum_{i=1}^3 \lambda_{B_i} = 9 \quad \sum_{i=1}^3 \lambda_{B_{3-i+1}} = 9$$

$$\rho_1 = \frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_i}} = \frac{18}{9} \quad \rho_2 = \frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_{n-i+1}}} = \frac{18}{9}$$

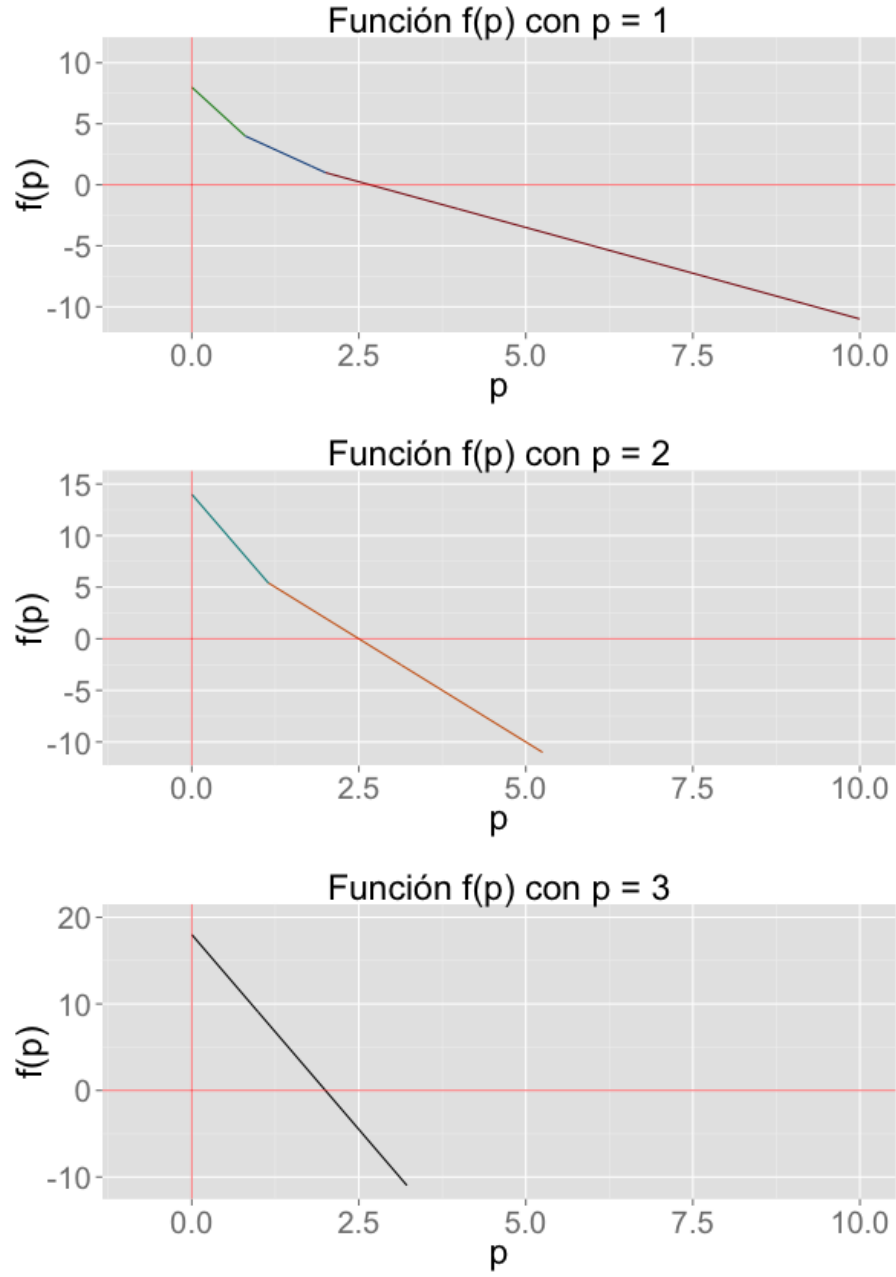


Figura 1.6: La figura superior representa a  $f(\rho) = \lambda_{G(\rho)_1}$ , la de enmedio  $f(\rho) = \lambda_{G(\rho)_1} + \lambda_{G(\rho)_2}$  y la de abajo  $f(\rho) = \lambda_{G(\rho)_1} + \lambda_{G(\rho)_2} + \lambda_{G(\rho)_3}$ .



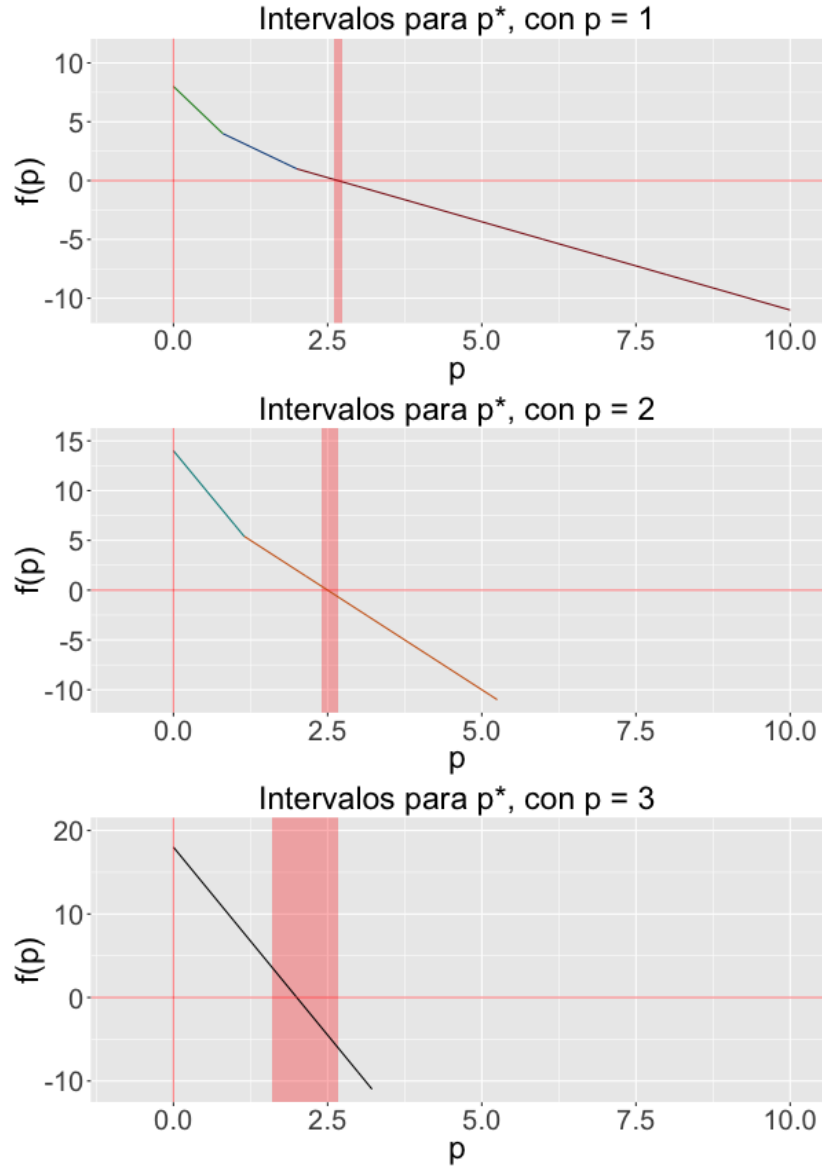


Figura 1.7: La figura superior representa intervalos para  $\rho^*$  cuando  $f(\rho) = \lambda_{G(\rho)_1}$ , la de enmedio cuando  $f(\rho) = \lambda_{G(\rho)_1} + \lambda_{G(\rho)_2}$  y la de abajo cuando  $f(\rho) = \lambda_{G(\rho)_1} + \lambda_{G(\rho)_2} + \lambda_{G(\rho)_3}$ .

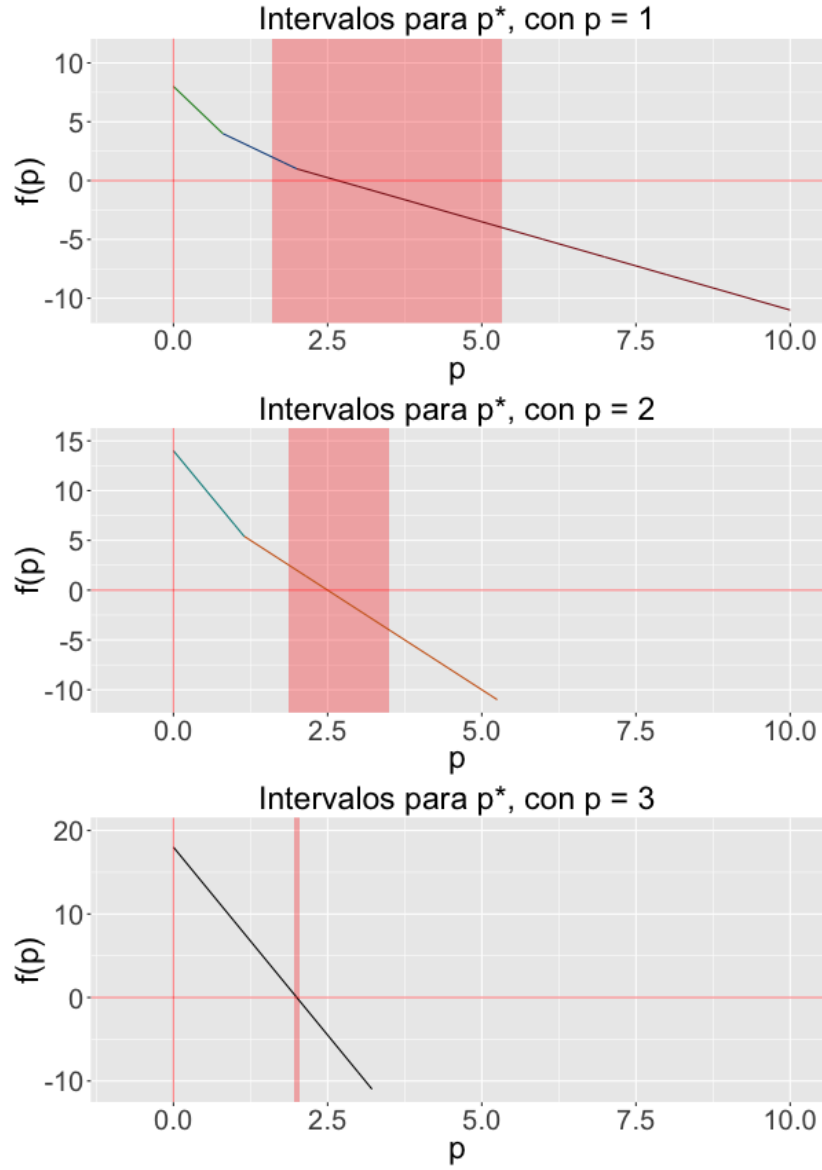


Figura 1.8: La figura superior representa intervalos para  $\rho^*$  cuando  $f(\rho) = \lambda_{G(\rho)_1}$ , la de enmedio cuando  $f(\rho) = \lambda_{G(\rho)_1} + \lambda_{G(\rho)_2}$  y la de abajo cuando  $f(\rho) = \lambda_{G(\rho)_1} + \lambda_{G(\rho)_2} + \lambda_{G(\rho)_3}$ .

## Capítulo 2

# El Método Newton-Lanczos

En el capítulo anterior se propuso una función no creciente  $f(\rho)$  cuya raíz resulta ser la solución óptima para el problema del Discriminante Lineal de Fisher:

$$f(\rho) = \max_{V^T V = I} \text{Tr}(V^T (S_E - \rho S_I) V) \quad (2.1)$$

El algoritmo propuesto para encontrar la solución recibe el nombre de Newton-Lanczos [8]. Para entenderlo a profundidad, se explicarán brevemente los métodos de Lanczos que tridiagonalizan una matriz simétrica, para después calcular eficientemente los primeros (últimos) eigenvalores. Después, será implementado junto al método iterativo de Newton, que calcula el nuevo valor de  $\rho_n$  para cada paso. Para esto, se requiere el cómputo de la derivada de  $f(\rho)$ , por lo que se desarrollará su forma analítica. Finalizando, se proporcionarán las condiciones necesarias de optimalidad.

La primera división de los métodos para calcular eigenvectores y eigenvalores que propone J. Demmel [1] depende si la matriz es simétrica o no lo es. Después hace una sub-clasificación dependiendo si el método es iterativo o directo. Para este texto solo se calcularán los eigenvalores de matrices simétricas, por lo que el procedimiento a seguir será el siguiente:

- Tridiagonalizar la matriz simétrica por un método iterativo
- Encontrar los eigenvalores por medio de la iteración tridiagonal QR (LA-

PACK DSTEVD)

## 2.1. Métodos de Lanczos

Antes de presentar los métodos de Lanczos, se dará una breve introducción acerca del costo computacional del algoritmo QR y la importancia de tridiagonalizar la matriz [1]. Sea  $A \in \mathbb{R}^{n \times n}$ , entonces su descomposición  $QR$  toma  $O(n^3)$  flops. Suponiendo el mejor escenario en el que cada eigenvalor se encuentra con una iteración, tomaría  $O(n^4)$  flops para calcular todos los eigenvalores de una matriz. Por otra parte, al tridiagonalizar la matriz  $A$  se reduce el costo computacional de la descomposición  $QR$  a  $O(n^2)$  flops. De esta manera, tomaría  $O(n^3)$  flops encontrar todos los eigenvalores. Una descripción más detallada del costo computacional de los algoritmos puede encontrarse en [1].

El primer paso, la tridiagonalización, ha sido muy estudiado y existen algoritmos especializados para distintos tipos de matrices simétricas. Si la matriz es de gran dimensión y rala, entonces se recomienda usar el método de Lanczos [4]. En otro caso, existen las transformaciones Householder y las rotaciones de Givens [4]. El método de Lanczos realiza tridiagonalizaciones parciales de la matriz original  $A$ , donde cada una es de tamaño  $p \times p$  con  $p \leq n$ . Un aspecto interesante es que las matrices parciales van aproximando los eigenvectores extremos antes que la tridiagonalización esté completa. Por este motivo, el algoritmo es usado cuando se requieren solo algunos de los eigenvalores. Con respecto al costo computacional, es del orden  $O(n^3)$ , por lo que el algoritmo completo se mantiene en el mismo orden.

Lanczos en aritmética exacta tiene muchas ventajas computacionales y converge rápidamente a los eigenvalores reales, pero con aritmética inexacta es difícil usarlo en la práctica [4]. El problema que se presenta es que los eigenvectores van perdiendo la ortogonalidad entre ellos conforme la dimensionalidad y las iteraciones incrementan. Los textos [1] [4] incluyen algoritmos para solucionar este problema.

En la siguiente subsección se presentará el algoritmo original de Lanczos; se ejemplificará como los eigenvalores de la matriz tridiagonal convergen a los originales y se presenta el problema de *ghost eigenvalues* [1], que son los eigenvalores

que surgen al perder la ortogonalidad en aritmética inexacta.

### 2.1.1. Algoritmo de Lanczos

El algoritmo de Lanczos busca calcular los elementos de esta matriz tridiagonal directamente. Definiendo  $Q^T A Q = T$ , con  $Q = [q_{(1)} \mid q_{(2)} \mid \dots \mid q_{(n)}]$  ortogonal, y  $T_n$  tridiagonal igual a:

$$T_n = \begin{pmatrix} \alpha_1 & \beta_2 & & & & 0 \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \ddots & & \\ & & \ddots & \ddots & \beta_{n-1} & \\ & & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ 0 & & & & \beta_n & \alpha_n \end{pmatrix}, \quad (2.2)$$

Como  $q_{(1)}$  es una columna de  $Q$ , entonces  $q_1^T = [q_{11}, q_{21}, \dots, q_{n1}]$ . De esta manera, se tiene que  $AQ = QT$ . Descomponiendo la multiplicación  $AQ = [Aq_{(1)} \mid Aq_{(2)} \mid \dots \mid Aq_{(n)}]$  e igualándola a cada columna de  $QT$ , se tiene que  $Aq_{(1)}$  [4]:

$$\begin{aligned} Aq_{(1)} &= q_{(1)(1)}\alpha_{(1)} + q_{(1)(2)}\beta_{(2)} + \\ &\quad q_{(2)(1)}\alpha_{(1)} + q_{(2)(2)}\beta_{(2)} + \\ &\quad \vdots \\ &\quad q_{(n)(1)}\alpha_{(1)} + q_{(n)(2)}\beta_{(2)} \\ Aq_{(1)} &= \alpha_{(1)}q_{(1)} + \beta_{(2)}q_{(2)} \end{aligned}$$

Ahora, para  $i = 2, \dots, n-1$

$$\begin{aligned} Aq_{(i)} &= q_{(i-1)(i-1)}\beta_{(i)} + q_{(i-1)(i)}\alpha_{(i)} + q_{(i-1)(i+1)}\beta_{(i+1)} + \\ &\quad q_{(i)(i-1)}\beta_{(i)} + q_{(i)(i)}\alpha_{(i)} + q_{(i)(i+1)}\beta_{(i+1)} + \\ &\quad \vdots \\ &\quad q_{(n)(i-1)}\beta_{(i)} + q_{(n)(i)}\alpha_{(i)} + q_{(n)(i+1)}\beta_{(i+1)} \\ Aq_{(i)} &= \beta_{(i)}q_{(i-1)} + \alpha_{(i)}q_{(i)} + \beta_{(i+1)}q_{(i+1)} \end{aligned}$$

## CAPÍTULO 2: EL MÉTODO NEWTON-LANZOS

Por último, para  $Aq_n$ :

$$\begin{aligned}
 Aq_{(n)} &= \begin{matrix} q_{(1)(n)}\alpha_{(n)} & + & q_{(1)(n-1)}\beta_{(n)} & + \\ q_{(2)(n)}\alpha_{(n)} & + & q_{(2)(n-1)}\beta_{(n)} & + \\ \vdots & & \vdots & \\ q_{(n)(n)}\alpha_{(n)} & + & q_{(n)(n-1)}\beta_{(n)} \end{matrix} \\
 Aq_{(n)} &= \alpha_{(n)}q_{(n)} + \beta_{(n)}q_{(n-1)}
 \end{aligned}$$

Si se define  $q_{(0)} = 0$ , entonces se puede resumir el paso como:

$$Aq_{(i)} = \beta_{(i)}q_{(i-1)} + \alpha_{(i)}q_{(i)} + \beta_{(i+1)}q_{(i+1)} \quad (2.3)$$

para  $i = 1, \dots, n-1$ . Multiplicando esta expresión por  $q_{(i)}^T$ , y usando el supuesto de ortogonalidad, entonces  $q_{(i)}^T q_{(j)} = 0$  con  $i \neq j$ . De esta manera resulta la siguiente expresión:

$$q_{(i)}^T Aq_{(i)} = q_{(i)}^T \alpha_{(i)} q_{(i)} = \alpha_{(i)} \quad (2.4)$$

Por otra parte, despejando  $\beta_{(i+1)}q_{(i+1)}$  de 2.3, se tiene que

$$\begin{aligned}
 \beta_{(i+1)}q_{(i+1)} &= Aq_{(i)} - \beta_{(i)}q_{(i-1)} - \alpha_{(i)}q_{(i)} \\
 &= (A - \alpha_{(i)}I)q_{(i)} - \beta_{(i)}q_{(i-1)} = r_{(i)}
 \end{aligned} \quad (2.5)$$

Con la ecuación 2.5,  $q_{(i+1)} = \frac{r_{(i)}}{\beta_{(i+1)}}$ . Calculando la norma de  $r_{(i)}$ , se tiene que

$$\begin{aligned}
 \|r_{(i)}\|_2 &= |\beta_{(i+1)}| \|q_{(i+1)}\|_2 \\
 &= |\beta_{(i+1)}|
 \end{aligned} \quad (2.6)$$

Cuando  $r_k = 0$  entonces la iteración se detiene. [4]

### Implementación en aritmética exacta

Sea  $A \in \mathbb{R}^{n \times n}$  una matriz simétrica y  $q_i \in \mathbb{R}^n$ . Entonces el algoritmo que se presenta a continuación produce una matriz  $T_k \in \mathbb{R}^{k \times k}$  tridiagonal tal que los eigenvalores de  $T_k$  convergen a los de la matriz original  $A$  [4].

```

 $q_0 \leftarrow 0;$ 
 $r_0 \leftarrow$  vector aleatorio;
 $\beta_1 \leftarrow \|r_0\|_2;$ 
 $q_1 \leftarrow r_0 / \beta_1;$ 
 $\alpha_1 \leftarrow q_1^T A q_1;$ 
 $eps = 0.0000001;$ 
 $k = 1 ;$ 
while ( $\beta_k > eps$ ) do
     $r_k \leftarrow A q_k - \alpha_k q_k - \beta_k q_{k-1};$ 
     $\beta_{k+1} \leftarrow \|r_k\|_2;$ 
     $q_{k+1} \leftarrow r_k / \beta_{k+1};$ 
     $\alpha_{k+1} \leftarrow q_{k+1}^T A q_{k+1};$ 
     $k \leftarrow k + 1;$ 
end

```

#### Algorithm 1: Algoritmo de Lanczos

Para hacer frente a la pérdida de ortogonalidad entre los vectores, se han creado distintos métodos, los cuales se basan en la reortogonalización de la base de vectores. [1] Esta puede realizarse en  $O(n^2)$  operaciones, por lo que no afecta el orden de  $O(n^3)$  flops.

## 2.2. Derivada de $f(\rho)$

La fórmula analítica de la derivada de  $f(\rho)$  se puede obtener con cálculo multivariado. Para encontrarla, sea  $V(\rho) \in \mathbb{R}^{n \times p}$  una función diferenciable con respecto a  $\rho$ . Esta cumple la característica de ser una matriz ortogonal con columnas:

$$V(\rho) = (v_1(\rho) \mid v_2(\rho) \mid \dots \mid v_p(\rho))$$

## CAPÍTULO 2: EL MÉTODO NEWTON-LANCZOS

Antes de calcular la derivada de  $f(\rho)$ , es conveniente examinar la derivada de  $V(\rho)^T V(\rho)$  con respecto a  $\rho$ . Sea  $V(\rho)$  una matriz ortogonal; es decir, que cumpla  $V^T(\rho)V(\rho) = I$ , entonces [8]:

$$\frac{d}{d\rho} V(\rho)^T V(\rho) = \left( \frac{d}{d\rho} V(\rho)^T \right) V(\rho) + V(\rho)^T \left( \frac{d}{d\rho} V(\rho) \right)$$

La derivada de  $V(\rho)$  no se conoce explícitamente, pero se puede derivar componente a componente. De esta manera:

$$\begin{aligned} V(\rho)^T V(\rho) &= \begin{pmatrix} v_{11}(\rho) & v_{21}(\rho) & \dots & v_{n1}(\rho) \\ v_{12}(\rho) & v_{22}(\rho) & \dots & v_{n2}(\rho) \\ \vdots & \vdots & \ddots & \vdots \\ v_{1p}(\rho) & v_{2p}(\rho) & \dots & v_{np}(\rho) \end{pmatrix} \begin{pmatrix} v_{11}(\rho) & v_{12}(\rho) & \dots & v_{1p}(\rho) \\ v_{21}(\rho) & v_{22}(\rho) & \dots & v_{2p}(\rho) \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1}(\rho) & v_{n2}(\rho) & \dots & v_{np}(\rho) \end{pmatrix} \\ &= \begin{pmatrix} v_1(\rho)^T \\ v_2(\rho)^T \\ \vdots \\ v_p(\rho)^T \end{pmatrix} \begin{pmatrix} v_1(\rho) & v_2(\rho) & \dots & v_p(\rho) \end{pmatrix} \end{aligned}$$

Entonces la entrada  $(i, j)$  de  $V(\rho)^T V(\rho)$  es:

$$[V(\rho)^T V(\rho)]_{ij} = v_i(\rho)^T v_j(\rho)$$

Calculando la derivada:

$$\frac{d}{d\rho} [V(\rho)^T V(\rho)]_{ij} = \left( \frac{d}{d\rho} v_i(\rho)^T \right) v_j(\rho) + v_i(\rho)^T \left( \frac{d}{d\rho} v_j(\rho) \right)$$

En específico para el caso  $i = j$ :

$$\frac{d}{d\rho} [V(\rho)^T V(\rho)]_{ii} = 2 \left( \frac{d}{d\rho} v_i(\rho)^T \right) v_i(\rho) \quad (2.7)$$



## CAPÍTULO 2: EL MÉTODO NEWTON-LANCZOS

**Lema 2.1.** Sea  $V(\rho) \in \mathbb{R}^{n \times p}$  una matriz ortogonal y  $\rho$  su parámetro. Entonces [8]:

$$\begin{aligned} (i) \quad & \frac{d}{d\rho} V(\rho)^T V(\rho) = \left( \frac{d}{d\rho} V(\rho)^T \right) V(\rho) + V(\rho)^T \left( \frac{d}{d\rho} V(\rho) \right) = 0 \\ (ii) \quad & \text{Diag} \left( \left( \frac{d}{d\rho} V^T \right) V(\rho) \right) = 0 \end{aligned}$$

*Demostración.* (i) Para demostrar la primer propiedad se hace uso de que  $V(\rho)^T V(\rho) = I_p$ , entonces:

$$\frac{d}{d\rho} [V(\rho)^T V(\rho)] = 0$$

(ii) Se parte de la ecuación (2.7), y se usa el punto (i):

$$\frac{d}{d\rho} [V(\rho)^T V(\rho)]_{ii} = 2 \left( \frac{d}{d\rho} v_i(\rho)^T \right) v_i(\rho) = 0$$

Como cada elemento  $i$  es cero, entonces en particular  $\text{Diag} \left( \left( \frac{d}{d\rho} V(\rho)^T \right) V(\rho) \right) = 0$ . □

Del lema 2.1 se tiene que  $\left( \frac{d}{d\rho} V(\rho)^T \right) V(\rho)$  tiene diagonal igual a 0. Ahora, para derivar  $f(\rho)$ , primero se deriva la expresión  $\frac{d}{d\rho} [V^T(A - \rho B)V]$ :

$$\begin{aligned} \frac{d}{d\rho} [V^T(A - \rho B)V] &= \frac{d}{d\rho} [V^T A V] - \frac{d}{d\rho} [V^T \rho B V] \\ &= \frac{dV^T}{d\rho} A V + V^T A \frac{dV}{d\rho} - \frac{dV^T}{d\rho} \rho B V - V^T \left[ B V + \rho B \left( \frac{dV}{d\rho} \right) \right] \\ &= \frac{dV^T}{d\rho} [A - \rho B] V + V^T [A - \rho B] \frac{dV}{d\rho} - V^T B V \end{aligned} \tag{2.8}$$

Sea  $V$  la matriz que diagonaliza  $(A - \rho B)$ , de manera que  $V^T(A - \rho B)V = D$ . Entonces 2.8:

$$\frac{d}{d\rho} [V^T(A - \rho B)V] = \frac{dV^T}{d\rho} V D + D V^T \frac{dV}{d\rho} - V^T B V \tag{2.9}$$

## CAPÍTULO 2: EL MÉTODO NEWTON-LANCZOS

Al calcular la traza de (2.9) y usando del lema 2.1, se tiene que:

$$\begin{aligned}
 \text{Tr} \left[ \frac{d}{d\rho} [V^T(A - \rho B)V] \right] &= \text{Tr} \left[ \frac{dV^T}{d\rho} V D + D V^T \frac{dV}{d\rho} - V^T B V \right] \\
 &= 2\text{Tr} \left[ D V^T \frac{dV}{d\rho} \right] - \text{Tr} [V^T B V] \\
 &= -\text{Tr} [V^T B V]
 \end{aligned} \tag{2.10}$$

### 2.3. Método Newton-Lanczos

El método de Newton establece que la iteración está dada por:

$$x_{(n+1)} = x_{(n)} - \frac{f(x_n)}{f'(x_n)}$$

con  $f'(x_n)$  la derivada de  $f(x_n)$ .

Con esta fórmula y con (2.10), se puede calcular explícitamente  $\rho_{n+1}$  para la iteración de Lanczos [8]:

$$\begin{aligned}
 \rho_{n+1} &= \rho_n - \frac{\text{Tr} [V^T(A - \rho_n B)V]}{-\text{Tr} [V^T B V]} \\
 &= \frac{\rho_n \text{Tr} [V^T B V] + \text{Tr} [V^T(A - \rho_n B)V]}{\text{Tr} [V^T B V]} \\
 &= \frac{\rho_n \text{Tr} [V^T B V] + \text{Tr} [V^T A V] - \rho_n \text{Tr} [V^T B V]}{\text{Tr} [V^T B V]} \\
 &= \frac{\text{Tr} [V^T A V]}{\text{Tr} [V^T B V]}
 \end{aligned} \tag{2.11}$$

Una vez calculado el paso de la iteración, se enuncia el método de Newton-Lanczos para maximizar el cociente de trazas. El método recibe como entrada la matriz  $A$ ,  $B$  y la dimensión a la cual se desea proyectar  $p$  [8]:

## CAPÍTULO 2: EL MÉTODO NEWTON-LANCZOS

```

i = 1;
ρ1 un número aleatorio;
ρ0;
f(ρ1) = ∑j=1p λ(A-ρ1B)j;
V = primeros p eigenvectores de A - ρ1B;
tol = 1e - 10;
while (i < 50 , abs(ρi - ρi-1) > tol) do
    i = i + 1;
    V = primeros p eigenvectores de (A - ρiB) ;
    ρi = Tr(VTAV)/Tr(VTBV);
    f(ρi) = ∑j=1p λ(A-ρiB)j;
end

```

**Algorithm 2:** Algoritmo de Newton-Lanczos

### 2.3.1. Condiciones necesarias de optimalidad

Considerando el problema de maximizar las trazas:

$$\max_{\substack{V \in \mathbb{R}^{n \times p} \\ V^T V = I}} \frac{\text{Tr}(V^T S_E V)}{\text{Tr}(V^T S_I V)} \quad (2.12)$$

La función lagrangiana asociada a este problema es la siguiente [8]:

$$L(V, \Gamma) = \frac{\text{Tr}(V^T S_E V)}{\text{Tr}(V^T S_I V)} - \text{Tr}[\Gamma(V^T V - I)]$$

Con  $\Gamma \in \mathbb{R}^{p \times p}$ , la matriz de multiplicadores de Lagrange de la forma:

$$\Gamma = \begin{pmatrix} \gamma_{(1)(1)} & \gamma_{(1)(2)} & \cdots & \gamma_{(1)(p-1)} & \gamma_{(1)(p)} \\ \gamma_{(2)(1)} & \gamma_{(2)(2)} & \cdots & \gamma_{(2)(p-1)} & \gamma_{(2)(p)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_{(p-1)(1)} & \gamma_{(p-1)(2)} & \cdots & \gamma_{(p-1)(p-1)} & \gamma_{(p-1)(p)} \\ \gamma_{(p)(1)} & \gamma_{(p)(2)} & \cdots & \gamma_{(p)(p-1)} & \gamma_{(p)(p)} \end{pmatrix},$$

## CAPÍTULO 2: EL MÉTODO NEWTON-LANCZOS

Ya que al multiplicar esta matriz por  $[V^T V - I]$  (con  $V = (v_1 \mid v_2 \mid \dots \mid v_{p-1} \mid v_p)$ ) se tiene que:

$$V^T V - I = \begin{pmatrix} v_1^T v_1 - 1 & v_1^T v_2 & \dots & v_1^T v_{p-1} & v_1^T v_p \\ v_2^T v_1 & v_2^T v_2 - 1 & \dots & v_2^T v_{p-1} & v_2^T v_p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{p-1}^T v_1 & v_{p-1}^T v_2 & \dots & v_{p-1}^T v_{p-1} - 1 & v_{p-1}^T v_p \\ v_p^T v_1 & v_p^T v_2 & \dots & v_p^T v_{p-1} & v_p^T v_p - 1 \end{pmatrix},$$

entonces, la diagonal de  $(\Gamma(V^T V - I))$  contiene  $p$  elementos que son los siguientes:

$$\begin{pmatrix} \gamma_{(1)(1)}(v_1^T v_1 - 1) & +\gamma_{(1)(2)}(v_2^T v_1) & +\dots & +\gamma_{(1)(p)}(v_p^T v_1) \\ \gamma_{(2)(1)}(v_1^T v_2) & +\gamma_{(2)(2)}(v_2^T v_2 - 1) & +\dots & +\gamma_{(2)(p)}(v_p^T v_2) \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{(p-1)(1)}(v_1^T v_{p-1}) & +\gamma_{(p-1)(2)}(v_2^T v_{p-1}) & +\dots & +\gamma_{(p-1)(p)}(v_p^T v_{p-1}) \\ \gamma_{(p)(1)}(v_1^T v_p) & +\gamma_{(p)(2)}(v_2^T v_p) & +\dots & +\gamma_{(p)(p)}(v_p^T v_p - 1) \end{pmatrix},$$

Para calcular la traza se suman estos elementos de la diagonal. De esta forma, se tienen  $p$  restricciones de la forma  $v_i^T v_i = 1 \quad i = 1, \dots, p$  y  $p(p-1)$  restricciones de la forma  $v_i^T v_j = 0 \quad i \neq j, \quad i, j = 1, \dots, p$ , cada una con su multiplicador lagrangiano.

Como la ecuación a maximizar tiene un maximizador global  $V^*$ , entonces existe un multiplicador matricial lagrangiano tal que en el óptimo:

$$\frac{\partial \mathcal{L}(V^*, \Gamma^*)}{\partial V} = 0 \quad \text{con} \quad V^{T*} V^* = I \quad (2.13)$$

Para encontrar  $(V^*, \Gamma^*)$  de (2.13), primero se debe conocer la derivada de  $\frac{\partial \text{Tr}(V^T M V)}{\partial V}$ , con  $M$  cualquier matriz:

$$\frac{\partial \text{Tr}(V^T M V)}{\partial V} = (M^* + M)V \quad (2.14)$$

## CAPÍTULO 2: EL MÉTODO NEWTON-LANCZOS

Derivando el lagrangiano (2.13) y utilizando (2.14) se tiene que:

$$\begin{aligned} \frac{\partial \mathcal{L}(V, \Gamma)}{\partial V} = & \left[ \frac{\partial \text{Tr}(V^T AV)}{\partial V} \right] [\text{Tr}(V^T BV)^{-1}] + \\ & [\text{Tr}(V^T AV)] \left[ \frac{\partial (\text{Tr}(V^T BV))^{-1}}{\partial V} \right] - \\ & \left[ \frac{\partial \text{Tr}[\Gamma(V^T V - I)]}{\partial V} \right] \end{aligned}$$

$$\begin{aligned} \frac{\partial \mathcal{L}(V, \Gamma)}{\partial V} = & [2AV] [\text{Tr}(V^T BV)^{-1}] + \\ & [\text{Tr}(V^T AV)] \left[ \frac{2BV}{\text{Tr}(V^T BV)^2} \right] - \\ & [V(\Gamma^* + \Gamma)] \end{aligned}$$

$$\begin{aligned} \frac{\partial \mathcal{L}(V, \Gamma)}{\partial V} = & \left[ \frac{2AV\text{Tr}(V^T BV) - 2BV\text{Tr}(V^T AV)}{(\text{Tr}(V^T BV))^2} \right] + \\ & [V(\Gamma^* + \Gamma)] \end{aligned} \quad (2.15)$$

Igualando (2.15) a cero y acomodando la expresión, resulta:

$$\begin{aligned} \left[ A - \frac{\text{Tr}(V^{T*} AV^*)}{\text{Tr}(V^{T*} BV^*)} B \right] V^* &= \left[ \frac{\text{Tr}(V^{T*} BV^*)}{2} V^* (\Gamma^{T*} + \Gamma^*) \right] \\ [A - \rho^* B] V^* &= \left[ \frac{\text{Tr}(V^{T*} BV^*)}{2} V^* (\Gamma^{T*} + \Gamma^*) \right] \end{aligned} \quad (2.16)$$

Con  $\rho^* = \frac{\text{Tr}(V^{T*} AV^*)}{\text{Tr}(V^{T*} BV^*)}$ . Sea  $Q$  la matriz ortogonal que diagonaliza  $\Gamma^{T*} + \Gamma^*$ , entonces se puede escribir la expresión  $(\Gamma^{T*} + \Gamma^*)$  como:

$$(\Gamma^{T*} + \Gamma^*) = Q \Sigma^* Q^T \quad \text{con} \quad Q^T Q = I$$

Con  $\Sigma^*$  una matriz diagonal. Multiplicando (2.16) por  $V^{T*}$  y calculando la traza de ambos lados de la ecuación y se tiene que:

CAPÍTULO 2: EL MÉTODO NEWTON-LANCZOS

$$Tr[V^{T*}(A - \rho^*B)V^*] = \left[ \frac{Tr(V^{T*}BV^*)}{2} \right] Tr(\Gamma^{T*} + \Gamma^*)$$

$$Tr(\Gamma^{T*} + \Gamma^*) = 2 \frac{Tr(V^{T*}(A - \rho^*B)V^*)}{Tr(V^{T*}BV^*)} = 0$$

$$Tr(Q\Sigma^*Q^T) = 2 \frac{Tr(V^{T*}(A - \rho^*B)V^*)}{Tr(V^{T*}BV^*)} = 0 \quad (2.17)$$

Como  $Tr(V^{T*}(A - \rho^*B)V^*) = 0$  entonces  $Tr(Q\Sigma^*Q^T) = 0$ , por lo que  $Tr(\Sigma^*) = 0$ . Definiendo  $U^* = V^*Q$ , con  $U^*U = I$ , la expresión (2.16) se puede escribir como:

$$\begin{aligned} [A - \rho^*B]V^* &= \left[ \frac{Tr(V^{T*}BV^*)}{2} V^*(Q\Sigma^*Q^T) \right] \\ &= \left[ \frac{Tr(V^{T*}BV^*)}{2} U^*Q^{T*}(Q\Sigma^*Q^T)Q \right] \\ &= \left[ \frac{Tr(V^{T*}BV^*)}{2} U^*\Sigma^* \right] \end{aligned}$$

Definiendo  $\Lambda_* = \frac{Tr(V^{T*}BV^*)}{2} \Sigma^*$ , entonces:

$$\begin{aligned} [A - \rho^*B]U^* &= U^*\Lambda_* \\ U^{T*}(A - \rho^*B)U^* &= \Lambda_* \\ Tr(U^{T*}(A - \rho^*B)U^*) &= Tr(\Lambda_*) = 0 \end{aligned} \quad (2.18)$$

De esta manera se tiene que la ecuación (2.18) es la condición necesaria para que  $U^*, \rho^*$  sean las óptimas

## Capítulo 3

# Experimentos numéricos

En este capítulo se muestran los resultados obtenidos de la implementación del algoritmo de Newton-Lanczos, el cual se usó para maximizar el cociente planteado por el ADLF (Análisis Discriminante Lineal de Fisher). Este método encuentra la matriz de proyección óptima que maximiza el cociente de la matriz de dispersión entre clases y la matriz de dispersión intra clase. Maximizar esta formulación tiene como resultado que la proyección mantenga a los individuos de una misma clase cercanos uno a otro además que mantiene alejados a los centroides de las distintas clases.

Como se mencionó en capítulos anteriores, el ADLF es un método de reducción dimensional previo a un algoritmo de clasificación. Esta reducción dimensional es importante en problemas de dimensionalidad alta, ya que pueden reducir el problema a uno mucho más sencillo y computacionalmente eficiente. El aumento en la eficiencia computacional se debe a que se trabaja sobre un espacio de menor dimensionalidad, por lo que las operaciones a realizar son más rápidas. Otros problemas como *The curse of dimensionality* también se ven solucionados. Esto tiene consecuencias computacionales inmediatas, en particular para métodos que requieren el cómputo exhaustivo de distancias como *k-vecinos más cercanos*. Por esto motivo, en esta tesis, se utiliza esta técnica para calcular la probabilidad de pertenencia a cada clase.

Para comparar la precisión y el tiempo de cómputo del método de ADLF implementado con el algoritmo de Newton-Lanczos en problemas de dimensiona-

### CAPÍTULO 3: EXPERIMENTOS NUMÉRICOS

lidad alta se comparará con otros dos métodos que realizan clasificación lineal: el Análisis Discriminante Lineal (ADL) y la Regresión Logística Multinomial (RLM). Estos métodos se seleccionaron debido a que tienen un fuerte fundamento teórico y han sido probados en múltiples conjuntos de datos. Para comparar estos métodos se utilizaron dos conjuntos de datos, los cuales fueron cuidadosamente seleccionados para ejemplificar las ventajas del ADLF. Una de las características buscadas en estos conjuntos de datos fue que presentaran dimensionalidad alta; es decir, un alto número de variables. La otra característica que se consideró para seleccionar la base de datos fue que clasificación lineal sea adecuada al problema.

La primer base de datos utilizada fue liberada por la compañía aseguradora *State Farm*, la cual consiste en fotografías de conductores distraídos mientras manejan. El objetivo de la compañía al liberar esta base de datos fue contar con un clasificador de individuos que permitiera asignarlo a distintos grupos de riesgo de acuerdo a sus hábitos de manejo. Para lograr esto, se debe clasificar correctamente cada imagen de acuerdo a la actividad que está realizando (hablar por telefono, cambiar estación de radio, mandar mensajes de texto, etc.). En esta imagen las variables tomadas en cuenta fueron los píxeles; es decir, cada píxel se convirtió en una variable.

La segunda base de datos utilizada fue liberada por una de las compañías más grandes de comercio electrónico del mundo: *Otto group*. Esta empresa actualmente tiene problemas para clasificar sus productos en 9 grupos dependiendo de 93 variables de las cuales no reveló la información de cada una. Como consecuencia no se puede realizar *feature extraction* ya que se desconoce la información brindada por cada variable.

El capítulo se divide en cinco partes. En la sección 3.1 se compara el tiempo de cómputo del método de Lanczos con reortogonalización completa para calcular los eigenpares de una matriz con el tiempo de la rutina *SVD* de R.<sup>1</sup> En la sección 3.2, se explica el preprocesamiento de las bases de datos y el proceso para crear los conjuntos de entrenamiento y prueba. En la sección 3.3 y 3.4 se realizan los experimentos con la bases de *State Farm* y *Otto group* respectivamente. Al final, se presentan las conclusiones de los experimentos numéricos.

---

<sup>1</sup>La rutina *svd* de *R* ocupa la función *gesdd* de la librería *LAPACK*. Para más información puede consultarse la página [http : //www.netlib.org/lapack/](http://www.netlib.org/lapack/).



Para calcular los resultados presentados en este capítulo se utilizó el lenguaje de programación R en su versión 3.3.0 *Supposedly Educational*. Los cálculos fueron realizados en una iMac 3.2 Ghz Intel Core i3 con 12 GB de RAM.

### 3.1. Comparación de tiempo entre el método de Lanczos y la factorización SVD

Como se mostró en el capítulo anterior, la parte central del algoritmo de Newton-Lanczos involucra la maximización del cociente de trazas. Para lograrlo es necesario calcular los primeros eigenpares de una resta de matrices. Estos pueden ser calculados con el método de Lanczos o con la factorización *SVD*, por lo que a continuación se mostrará una breve comparación de ambos métodos. Es importante recordar que el método de Lanczos en aritmética exacta realiza eficientemente estos cálculos, pero en aritmética inexacta requiere la reortogonalización de los eigenvectores calculados, proceso que toma tiempo.

Para ejemplificar las ventajas comparativas de los métodos se realizará un experimento numérico. La matriz utilizada para la prueba es  $A - \rho B$  con  $A$  y  $B$  la matriz de dispersión intra clase y la matriz de dispersión entre clases de la base de datos de *State Farm* respectivamente.  $\rho$  se tomo en particular con el valor de 3. Ambas matrices tienen una dimensionalidad de  $400 \times 400$ . La siguiente gráfica resume el desempeño de ambos métodos cuando se desean calcular los primeros 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 70, 80, 90, 100, 110, 120 componentes principales de la matriz en cuestión.

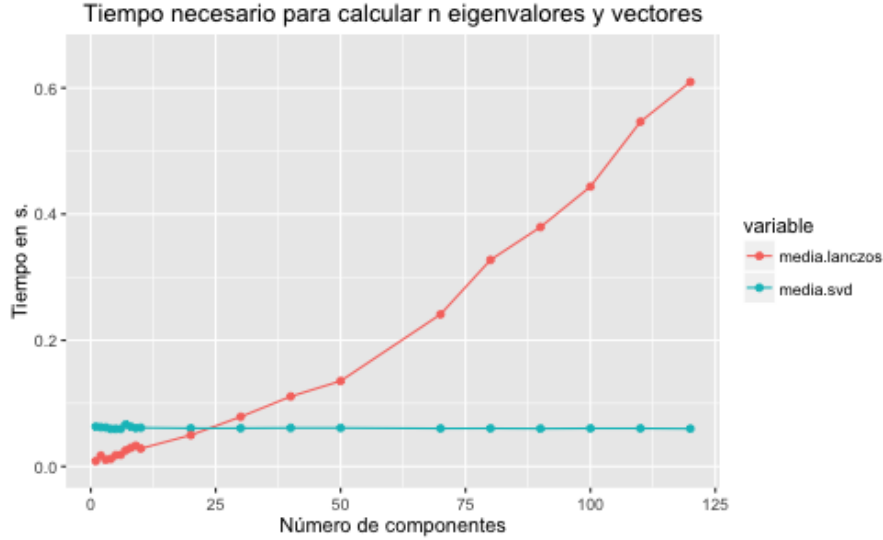


Figura 3.1: En el eje  $x$  se representa  $k$ , el número de eigenpares a calcular de la matriz  $A - \rho B$ . La matriz original es de tamaño  $400 \times 400$ . El tiempo reportado para cada  $k$ , es el promedio de 30 repeticiones para cada método.

En la figura 3.1 se observa que, para  $k$  menor a 25, el algoritmo de Lanczos es más rápido que la factorización  $SVD$  de la matriz completa. Para valores más grandes, es más eficiente factorizar toda la matriz. Como distintos textos mencionan, el algoritmo de Lanczos en aritmética inexacta presenta problemas numéricos y surge el problema de *Ghost eigenvalues* (eigenvalores que se repiten, pero que son espurios) [4]. La causa de la inestabilidad numérica es que ortogonalidad entre los eigenvectores se pierde conforme la dimensionalidad aumenta. Para solucionarlo se recomienda reortogonalizar la base de eigenvectores con métodos como reortogonalización completa y reortogonalización selectiva. [1]

Es importante mencionar que en la literatura existen comparaciones entre los dos métodos de manera que se aconseja utilizar Lanczos cuando se desea calcular solo los primeros o últimos eigenpares; en cambio, si la cantidad a calcular es alta, se recomienda encontrar todos los eigenpares con la factorización  $SVD$ . [1] [6]

### 3.2. Modelos comparados y preprocesamiento de las bases

En esta sección se comparó el método del ADLF usando Newton-Lanczos con la RLM y con el ADL. Para el método de RLM se utilizó la función *multinom* del paquete *nnet*; por otra parte, para el ADL, se utilizó una modificación de la función *lda*<sup>2</sup> del paquete *MASS*. Ambas funciones fueron desarrolladas por Brian Ripley, profesor de estadística aplicada de la Universidad de Oxford.

Como primer paso, se definieron los tamaños del conjunto de entrenamiento y el de prueba. Para el entrenamiento de *State Farm* se utilizaron 12,000 individuos (1,200 por clase); es decir, el 63.2 % del total. En el caso del ejemplo de *Otto group* también se utilizaron 9,500 datos (950 por clase); es decir, el 50 % del total de individuos. En ambos, el resto de los datos se ocupó en el conjunto de prueba.

Como segundo paso, se procede al preprocesamiento de las bases. En el ejemplo de *State Farm* cada imagen se convirtió en un vector con tamaño asociado al número de píxeles. Debido a la cantidad de imágenes y a la resolución de estas ( $640 \times 480$  píxeles), se escalaron a un tamaño de  $64 \times 48$ <sup>3</sup>. En cambio, en el caso de la base de *Otto group* solo se tienen 93 características por lo que no hay problema de dimensionalidad en lo que respecta a las variables.

Como tercer paso, solo en el caso de la base de *State Farm*, se decidió realizar reducción dimensional del conjunto de entrenamiento por medio de componentes principales, con esto se redujo el número de columnas de 3,072 ( $64 \times 48$ ) a 400. Después se proyectó al conjunto de prueba con la matriz de cargas. Esta decisión se tomó debido a que las imágenes a clasificar son muy parecidas entre sí (mismo fondo, misma gama de colores) lo que ocasiona problemas de singularidad en las matrices con las que trabaja el método de Newton-Lanczos.

Como se presentó en capítulos anteriores, se desea maximizar el cociente de la matriz de dispersión entre clases entre la matriz dispersión intra clase (ambas proyectadas a un espacio menor por medio de la matriz de proyección  $V$  con  $k$  columnas). Se demostró en el capítulo 3 que este es equivalente a un problema

<sup>2</sup>La modificación está presentada en los códigos del apéndice B

<sup>3</sup>Para esto, se promediaron los píxeles vecinos

## CAPÍTULO 3: EXPERIMENTOS NUMÉRICOS

escalar, donde se busca el escalar  $\rho^*$  y la matriz  $V^{**}$  que lo optimice. Para ejemplificar este proceso, se realizaron dos experimentos para cada base de datos. El primero consiste en encontrar  $\rho^*$  y la matriz  $V^{**}$  que optimiza este problema cuando se fija a  $(k)$  igual a 20 mediante el ADLF. Esto se realizó con tres objetivos:

- Ejemplificar las fórmulas planteadas para las cotas de la solución  $\rho^*$
- Analizar el cambio en el valor de  $f(\rho)$  conforme aumentan las iteraciones del método de Newton-Lanczos
- Graficar los datos proyectados en una dimensión 20-dimensional

En el segundo experimento para cada base de datos se entrenaron los tres métodos (ADLF, ADL, RLM) con distintos tamaños de proyección  $k$ . Después se calculó el error para el conjunto de prueba con cada uno de los métodos y el tiempo de cómputo.

### 3.3. Base de datos State Farm

*State Farm* es un grupo de compañías de seguros y servicios financieros en Estados Unidos. Esta compañía liberó una base de datos de 4 Gb de información en donde se muestran fotografías de conductores realizando 10 actividades:

- c0: Manejando con ambas manos
- c1: Escribiendo mensajes de texto con la mano derecha
- c2: Hablando por teléfono con la mano derecha
- c3: Mandando mensajes de texto con la mano izquierda
- c4: Hablando por teléfono con la mano izquierda
- c5: Manipulando el radio del automóvil
- c6: Tomando líquidos
- c7: Recogiendo objetos de la parte trasera del automóvil

### CAPÍTULO 3: EXPERIMENTOS NUMÉRICOS

- c8: Peinándose y/o maquillándose
- c9: Voltear con el copiloto

El objetivo de *State Farm* al liberar esta base de datos en la plataforma de *Kaggle* fue poder detectar y clasificar a conductores distraídos, ya que las estadísticas muestran que en Estados Unidos uno de cada cinco accidentes es causado por distracciones. Esto tiene una repercusión de 425,000 personas heridas y 3,000 fallecimientos cada año <sup>4</sup>.

La base de datos consta de 22,000 fotografías debidamente clasificadas, las que fueron recopiladas con cámaras instaladas en el tablero de los automóviles. La cámara se colocó en el mismo lugar en todos los automóviles, por lo que las imágenes son muy parecidas entre sí. El conjunto de datos puede descargarse de la página Web <https://www.kaggle.com/c/state-farm-distracted-driver-detection>, cuyo link sigue vigente al 25 de julio del 2016. Las imágenes están en formato *JPEG*. Para leerlas en R, se utilizó la función *readJPEG* del paquete *jpeg*. La siguiente figura muestra un extracto ellas:



Figura 3.2: Ejemplo de fotografías de la base de datos (State Farm).

---

<sup>4</sup><https://www.kaggle.com/c/state-farm-distracted-driver-detection>

### 3.3.1. Proyección sobre un espacio de dimensión 20

#### 1) Cotas para $\rho^*$

En el capítulo 2 se presentaron cotas para  $\rho^*$  y en esta sección se ejemplificará la segunda de ellas para esta base de datos. Se calcularon los eigenvalores de la matriz intraclass (B) y de la matriz entre clases (A). Los 20 eigenvalores más grandes y más pequeños de la primera son <sup>5</sup>:

291279	172080	135390	101406	42280	37705	30164
26542	24004	20359	17953	16532	15259	14248
13056	12014	11291	10387	10143	9486	

116	115	115	114	113	113	112	112	111	111
110	109	108	103	101	100	96	93	92	86

Mientras que los 9 eigenvalores más grandes de la matriz entre clases (A) son (los demás son 0):

1085	7484	4378	2794	2367	1830	1544	1029	748
------	------	------	------	------	------	------	------	-----

Sustituyendo estos valores en la fórmula de las cotas de la raíz, se tiene que:

$$\frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_i}} \leq \rho^* \leq \frac{\sum_{i=1}^p \lambda_{(A)_i}}{\sum_{i=1}^p \lambda_{(B)_{n-i+1}}}$$

$$\frac{33027.3}{1011588} \leq \rho^* \leq \frac{33027.3}{2128.5}$$

$$0.03265 \leq \rho^* \leq 13.51676$$

De esta manera, se sabe que la solución óptima se encuentra en el intervalo  $[0.03265, 13.51676]$ .

<sup>5</sup>los eigenvalores menores a  $1e^{-10}$  se consideraron numéricamente como 0

**2) Valores de  $(\rho^*, f(\rho))$** 

Para el punto inicial de este experimento se utilizará el punto medio del intervalo. Los criterios de paro se fijaron con una tolerancia de  $1e^{-10}$  y que las iteraciones sean menor a 50. Con este ejemplo, se obtienen los siguientes resultados para  $\rho$  y  $f(\rho)$ :

$iter$	$\rho$	$f(\rho)$
1	7.774706	-14,702.39
2	1.263660	4,833.57
3	1.974322	993.75
4	2.201148	48.01451
5	2.213198	0.11282
6	2.213226	$6.230598e^{-7}$
7	2.213226	$-2.097522e^{-11}$

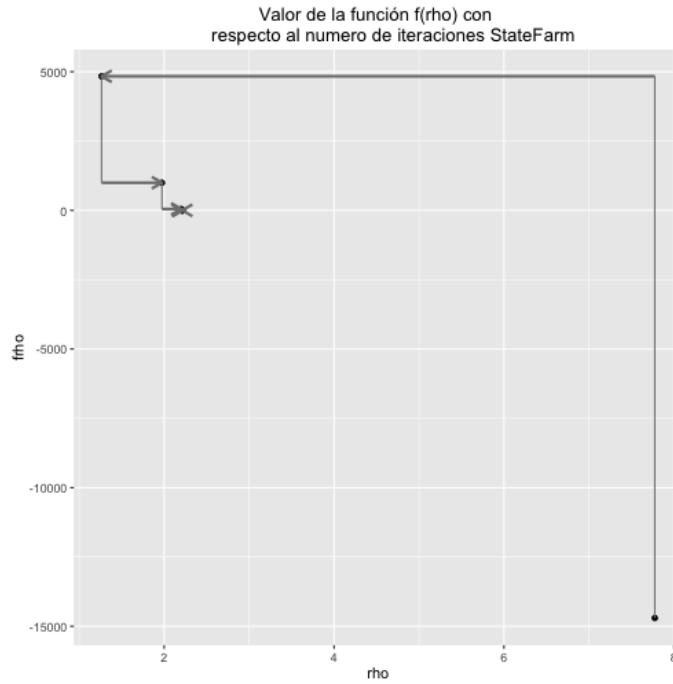


Figura 3.3: Valores de  $\rho$  y  $f(\rho)$  para distintas iteraciones (State Farm).

### 3) Primeras 4 componentes de los datos proyectados

Se maximizó la traza del cociente para un espacio de 20 dimensiones. En la figura 3.4 se muestra como se agrupan los individuos del conjunto de entrenamiento conforme las iteraciones avanzan.

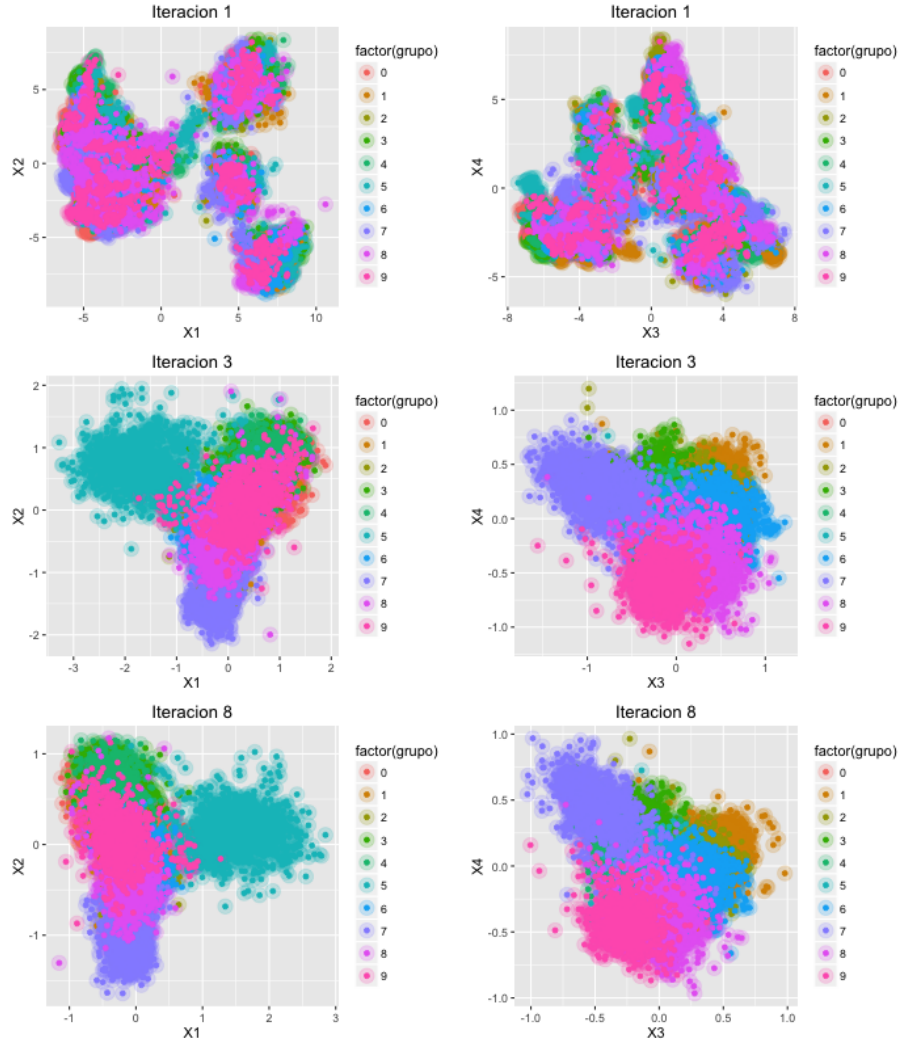


Figura 3.4: Las dos imágenes superiores son los datos de entrenamiento proyectadas sobre las 4 primeras componentes tras la primer iteración. Las dos intermedias son en la iteración 3 y las últimas 2 son la última iteración.



### 3.3.2. Comparación con otros métodos

Para la comparación del ADLF implementado con la RLM y el ADL se tomaron las siguientes consideraciones:

- Las dimensiones a considerar ( $k$ ) son: 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80 y 85
- El número de variables que entran en cada modelo es el mismo (dada la dimensión  $k$ )
- Los modelos se ajustan con el conjunto de entrenamiento y se reporta el error de prueba
- Se toman las primeras  $k$  componentes principales y se aplica el método de manera que sobre esta dimensión, se maximice el cociente de trazas. Después se usó 3-vecinos más cercanos para clasificarlos
- Para el caso de regresión logística multinomial se elige la clase que tenga una mayor probabilidad posterior de selección. Este modelo solo toma como entrada las primeras  $k$  componentes principales

#### 1) Tasa de reconocimiento

La tasa de reconocimiento se mide como el número de individuos clasificados correctamente entre el número total de individuos del conjunto de test. Como el objetivo de esta sección es observar como se modifica esta tasa con respecto a aumenta  $k$ , se fijó el número de vecinos más cercanos a 3 para el caso de ADL y del ADLF y la dimensionalidad del conjunto de entrenamiento se define como  $k$ .

En la figura 3.5 se muestra que el método del ADLF se desempeña mucho mejor que los otros dos. Hay que recordar que este agrupa las clases por lo que es sensato que  $k$ -vecinos más cercanos se desempeñe bien con este preprocesamiento.

## CAPÍTULO 3: EXPERIMENTOS NUMÉRICOS

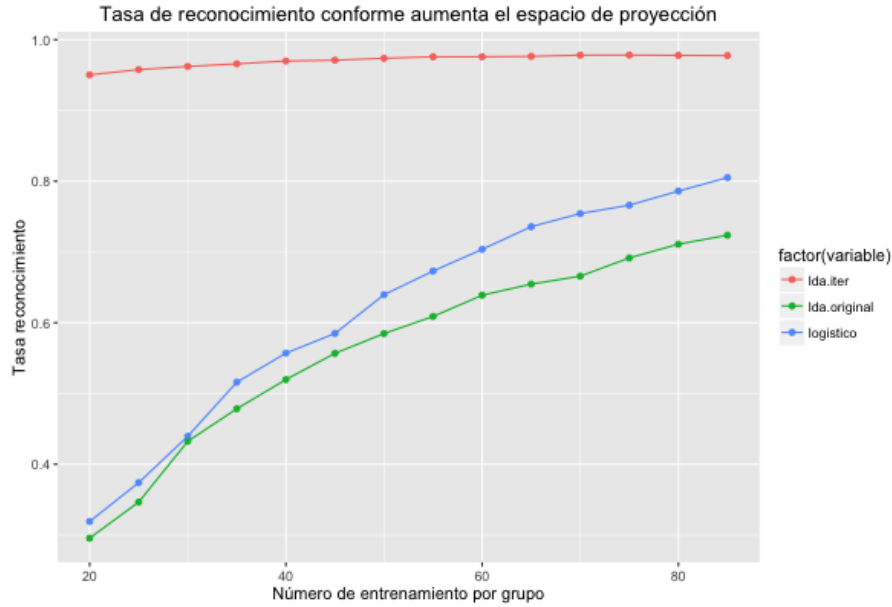


Figura 3.5: En azul se representa el RLM, en verde el ADL y en rojo el método de Newton-Lanczos. En el eje  $y$  se muestra la tasa de reconocimiento y en el eje  $x$  la dimensión a considerar ( $k$ ).

### 2) Tiempo de ejecución

A continuación, se comparan los tiempos de ejecución de cada método. Para compararlos justamente, se tomaron 3 distintos:

- 1) (PCA) Tiempo para calcular los componentes principales del conjunto de entrenamiento
- 2) (Proyectar) Tiempo para calcular la proyección del conjunto de test
- 3) (Modelo) Tiempo necesario para hacer las operaciones de cada método

PCA (en seg)	Proyectar (en seg)
178.602	3.112

### CAPÍTULO 3: EXPERIMENTOS NUMÉRICOS

Comp	lda.iter	logístico	lda.orig	Comp	lda.iter	logístico	lda.orig
20	0.0372	3.1100	0.0696	55	0.1050	9.6192	0.2304
25	0.0454	3.6184	0.0984	60	0.1114	10.5422	0.2904
30	0.0566	6.3038	0.1054	65	0.1358	11.7946	0.3200
35	0.0630	6.9988	0.1386	70	0.1378	12.5136	0.3400
40	0.0744	7.7222	0.1764	75	0.1422	13.1488	0.3788
45	0.1038	8.6158	0.2116	80	0.1522	14.0320	0.4220
50	0.0954	9.1860	0.1996	85	0.1644	13.9156	0.4920

El tiempo está en segundos (s).

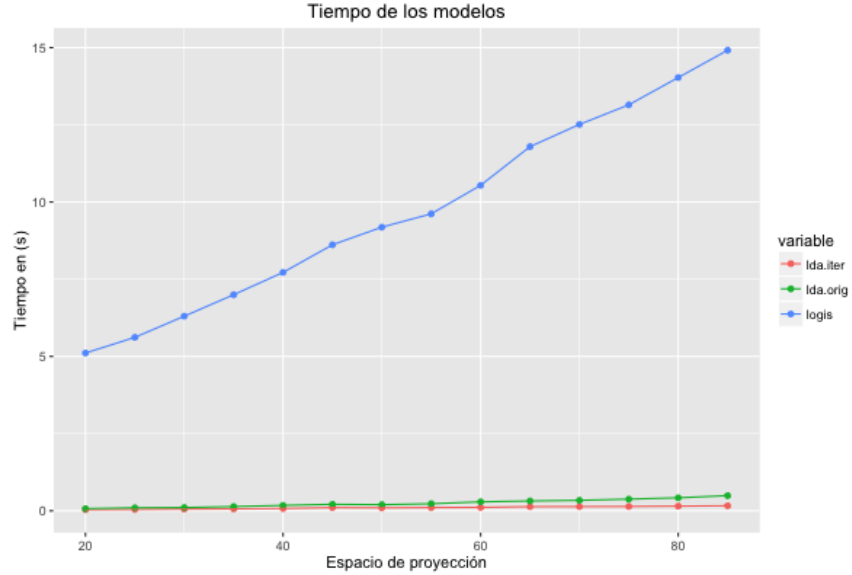


Figura 3.6: Tiempo requerido para cada método. En el eje  $x$  se encuentra el número de componentes tomadas, en el  $y$  el tiempo en segundos

Para cada iteración del método de Newton-Lanczos, se calcularon todos los eigenvalores ya que al no calcularlos, solo se tendrá una aproximación a sus valores y sus eigenvectores. Por este motivo, se decidió sacrificar tiempo de cómputo por precisión.

### 3.4. Base de datos Otto Group

Ahora se analizará la base de artículos de *Otto Group*, una compañía de comercio electrónico con presencia en más de 20 países. Actualmente la compañía tiene problemas con el clasificador que utilizan, ya que artículos muy parecidos están asignados en grupos distintos, por lo que desean reducir el error de clasificación. La base consta de 61,878 productos que pertenecen a 9 clases distintas, entre ellas moda, electrónicos, etc. De este número se seleccionaron a 19,000 (1,900 de cada clase) para el análisis. Cada artículo tiene asociado 93 columnas con características numéricas, de las cuales no se proporcionó descripción. La base fue liberada en la plataforma de *Kaggle* con el objetivo de encontrar un clasificador para los artículos de su base de datos. Puede descargarse de la página Web <https://www.kaggle.com/c/otto-group-product-classification-challenge>, cuyo link sigue vigente al 30 de julio del 2016.

La base de datos está en un formato *.csv*, por lo que se requirió la función *read.csv* del paquete *utils* para cargarlas en R.

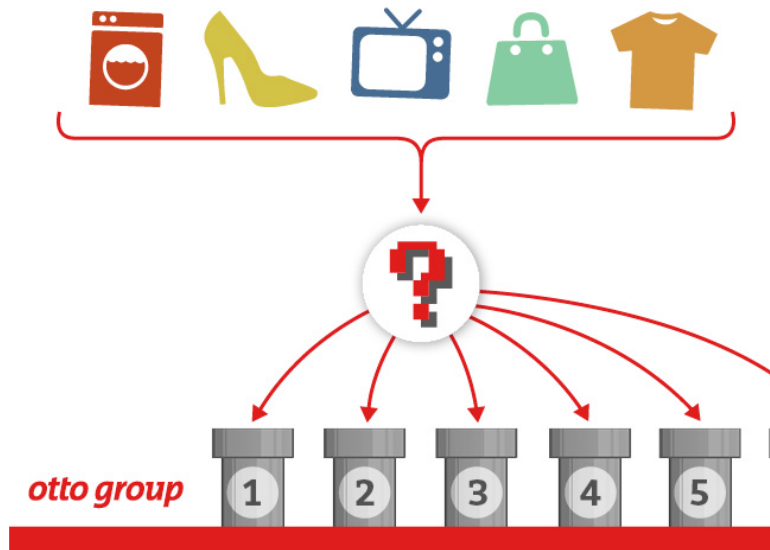


Figura 3.7: Base de datos Otto Grup. Su objetivo es clasificar artículos dentro de 9 categorías distintas.

### 3.4.1. Proyección sobre un espacio de dimensión 20

#### 1) Cotas para $\rho^*$

Al igual que el ejemplo de *State Farm*, se ejemplificará la segunda cota presentada en el capítulo 2. Para esto se calcularon los eigenvalores de la matriz intracase (B) y de la matriz entre clases (A). Respectivamente los 20 eigenvalores más grandes y más pequeños de la primera son: <sup>6</sup>

459463	284378	266461	227267	203680	154707	132582
126935	121803	109100	101550	93143	83423	80121
76779	74474	73521	66576	64987	63187	

195057	145551	99689	86433	61103	17601	8990	2517	0	0
0	0	0	0	0	0	0	0	0	0

De la misma manera, se calculan los 9 eigenvalores más grandes de la matriz entre clases (Los demás son 0):

14051	12922	7775	3247	3138	2849	1918	1336	1317
-------	-------	------	------	------	------	------	------	------

Usando las formulas del capítulo 2, se tienen las siguientes cotas para  $\rho^*$ :

$$\frac{\sum_{i=1}^p \lambda_{A_i}}{\sum_{i=1}^p \lambda_{B_i}} \leq \rho^* \leq \frac{\sum_{i=1}^p \lambda_{(A)_i}}{\sum_{i=1}^p \lambda_{(B)_{n-i+1}}}$$

$$\frac{616,943}{2,864,149} \leq \rho^* \leq \frac{616,943}{61,150}$$

$$0.215402 \leq \rho^* \leq 10.08895$$

---

<sup>6</sup>los eigenvalores menores a  $1e^{-10}$  se consideraron numéricamente como 0

## 2) Valores de $(\rho^*, f(\rho^*))$

Los criterios de paro se fijaron igual que el ejemplo pasado (tol:  $1e^{-10}$  y menos de 50 iteraciones). Para la base de *Otto Group*, se encontraron los siguientes resultados:

$iter$	$\rho$	$f(\rho)$
1	3.152175	-313,852
2	0.034971	576,318
3	0.558585	250,639
4	1.204773	71,642
5	1.543072	4,015
6	1.564269	13.14
7	1.564349	$2.26e^{-04}$
8	1.564349	$3.64e^{-11}$

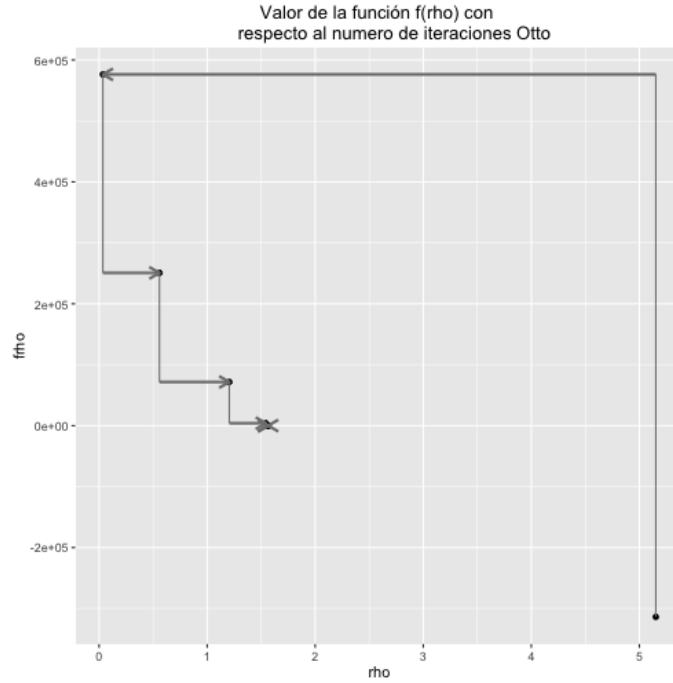


Figura 3.8: Valores de  $\rho$  y  $f(\rho)$  para las iteraciones (Otto Group).

### 3) Primeras 4 componentes de los datos proyectados

De nuevo se maximizó la traza del cociente para un espacio de 20 dimensiones. En la figura 3.8 se muestra como se agrupan los individuos del conjunto de entrenamiento conforme las iteraciones aumentan.

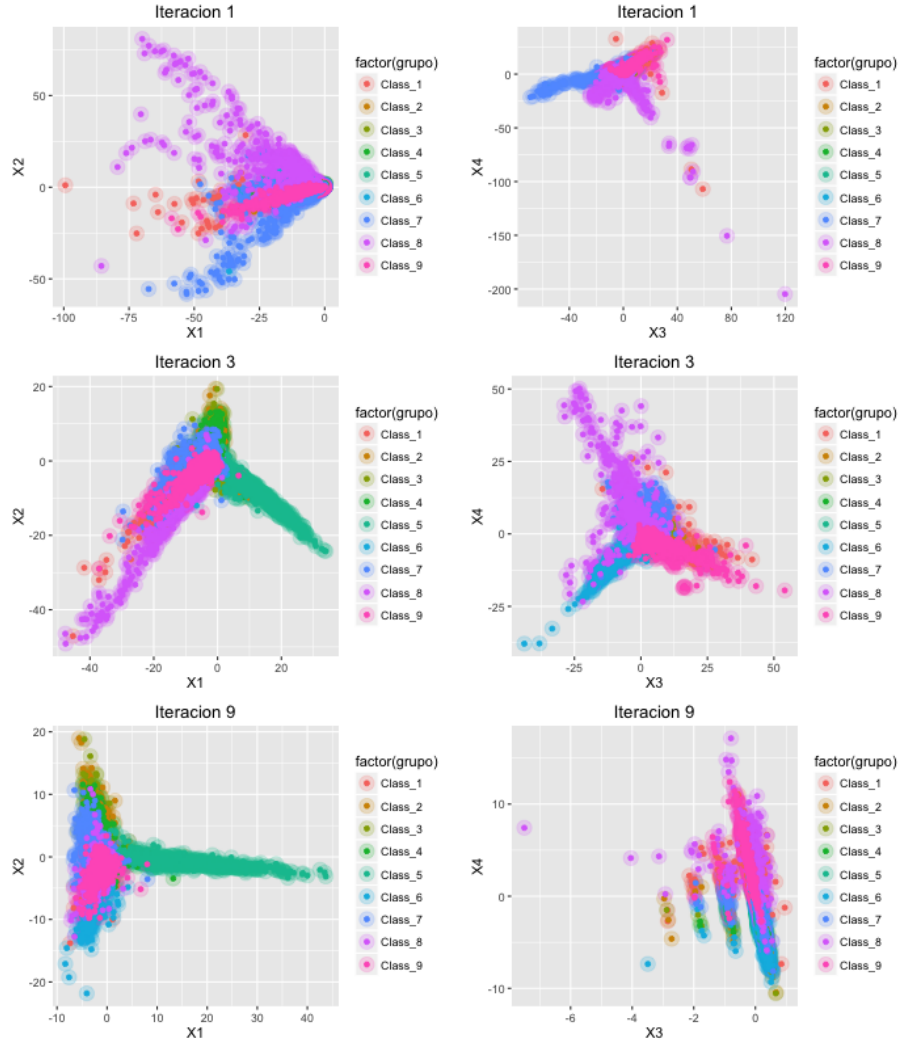


Figura 3.9: Las dos imágenes superiores son los datos de entrenamiento proyectadas sobre las 4 primeras componentes tras la primer iteración. Las dos intermedias son en la iteración 3 y las últimas 2 son la última iteración.

### 3.4.2. Comparación con otros métodos

De la misma forma que el ejemplo pasado se compararon los métodos en este caso. Las dimensiones a considerar  $k = 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80$  y  $85$ .

#### 1) Tasa de reconocimiento

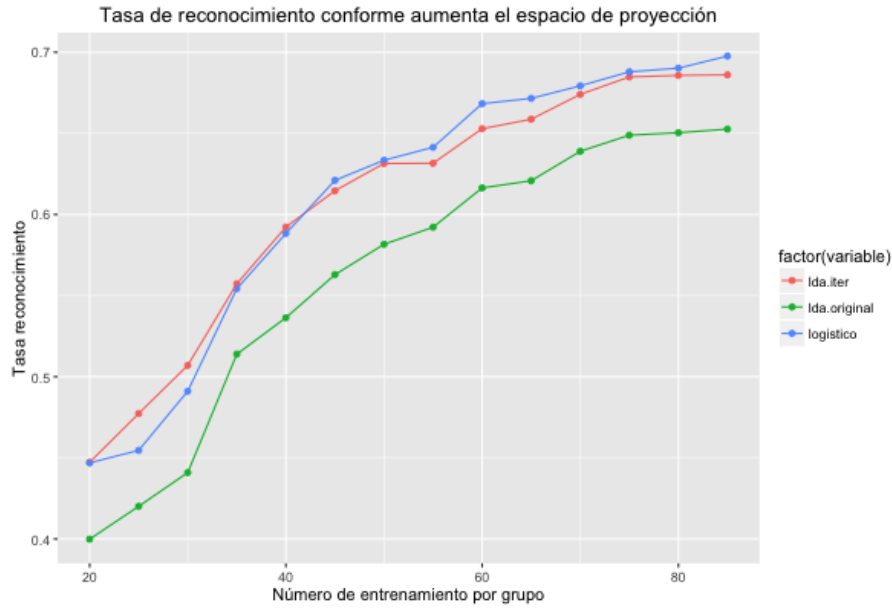


Figura 3.10: En azul se representa el RLM, en verde el ADL y en rojo el método de Newton-Lanczos. En el eje  $y$  se muestra la tasa de reconocimiento y en el eje  $x$  la dimensión a considerar ( $k$ ).

En la figura 3.9 se observa que el método de ADLF compite con el método de RLM y por debajo se encuentra el LDA. Para más de 80 componentes el método de ADLF supera la barrera del 70 % de clasificación. Hay que tomar en cuenta que el número de clases son 9, por lo que una clasificación aleatoria acertaría colamente en el 11.11 %.



**2) Tiempo de ejecución** Para el conjunto de datos no se realizó componentes principales; por esto solo se reportan el tiempo que tomó para cada modelo.

Comp	lda.iter	logístico	lda.orig	Comp	lda.iter	logístico	lda.orig
20	0.0276	2.7682	0.0534	55	0.0746	6.8098	0.1694
25	0.0316	3.5466	0.0690	60	0.0806	7.4876	0.2314
30	0.0428	3.9798	0.0830	65	0.1166	8.4616	0.2250
35	0.0468	3.8974	0.1102	70	0.1034	9.5988	0.2714
40	0.0544	3.7446	0.1134	75	0.1174	9.7264	0.2738
45	0.0590	3.4342	0.1308	80	0.1140	10.0674	0.3002
50	0.0676	6.0322	0.1492	85	0.1298	10.7434	0.3502

El tiempo está en segundos (s).

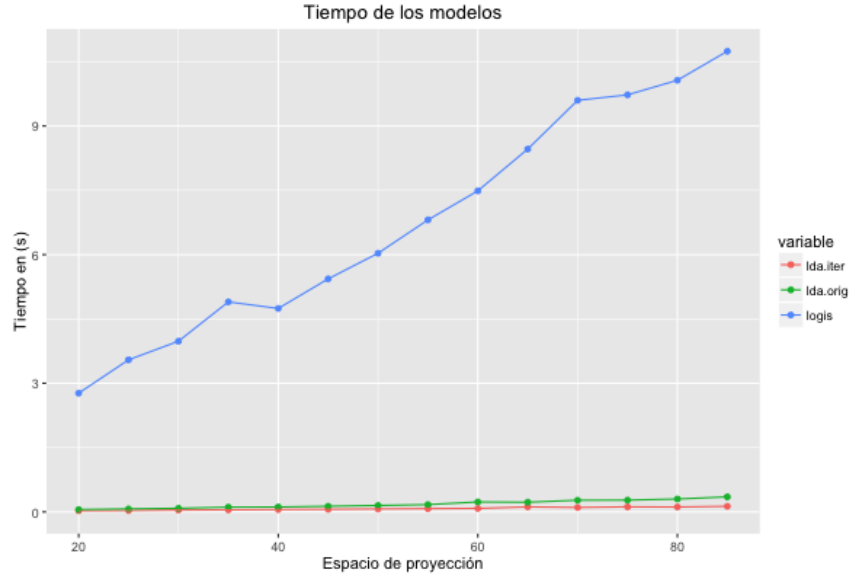


Figura 3.11: Tiempo requerido para cada método. En el eje  $x$  se encuentra el número de componentes tomadas, en el  $y$  el tiempo en segundos.

Para cada iteración del método de Newton-Lanczos, se calcularon todos los eigenvalores ya que al no calcular todos, solo se tendrá una aproximación a sus valores y sus eigenvectores. Por este motivo, se decidió sacrificar tiempo de cómputo por precisión.

## Capítulo 4

# Conclusiones

En este trabajo se expone la técnica de optimización de Newton-Lanczos aplicada al problema de aprendizaje supervisado del Análisis Discriminante Lineal de Fisher (ADLF). El ADLF busca encontrar la mejor matriz de proyección que maximice el cociente de trazas, logrando que los individuos de una misma clase sean proyectados lo más cercano posible entre ellos y lo más separado posible de otra clase. Anteriormente, la solución era considerada computacionalmente costosa de resolver; sin embargo, en la actualidad se puede aprovechar la velocidad de convergencia del método de Newton y la rapidez del algoritmo de Lanczos para calcular eigenpares.

El método de Lanczos es muy efectivo para calcular solamente algunos de los eigenpares de las matrices. Bajo el contexto de aritmética inexacta, Lanczos implementado con reortogonalización completa resulta ser más eficiente que la factorización SVD, cuando se desea calcular menos del 5% de los eigenpares. Aunque este número parece ser demasiado pequeño resulta de gran utilidad; por ejemplo, si la matriz tiene dimensión de  $1000 \times 1000$ , tener 50 eigenpares podría ser adecuado para el problema.

## CAPÍTULO 4: CONCLUSIONES

En este trabajo de tesis se comparó el método ADLF vía Newton-Lanczos con el Análisis Discriminante Lineal (ADL) y la técnica de Regresión Lineal Múltiple (RLM). La comparación se realizó empleando las bases *State Farm* y *Otto Group*. En el primero, se logró una precisión mejor que los otros dos métodos con un tiempo de cómputo similar. Para el segundo, la precisión entre los tres métodos es muy similar con tiempos de cómputo similares.

El método funciona muy bien para problemas de mediana y gran dimensionalidad. Si se trata de imágenes es muy importante que tengan un preprocesamiento adecuado, ya que si el objeto a clasificar se encuentra en distintos píxeles de la imagen, no se tendrá buenos resultados.

Las principales conclusiones, aportaciones y aprendizajes de esta tesis son las siguientes:

- Se implementó computacionalmente una técnica de optimización que anteriormente resultaba difícil de resolver.
- El ADLF es un método muy flexible, ya que no requiere ningún supuesto sobre la distribución de los datos.
- La eficiencia del ADLF es superior a la RLM y similar al ADL. Los resultados fueron satisfactorios.
- La precisión del método resulta comparable con los métodos de clasificación lineal, llegando incluso a mejorarlos.
- El ADLF permite visualizar los datos en dimensiones menores separando las clases lo mejor posible.

Una de las complicaciones del algoritmo de Lanczos es la reortogonalización de la base. En esta tesis se utilizó el método de reortogonalización completa; sin embargo, existen modificaciones al algoritmo que pueden ser exploradas con el objetivo de lograr mayor eficiencia en términos computacionales. Por ejemplo, J.W. Demmel (1997) [1] propone algunas alternativas que pueden ser utilizadas para mejorar el proceso de reortogonalización de la base en el algoritmo de Lanczos.

## Apéndice A

# Apéndice A: Optimización de $Tr(V^T AV)$

Sea  $A \in \mathbb{R}^{n \times n}$  una matriz simétrica cuya factorización espectral es:

$$A = U^T \Lambda U \quad U^T U = I_n$$
$$\Lambda = \text{diag}(\lambda_{A_1}, \dots, \lambda_{A_n})$$

donde  $\lambda_{A_1} \geq \lambda_{A_2} \geq \dots \geq \lambda_{A_n}$  son los valores propios de  $A$  y  $U$  una matriz ortogonal. En este apéndice se demostrará que:

$$\max_{\substack{V^T V = I \\ V \in \mathbb{R}^{n \times p}}} Tr(V^T AV) = \lambda_{A_1} + \lambda_{A_2} + \dots + \lambda_{A_p} \quad (\text{A.1})$$

### A.1. Problema relajado

Sean  $\mathcal{U}_p$  y  $\mathcal{V}_p$  dos conjuntos de matrices en  $\mathbb{R}^{n \times p}$  tales que:

$$\mathcal{U}_p = \{V \in \mathbb{R}^{n \times p} | V^T V = I_p\}$$
$$\mathcal{V}_p = \{V \in \mathbb{R}^{n \times p} | \text{diag}(V^T V) = I_p\}$$

El conjunto  $\mathcal{V}_p$  contiene a todas las matrices donde que cada columna tiene norma euclidiana igual a 1.

APÉNDICE A: APÉNDICE A: OPTIMIZACIÓN DE  $Tr(V^T AV)$

**Lema A.1.** *El conjunto  $\mathcal{V}_p$  es compacto en  $\mathbb{R}^{n \times p}$*

*Demostración.* Por definición, si  $V \in \mathcal{V}_p$  entonces  $\|V\|_F = \sqrt{p}$ . Más aún,  $\mathcal{V}_p$  contiene todos sus puntos límite.  $\square$

Si se relaja el problema original como:

$$\max_{V \in \mathcal{V}_p} Tr(V^T AV) \quad (\text{A.2})$$

Como  $\mathcal{V}_p$  es un conjunto compacto, la función continua  $Tr(V^T AV)$  alcanza su valor máximo (o mínimo) en este conjunto. Ahora, como  $\mathcal{U}_p \subset \mathcal{V}_p$ , se tiene inmediatamente la desigualdad:

$$\max_{V \in \mathcal{U}_p} Tr(V^T AV) \leq \max_{V \in \mathcal{V}_p} Tr(V^T AV) \quad (\text{A.3})$$

Por lo tanto, se se procede a establecer el siguiente resultado:

**Teorema A.1.**

$$\max_{V \in \mathcal{V}_p} Tr(V^T AV) = \lambda_{A_1} + \lambda_{A_2} + \dots + \lambda_{A_p}$$

*Demostración.* Para  $V \in \mathcal{V}_p$  con  $V = (v_1 \ v_2 \ \dots \ v_p)$  y  $v_j$  la  $j$ -ésima columna de  $V$ . Se define el vector:

$$\mathbf{v} = (v_1^T \ v_2^T \ \dots \ v_p^T)^T \in \mathbb{R}^{np}$$

o lo que es equivalente:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_p \end{pmatrix}$$

Entonces la  $Tr(V^T AV) = v_1^T A v_1 + v_2^T A v_2 + \dots + v_p^T A v_p$  y el problema A.2 puede ser formulado como sigue:

$$\max_{\substack{\mathbf{v}^T \mathbb{B}_j \mathbf{v} = 1 \\ j=1, \dots, p}} \mathbf{v}^T \mathbb{A} \mathbf{v} \quad (\text{A.4})$$

APÉNDICE A: APÉNDICE A: OPTIMIZACIÓN DE  $Tr(V^T AV)$

con las matrices  $\mathbb{A}$  y  $\mathbb{B}_j$ :

$$\mathbb{A} = \begin{pmatrix} A & 0 & \dots & 0 & \dots & 0 \\ 0 & A & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & A & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \dots & A \end{pmatrix}$$

$$\mathbb{B}_j = \begin{pmatrix} 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & I_n & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix}$$

En ambas matrices hay  $p$  bloques en los renglones y  $p$  bloques en las columnas. Para  $\mathbb{B}_j$  el bloque  $(j, j)$  es  $I_n$  y todos los demás son matrices cero. De esta manera, la función lagrangiana asociada al problema es:

$$\mathcal{L}_j(\mathbf{v}, \eta) = \mathbf{v}^T \mathbb{A} \mathbf{v} - \sum_{j=1}^p (\eta_j (\mathbf{v}^T \mathbb{B}_j \mathbf{v} - 1))$$

Las condiciones de primer orden para la solución óptima son (considerando que  $A$  es simétrica:

$$\begin{aligned} \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}, \eta) &= 2\mathbb{A} \mathbf{v} - 2 \sum_{j=1}^p (\eta_j (\mathbf{v}^T \mathbb{B}_j \mathbf{v})) = 0 \\ \nabla_{\eta_j} \mathcal{L}(\mathbf{v}, \eta) &= \mathbf{v}^T \mathbb{B}_j \mathbf{v} - 1 \quad \text{con } j = 1, \dots, p \end{aligned} \tag{A.5}$$

De la primera ecuación de A.5, se tiene que:

$$\left( \mathbb{A} - \sum_{j=1}^p \eta_j \mathbb{B}_j \right) \mathbf{v} = 0$$

Donde la matriz  $(\mathbb{A} - \sum_{j=1}^p \eta_j \mathbb{B}_j)$  es diagonal a bloques. De hecho, el bloque  $(j, j)$  es:

APÉNDICE A: APÉNDICE A: OPTIMIZACIÓN DE  $Tr(V^T AV)$

$$(A - \eta_j I_n)v_j = 0 \quad \Rightarrow \quad Av_j = \eta_j v_j$$

Entonces,  $\eta_j$  debe ser un eigenvalor de A y  $v_j$  su correspondiente eigenvector. Tomando de nuevo la ecuación de arriba, y multiplicándola por  $v^T$  se obtiene:

$$v^T \mathbb{A} v = \sum_{j=1}^p (\eta_j) (v^T \mathbb{B}_j v)$$

Ahora, usando la segunda ecuación de A.5, se tiene que:

$$v^T \mathbb{A} v = \sum_{j=1}^p \eta_j$$

Por lo tanto, con el fin de maximizar la función  $v^T \mathbb{A} v$  en  $\mathcal{V}_p$ , se deben escoger los primeros  $p$  eigenvalores de la matriz A. Esto es:

$$\eta_j = \lambda_{A_j}, \quad j = 1, \dots, p$$

□

Ahora, se tienen todas las piezas para demostrar el resultado principal.

**Teorema A.2.**

$$\max_{\substack{V^T V = I_p \\ V \in \mathbb{R}^{n \times p}}} Tr(V^T AV) = \lambda_{A_1} + \lambda_{A_2} + \dots + \lambda_{A_p}$$

*Demostración.* Se ha mostrado que:

$$\max_{V \in \mathcal{U}_p} Tr(V^T AV) \leq \lambda_{A_1} + \lambda_{A_2} + \dots + \lambda_{A_p}$$

□

Considerando la matrix  $U \in \mathbb{R}^{n \times n}$  de la factorización espectral de A, tal que  $U = (u_1 \ u_2 \ \dots \ u_n)$ . Tomando las primeras  $p$  columnas de  $U$  se define  $U^*$ :

$$U^* = (u_1 \ u_2 \ \dots \ u_p) \quad \text{con} \quad U^* \in \mathcal{U}_p \quad y$$

$$Tr(U^* AU) = \lambda_{A_1} + \lambda_{A_2} + \dots + \lambda_{A_p}$$

## A.2. Observaciones finales

Se toman los resultados de la sección anterior para hacer las siguientes observaciones:

**Observación A.1.** *Se puede extender el último teorema, usando el mismo conjunto  $\mathcal{V}_p$  y cambiando la maximización a minimización para obtener que:*

$$\min_{V \in \mathcal{U}_p} Tr(V^T AV) = \lambda_{A_n} + \lambda_{A_{n-1}} + \dots + \lambda_{A_{n-(p-1)}} \quad (\text{A.6})$$

**Observación A.2.** *Cuando  $p = 1$ , se pueden obtener las bien conocidas propiedades del cociente de Raleigh:*

$$\begin{aligned} \left( \min_{v^v=1} v^T Av \right) &= \lambda_{A_n} \\ \left( \max_{v^v=1} v^T Av \right) &= \lambda_{A_1} \end{aligned} \quad (\text{A.7})$$



# Bibliografía

- [1] J.W. Demmel. *Applied Numerical Linear Algebra*. Miscellaneous Bks. Society for Industrial and Applied Mathematics, 1997.
- [2] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.
- [3] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 2013.
- [4] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
- [5] Trevor Hastie, Robert Tibshirani, and Friedman. *The elements of statistical learning*, volume 2. Springer, 2009.
- [6] Cornelius Lanczos. *Applied analysis*. Courier Corporation, 1988.
- [7] Tom Michael Mitchell. *The discipline of machine learning*. Carnegie Mellon University, School of Computer Science, Machine Learning Department, 2006.
- [8] Thanh T Ngo, Mohammed Bellalij, and Yousef Saad. The trace ratio optimization problem. *SIAM review*, 54(3):545–569, 2012.
- [9] Huan Wang, Shuicheng Yan, Dong Xu, Xiaoou Tang, and Thomas Huang. Trace ratio vs. ratio trace for dimensionality reduction. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.