

Departamento de Matemáticas, Facultad de Ciencias
Universidad Autónoma de Madrid

Análisis de arquetipos y aplicaciones

TRABAJO DE FIN DE GRADO

Grado en Matemáticas

Autor: Guillermo García Cobo
Tutor: Javier Cárcamo Urtiaga

Curso 2021-2022

Resumen

El análisis de arquetipos es una técnica de aprendizaje no supervisado que busca representar los datos de una muestra a través de combinaciones convexas de los puntos extremos de la misma. Estos puntos extremos, los llamados arquetipos, pueden considerarse representantes de la muestra, por lo que pueden aplicarse en reducción de la dimensionalidad y *clustering*. Además, el uso de combinaciones convexas aporta interpretabilidad, una propiedad deseable que no tienen otros métodos.

Este trabajo consta de dos partes. La primera de ellas busca definir de forma teórica el análisis de arquetipos. Para ello, primero presentaremos los conceptos matemáticos necesarios para entender la base que sustenta a los arquetipos. Entre otras cosas, recordaremos nociones de convexidad y optimización. Con estos conceptos claros, se definirán los arquetipos y el procedimiento originalmente propuesto para calcularlos.

En la segunda parte, nos centraremos en la aplicación práctica del método descrito. En primer lugar, describiremos y compararemos variantes más eficientes del procedimiento de cálculo de los arquetipos. Por otro lado, tras obtener una mejora sustancial en los tiempos de computación, aplicaremos el análisis de arquetipos a dos ejemplos reales: reconocimiento facial y segmentación de clientes. Gracias a esto, observaremos de primera mano las ventajas que ofrecen los arquetipos frente a otras técnicas de aprendizaje no supervisado.

Abstract

Archetypal analysis is an unsupervised learning technique that seeks to represent the observations of a sample through convex combinations of its extreme points. These extreme points, the so-called archetypes, can be considered as representatives of the sample, so they can be applied in dimensionality reduction and clustering. In addition, the use of convex combinations provides interpretability, a desirable property that other methods do not have.

This work consists of two parts. The first one seeks to theoretically define the archetypal analysis. To do this, we will first present the mathematical concepts necessary to understand the foundational basis of the archetypes. Among other things, we will remind notions of convexity and optimization. With these concepts clear, we will proceed to define the archetypes and the procedure originally proposed to calculate them.

In the second part, we will focus on the practical application of the described method. First, we will describe and compare more efficient variants of the archetype calculation procedure. On the other hand, after obtaining a substantial improvement in computing times, we will apply the archetypal analysis to two real examples: facial recognition and customer segmentation. Thanks to this, we will observe first-hand the advantages offered by archetypes over other unsupervised learning techniques.

Índice general

1	Introducción y preliminares	1
1.1	Introducción	1
1.2	Resultados preliminares	2
1.2.1	Conjuntos convexos	2
1.2.2	Funciones convexas	5
1.2.3	Optimización convexa	7
1.2.4	Biconvexidad	8
1.2.5	Descenso por gradiente en optimización convexa	9
2	Descripción del método original	13
2.1	Motivación y definición	13
2.2	Localización de los arquetipos	14
2.3	Cálculo de los arquetipos	16
3	Métodos mejorados de cálculo de arquetipos	19
3.1	Cálculo del gradiente	19
3.2	Descripción de los métodos mejorados	20
3.2.1	Gradiente proyectado	20
3.2.2	Adaptación del Algoritmo de Frank-Wolfe	21
3.3	Comparación de los métodos	21
4	Ejemplos reales de análisis de arquetipos	25
4.1	Reconocimiento facial	25
4.2	Segmentación de clientes	28
5	Conclusiones y trabajo futuro	33

CAPÍTULO 1

Introducción y preliminares

1.1. Introducción

Con la proliferación de áreas relacionados con el aprendizaje automático, el aprendizaje no supervisado es una de las que más atención está recibiendo últimamente. El hecho de no necesitar intervención humana alguna para aprender, con el consiguiente ahorro de recursos que eso conlleva, ha suscitado el interés de muchos. Por otro lado, una de las propiedades más deseadas de los modelos es que sean interpretables. Esto supone un gran valor añadido, ya que nos permite comprender completamente el resultado que producen y en qué se basan para producirlo.

El análisis de arquetipos es una técnica que auna ambas características, haciendo uso de combinaciones convexas y datos extremos, tal y como iremos describiendo en las siguientes páginas. Para lograr comprender esta técnica y sus ventajas frente a otras, se ha organizado el documento de la siguiente manera. En este Capítulo 1, se describirán con detalle los conceptos matemáticos en los que se sustentan los arquetipos. En el Capítulo 2, definiremos formalmente los arquetipos y cómo se propuso inicialmente su cálculo. En el Capítulo 3, presentaremos alternativas más eficientes de obtener arquetipos, comparándolas sobre distintos tamaños de datos y obteniendo mejoras significativas con respecto al procedimiento original. En el Capítulo 4, aplicaremos el análisis de arquetipos a dos ejemplos reales, comparando su rendimiento con respecto a otras técnicas de aprendizaje no supervisado. Finalmente, en el Capítulo 5, se describirán brevemente posibles líneas futuras de trabajo.

En cuanto a la bibliografía, artículos concretos se encuentran citados a lo largo del documento. Sin embargo, hacemos aquí un repaso de todos los documentos consultados para cada sección del trabajo. Para establecer las bases sobre convexidad, se ha consultado [6, 40, 19, 27, 34]. En cuanto a las funciones convexas, los artículos son [7, 1]; mientras que para tratar el concepto de biconvexidad el artículo en el que nos hemos basado es [20]. Para definir los conceptos relativos a optimización convexa, [7, 17, 24, 15, 2, 21]. Con el objetivo de definir el tema central que nos ocupa (análisis de arquetipos), se ha consultado [11, 16, 28, 5, 29, 14, 13, 22, 3, 30, 4]. Para comparar los distintos métodos propuestos que calculan los arquetipos, los artículos de referencia son [9, 33, 37, 13, 30, 4]. Por último, para usar los arquetipos en ejemplos reales, se han consultado [43, 42, 23, 36, 32].

1.2. Resultados preliminares

Se procede a continuación a recordar varias nociones y propiedades útiles para las secciones posteriores.

1.2.1. Conjuntos convexos

Definición 1.1. Un subconjunto $C \subset \mathbb{R}^n$ se dice *convexo* si dados $x, y \in C$, se tiene que para todo $t \in [0, 1]$

$$tx + (1 - t)y \in C.$$

Cabe mencionar que, aunque para este trabajo nos va a ser suficiente con la definición previa, evidentemente esta definición puede extenderse a espacios más generales que los euclídeos.

De manera intuitiva, un conjunto es convexo si podemos unir cualquier par de puntos con un segmento contenido en el conjunto. En la Figura 1.1 se muestra un ejemplo de esta idea.

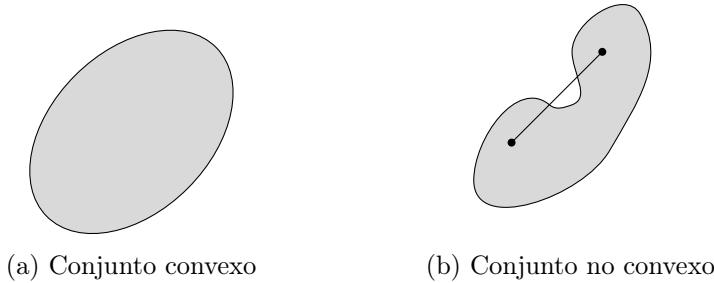


Figura 1.1: Ejemplos de convexidad en \mathbb{R}^2

Adaptación de [41]

Antes de proseguir, cabe mencionar que, salvo que se indique lo contrario, los vectores v son columna. Además, v^T indica el vector transpuesto.

Otros ejemplos de conjuntos convexos son los siguientes:

- Las bolas $B(x_0, \epsilon) = \{x \in \mathbb{R}^n : \|x - x_0\| < \epsilon\}$ (para cualquier norma).
- Hiperplanos: $C = \{x : p^T x = \alpha\}$ con $p \in \mathbb{R}^n, \alpha \in \mathbb{R}$.
- Semiespacios: $C = \{x : p^T x \leq \alpha\}$ con $p \in \mathbb{R}^n, \alpha \in \mathbb{R}$.
- Intersección arbitraria de convexos: si C_i es convexo para todo $i \in I$, entonces $C = \bigcap_{i=1}^I C_i$ es convexo.

Definición 1.2. La *combinación convexa* de $x_1, x_2, \dots, x_k \in \mathbb{R}^n$ viene dada por

$$t_1 x_1 + t_2 x_2 + \cdots + t_k x_k,$$

con $\sum_{i=1}^k t_i = 1$ y $t_i \geq 0$, para todo $i = 1, \dots, k$.

Se tiene la siguiente proposición que nos ayudará a relacionar las combinaciones convexas de un conjunto con su convexidad.

Proposición 1.3. *C es convexo si y solo si cualquier combinación convexa de puntos de C está en C.*

*Demuestra*ción. Supongamos primero que C es convexo. Procedemos por inducción sobre k , el número de puntos de la combinación convexa.

Si $k = 1$, y $x \in C$, entonces $1 \cdot x \in C$. Obsérvese que el caso $k = 2$ se tiene por definición de convexidad.

Supongámoslo cierto para $k > 1$, y los mostraremos para $k+1$. Sean $x_1, \dots, x_{k+1} \in C$. Queremos ver que

$$(1.1) \quad t_1x_1 + t_2x_2 + \cdots + t_{k+1}x_{k+1} \in C.$$

Por hipótesis de inducción, tenemos que

$$y = \frac{t_1}{t_1 + \cdots + t_k}x_1 + \frac{t_2}{t_1 + \cdots + t_k}x_2 + \cdots + \frac{t_k}{t_1 + \cdots + t_k}x_k \in C.$$

Obsérvese que los términos están bien definidos, porque si $t_1 + \cdots + t_k = 0$, la combinación convexa (1.1) equivale a $x_{k+1} \in C$, lo cual es trivialmente cierto.

Ahora, podemos expresar (1.1) como

$$t_1x_1 + t_2x_2 + \cdots + t_{k+1}x_{k+1} = (t_1 + \cdots + t_k)y + t_{k+1}x_{k+1},$$

que está en C por definición de convexidad.

Para el recíproco, basta observar que, como cualquier combinación convexa de puntos del conjunto está en el conjunto, es también cierto para combinaciones de dos elementos y por tanto se cumple la definición de convexidad. \square

Definición 1.4. El *cierre convexo* (en inglés *convex hull*) de un subconjunto C se define como el menor conjunto convexo que contiene a C y se denota por $\text{conv}(C)$. Equivalentemente, por la Proposición 1.3, podemos verlo como el conjunto de todas las posibles combinaciones convexas de C :

$$\text{conv}(C) = \left\{ \sum_{i=1}^k t_i x_i : x_i \in C, t_i \geq 0, i = 1, \dots, k, k \in \mathbb{N}, \sum_{i=1}^k t_i = 1 \right\}.$$

Se tienen las siguientes propiedades del cierre convexo:

- C es convexo si y solo si $C = \text{conv}(C)$.
- $C \subset \text{conv}(C)$.

En la Figura 1.2 podemos ver un ejemplo del cierre convexo de una nube de puntos en el plano.

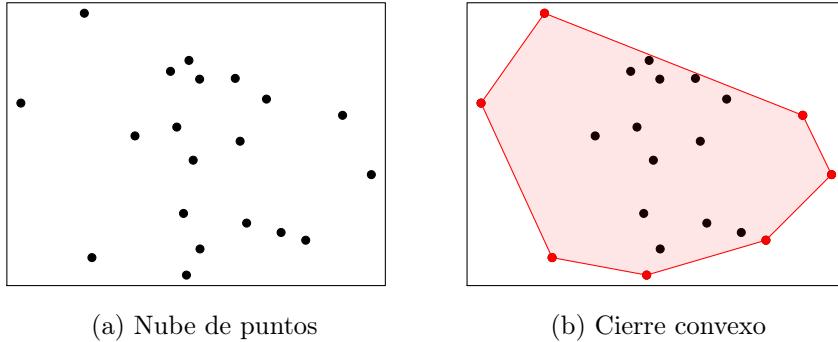


Figura 1.2: Ejemplo de cierre convexo de una nube de puntos en \mathbb{R}^2

Definición 1.5. Sea C un conjunto convexo no vacío. Se dice que $x \in C$ es un *punto extremo* si x no se puede expresar como una combinación convexa propia en C . Es decir, si $x = tx_1 + (1 - t)x_2$, con $x_1, x_2 \in C$, entonces $t = 0$ ó 1 . Denotaremos al conjunto de puntos extremos de C como $\text{Ext}(C)$.

Proposición 1.6. Si $C \subset \mathbb{R}^n$ es convexo, se verifica que $\text{Ext}(C) \subseteq \partial C$, donde ∂C indica la frontera del conjunto C .

*Demuestra*ción. Supongamos que $x \notin \partial C = \overline{C} - \text{int } C$, donde \overline{C} denota el cierre de C e $\text{int } C$ el interior de C . Entonces, $x \in \text{int } C$, luego existe $r > 0$ tal que la bola $B(x, r) \subset C$. Por tanto, x estaría en el medio de un segmento incluido en C , luego $x \notin \text{Ext}(C)$. \square

Proposición 1.7. Sea $A \subset \mathbb{R}^n$ un conjunto. Entonces, se verifica que $\text{Ext}(\text{conv}(A)) \subseteq A$.

*Demuestra*ción. Sea $x \in \text{conv}(A)$ un punto extremo. Por definición del cierre convexo, $x = \sum_i t_i a_i$ para $a_i \in A$, $\sum_i t_i = 1$ y $t_i \geq 0$. Ahora, por ser x un punto extremo, debe darse que solo un $t_i > 0$. Entonces, dicho t_i es 1 y $x = a_i$. Por tanto, $x \in A$. \square

Para ayudar a entender este concepto, en la Figura 1.2(b) se han señalado en rojo los puntos extremos del cierre convexo de la nube de puntos. Es decir, si A es la nube de puntos, los puntos señalados en rojo son $\text{Ext}(\text{conv}(A))$.

Procedemos a darle nombre al cierre convexo de un conjunto especial: el de la base de vectores unitarios estándar.

Definición 1.8. El *n-símplex* estándar, denotado por Δ^n , es el subconjunto formado por el cierre convexo de los $n + 1$ vectores unitarios estándar de \mathbb{R}^{n+1} . Es decir,

$$\Delta^n = \left\{ (t_1, \dots, t_{n+1}) \in \mathbb{R}^{n+1} : t_i \geq 0, i = 1, \dots, n+1, \sum_{i=1}^{n+1} t_i = 1 \right\}.$$

Obsérvese que los pesos de cualquier combinación convexa se pueden ver como puntos en Δ^n . Por lo tanto, siempre que queramos obtener pesos que formen una combinación convexa, debemos limitar la búsqueda al simplex adecuado. Esto será relevante más adelante, cuando busquemos combinaciones convexas válidas que minimicen cierto criterio.

En la Figura 1.3 se muestra el 2-símplex estándar contenido en \mathbb{R}^3 .

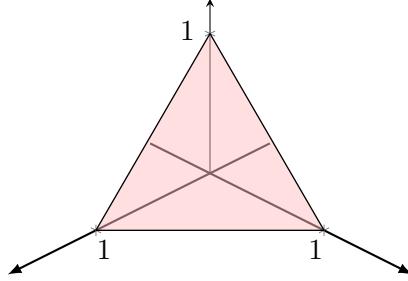


Figura 1.3: 2-símplex estándar (Δ^2)
Adaptación de [35]

Por último, enunciamos un teorema muy relevante para la construcción de arquetipos, como veremos más adelante. Este resultado relaciona las combinaciones convexas, el cierre convexo y los puntos extremos.

Teorema 1.9. *[Minkowski-Carathéodory] Sea C un subconjunto compacto y convexo contenido en \mathbb{R}^n de dimensión n_C . Entonces, cualquier punto en C es combinación convexa de a lo más $n_C + 1$ puntos extremos. En particular,*

$$C = \text{conv}(\text{Ext}(C)).$$

Este resultado es suficiente para los tipos de datos con los que vamos a tratar (reales y finitos), pero existe un teorema aún más general válido para conjuntos infinito-dimensionales.

Teorema 1.10. *[Krein-Milman] Sea C un subconjunto compacto y convexo contenido en un espacio vectorial localmente convexo. Entonces,*

$$C = \overline{\text{conv}(\text{Ext}(C))}.$$

Ambos teoremas, así como su demostración, pueden consultarse en [40, Capítulo 8] (Teoremas 8.11 y 8.14, respectivamente).

1.2.2. Funciones convexas

Definición 1.11. Una función $f : C \rightarrow \mathbb{R}$ es *convexa* si su dominio $C \subset \mathbb{R}^n$ es convexo y para todo $x_1, x_2 \in C$, y todo $t \in [0, 1]$, se tiene

$$f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2).$$

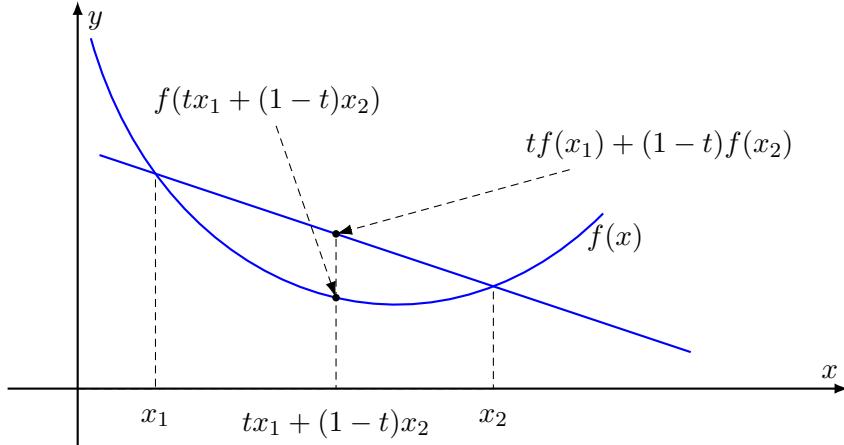


Figura 1.4: Ejemplo de función convexa $f(x)$ en \mathbb{R}^2
Adaptación de [25]

Procedemos ahora a enunciar dos resultados útiles si además de convexidad suponemos condiciones de diferenciabilidad para la función.

Teorema 1.12. *Sea $f : C \rightarrow \mathbb{R}$ diferenciable sobre un dominio convexo y abierto C . Entonces, f es convexa si y solo si para todo $x_1, x_2 \in C$*

$$(1.2) \quad f(x_1) \geq f(x_2) + \langle \nabla f(x_2), (x_1 - x_2) \rangle.$$

Demostración. Supongamos primero que f es diferenciable y convexa en C . Sean $x_1, x_2 \in C$. Como f es convexa en C , se tiene

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).$$

Despejando llegamos a

$$\frac{f(x_2 + t(x_1 - x_2)) - f(x_2)}{t} \geq f(x_1) - f(x_2).$$

Por último, tomando el límite cuando $t \rightarrow 0$ llegamos a (1.2).

Recíprocamente, sean $x, y \in C$, $t \in [0, 1]$. Sea $x^* = tx + (1-t)y$. Ahora, aplicamos (1.2) a los pares (x, x^*) , (y, x^*) para obtener:

$$(1.3) \quad f(x) \geq f(x^*) + \langle \nabla f(x^*), (x - x^*) \rangle.$$

Es decir,

$$(1.4) \quad f(y) \geq f(x^*) + \langle \nabla f(x^*), (y - x^*) \rangle.$$

Multiplicando (1.3) por t y (1.4) por $1-t$ y sumándolas obtenemos:

$$tf(x) + (1-t)f(y) \geq f(x^*) + t\langle \nabla f(x^*), (x - x^*) \rangle + (1-t)\langle \nabla f(x^*), (y - x^*) \rangle.$$

Recordando que $x^* = tx + (1 - t)y$, llegamos a

$$tf(x) + (1 - t)f(y) \geq f(tx + (1 - t)y).$$

□

Teorema 1.13. *Sea $f : C \rightarrow \mathbb{R}$ con dos derivadas continuas sobre un dominio convexo y abierto C . Entonces, f es convexa si y solo si para todo $x \in C$, $H_f(x)$ es semidefinida positiva, donde $H_f(x)$ es la matriz hessiana de f en el punto x .*

Demostración. Supongamos que H_f no es semidefinida positiva para un $x \in C$. Por continuidad del Hessiano, existe $y \in C$ tal que, para todo $t \in [0, 1]$,

$$\langle (y - x), H_f(x + t(y - x))(y - x) \rangle < 0.$$

Ahora, con esto y el desarrollo de segundo orden de Taylor, se tiene que para estos x, y ,

$$f(y) < f(x) + \langle \nabla f(x), (y - x) \rangle.$$

Usando el Teorema 1.12, concluimos que f no es convexa.

Para el recíproco, sean $x, y \in C$. De nuevo, por Taylor, sabemos que para algún $t \in [0, 1]$, se tiene

$$f(y) = f(x) + \langle \nabla f(x), (y - x) \rangle + \frac{1}{2} \langle (y - x), H_f(x + t(y - x))(y - x) \rangle.$$

Sabemos que H_f es semidefinida positiva en C y $x + t(y - x) \in C$, luego tenemos

$$f(y) \geq f(x) + \langle \nabla f(x), (y - x) \rangle.$$

Por el Teorema 1.12, sabemos que esto implica convexidad de f . □

1.2.3. Optimización convexa

Definición 1.14. Un *problema de optimización convexa* consiste en resolver lo siguiente:

$$\min_{x \in C} f(x),$$

donde $f : D \rightarrow \mathbb{R}$ es convexa y $C \subset D \subset \mathbb{R}^n$ es convexo.

Un teorema importante que nos permitirá eliminar la distinción entre mínimos locales y globales para este tipo de problemas es el siguiente:

Teorema 1.15. *Sea $f : D \rightarrow \mathbb{R}$ convexa. Si x^* es un mínimo local de f sobre un convexo $C \subset D$, entonces x^* es también un mínimo global de f sobre C .*

Demostración. Como x^* es mínimo local, para cualquier $y \in C$, podemos elegir un $t > 0$ suficientemente pequeño tal que

$$f(x^*) \leq f(x^* + t(y - x^*)).$$

Como además f es convexa, tenemos

$$f(x^* + t(y - x^*)) = f(ty + (1 - t)x^*) \leq tf(y) + (1 - t)f(x^*).$$

Juntando ambas expresiones, se llega a

$$f(x^*) \leq tf(y) + (1 - t)f(x^*),$$

lo que implica que $f(x^*) \leq f(y)$. Como y era un punto arbitrario de C , hemos probado que x^* es un mínimo global de f sobre C . \square

Daremos ahora una caracterización de estos mínimos para el problema de optimización convexa.

Teorema 1.16. *Sea $f : D \rightarrow \mathbb{R}$ convexa y diferenciable. Entonces, x^* es un mínimo (global) de f sobre el conjunto convexo $C \subset D$ si y solo si*

$$\langle \nabla f(x^*), (x - x^*) \rangle \geq 0, \text{ para todo } x \in C.$$

*Demuestra*ción. En primer lugar, supongamos que existe $x \in C$ tal que $\langle \nabla f(x^*), (x - x^*) \rangle < 0$. Entonces, $(x - x^*)$ es una dirección de descenso local. Luego, existe $t \in (0, 1)$ tal que

$$f(x^*) > f(x^* + t(x - x^*)) = f(tx + (1 - t)x^*).$$

Como $tx + (1 - t)x^* \in C$ por convexidad, x^* no sería mínimo global en C .

Para el recíproco, como f es convexa, por el Teorema 1.12 tenemos

$$f(x) \geq f(x^*) + \langle \nabla f(x^*), (x - x^*) \rangle \text{ para todo } x \in C.$$

Luego, si $\langle \nabla f(x^*), (x - x^*) \rangle \geq 0$, tenemos

$$f(x) \geq f(x^*) + \langle \nabla f(x^*), (x - x^*) \rangle \geq f(x^*).$$

Luego x^* es un mínimo. \square

1.2.4. Biconvexidad

Más adelante, la función que vamos a tratar de minimizar no es convexa sobre el conjunto total de las variables, pero sí es convexa en cada una de ellas fijando el resto. Definimos a continuación este concepto.

Para lo que sigue, consideramos $X \subseteq \mathbb{R}^n$ e $Y \subseteq \mathbb{R}^m$ dos conjuntos convexos no vacíos, y $B \subseteq X \times Y$ otro conjunto.

Definición 1.17. Definimos las x -secciones y las y -secciones de B como sigue:

$$\begin{aligned} B_x &:= \{y \in Y : (x, y) \in B\}, \\ B_y &:= \{x \in X : (x, y) \in B\}. \end{aligned}$$

Definición 1.18. El conjunto $B \subseteq X \times Y$ se dice *biconvexo* en $X \times Y$ si B_x es convexo para cada $x \in X$ y B_y es convexo para cada $y \in Y$.

Definición 1.19. Una función $f : B \rightarrow \mathbb{R}$ es *biconvexa* en B si para cada $x \in X$

$$f_x(\bullet) := f(x, \bullet) : B_x \rightarrow \mathbb{R}$$

es una función convexa, y para cada $y \in Y$

$$f_y(\bullet) := f(\bullet, y) : B_y \rightarrow \mathbb{R}$$

es una función convexa.

Como se observa en [20], las funciones biconexas pueden tener mínimos locales, al tratarse de un problema de optimización muy general. En este mismo artículo se menciona que sin imponer condiciones adicionales sobre la función a optimizar, lo máximo a lo que se puede aspirar es a encontrar óptimos parciales, definidos a continuación.

Definición 1.20. Sea $f : B \rightarrow \mathbb{R}$ una función biconvexa y sea $(x^*, y^*) \in B$. Se dice que (x^*, y^*) es un *óptimo parcial* de f en B si

$$f(x^*, y^*) \leq f(x, y^*) \quad \forall x \in B_{y^*} \text{ y } f(x^*, y^*) \leq f(x^*, y) \quad \forall y \in B_{x^*}.$$

En [20] se demuestra que un algoritmo que converge a estos óptimos parciales es el siguiente.

Definición 1.21. [Algoritmo de optimización convexa alternante] Fijado un punto inicial (x^*, y^*) , se itera como sigue:

1. $x^* := \operatorname{argmin}_{x \in B_{y^*}} f(x, y^*)$
2. $y^* := \operatorname{argmin}_{y \in B_{x^*}} f(x^*, y)$

Por biconvexidad, ambos pasos son un problema de optimización convexa. Además, el orden de estos pasos es obviamente intercambiable.

1.2.5. Descenso por gradiente en optimización convexa

Cuando no existe una solución cerrada para un problema de optimización, una técnica bastante utilizada es la de *descenso por gradiente*. Esta se basa en información local de primer orden para aproximarse iterativamente a un mínimo. En cada iteración, se toman pasos en la dirección de máximo crecimiento, la opuesta al gradiente: $-\nabla f(x)$. El mínimo al que lleguemos no tiene porque ser global en general, aunque, como ya hemos visto, en optimización convexa sí debe serlo.

Definimos a continuación este método en general y las adaptaciones que son necesarias para aplicarlos en la resolución de problemas de optimización convexa.

Definición 1.22. [Algoritmo de descenso por gradiente general] Partiendo de una función f diferenciable y un valor inicial x_0 arbitrario en su dominio, se itera en base a la siguiente regla:

$$x_{k+1} := x_k - \alpha_k \nabla f(x_k), \quad \alpha_k \in \mathbb{R}^+,$$

hasta que cierto criterio de convergencia se cumpla. Por ejemplo, si $\frac{\|x_{k+1} - x_k\|}{\|x_k\|} < \epsilon$ ó $\nabla f(x_k) < \epsilon$.

El parámetro α_k es el tamaño del paso que damos en cada iteración y se suele conocer como *tasa de aprendizaje* (en inglés *learning rate*). Hay multitud de formas de inicializar este parámetro, pero una deseable sería la que hiciera que avanzásemos hasta el mínimo valor de la función en la dirección opuesta al gradiente:

$$\alpha_k := \operatorname{argmin}_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k)).$$

Calcular dicho α_k constituye un nuevo problema de optimización (de hecho, convexa) que añadiría un coste extra por cada iteración. Para simplificar el cálculo de α_k , una estrategia común suele ser definirlo como una función inversamente proporcional al número de iteraciones.

Una vez hemos definido el algoritmo general, es momento de adaptarlo para resolver problemas de optimización convexa. Recordemos que nuestro objetivo es

$$\min_{x \in C} f(x),$$

luego la solución que demos debe estar en C .

Para empezar, f debe ser diferenciable para poder calcular su gradiente. Pero además, con el método que hemos descrito, es posible que el algoritmo nos guíe a soluciones fuera de este conjunto C , luego debemos adaptarlo para que cumpla esta restricción. Proponemos a continuación dos adaptaciones.

Para la primera de ellas, necesitamos la siguiente herramienta matemática:

Definición 1.23. La *proyección* de un punto y sobre C se define como

$$P_C(y) := \operatorname{argmin}_{x \in C} d(y, x),$$

para una distancia d dada.

Con esto, podemos definir una primera adaptación del algoritmo general que asegura que se cumplen las restricciones de la optimización convexa:

Definición 1.24. [Algoritmo de descenso por gradiente proyectado] Partiendo de una función f diferenciable y un valor inicial $x_0 \in C$ arbitrario, se itera en base a la siguiente regla:

$$x_{k+1} := P_C(x_k - \alpha_k \nabla f(x_k)), \quad \alpha_k \in \mathbb{R}^+.$$

La idea detrás de este algoritmo es avanzar en la dirección de máximo decrecimiento de f hasta donde nos permita llegar el conjunto C . Si bien esta versión resuelve el problema de optimización convexa, requiere de una operación en ocasiones costosa como es la proyección.

La segunda alternativa se basa también en restringir los movimientos de cada iteración al conjunto C en cuestión. Para ello, buscamos la dirección de máximo crecimiento del gradiente estudiando direcciones únicamente entre puntos del conjunto (a las que llamaremos *direcciones factibles*), para después desplazarnos hacia ella sin salirnos de C usando que es convexo. Este algoritmo fue descrito por primera vez en 1956 por los doctorandos de Princeton, Marguerite Frank y Philip Wolfe en [17], y fue revisitado por Jaggi en 2013 [24].

Definición 1.25. [Algoritmo de Frank-Wolfe] Partiendo de una función f diferenciable y un valor inicial $x_0 \in C$ arbitrario, se itera en base a la siguiente regla:

$$s_k := \underset{s \in C}{\operatorname{argmin}} \langle \nabla f(x_k), s \rangle$$

$$x_{k+1} := x_k + \alpha_k(s_k - x_k), \quad \alpha_k \in [0, 1].$$

Cabe hacer ciertas observaciones:

- Al calcular s_k , estamos obteniendo la dirección factible de máximo decrecimiento del gradiente desde x_k . Esto es porque

$$\underset{s \in C}{\operatorname{argmin}} \langle \nabla f(x_k), s \rangle = \underset{s \in C}{\operatorname{argmin}} \langle \nabla f(x_k), (s - x_k) \rangle.$$

En el algoritmo general, esta dirección es directamente la opuesta al gradiente, pero no podemos usarla ya que pueden no existir puntos de C a los que llegar desde x_k siguiéndola.

- En cada paso, $x_{k+1} \in C$ por convexidad, ya que

$$x_k + \alpha_k(s_k - x_k) = \alpha_k s_k + (1 - \alpha_k)x_k, \quad x_k, s_k \in C, \quad \alpha_k \in [0, 1].$$

- Se suele tomar $\alpha_k = \frac{2}{k+2}$, con $k = 0, 1, \dots$
- El coste extra de este algoritmo reside en encontrar la dirección factible de máximo decrecimiento del gradiente en C . Como veremos, para ciertos C esta operación es menos costosa que la proyección P_C .

CAPÍTULO 2

Descripción del método original

En este capítulo se va a presentar el método descrito por Cutler y Breiman en su trabajo seminal sobre el análisis de arquetipos [11]. Veremos qué motiva la propuesta del análisis de arquetipos, dónde se encuentran los denominados arquetipos en el conjunto de datos y el algoritmo de cálculo que se propuso originalmente.

2.1. Motivación y definición

En [11], los autores comienzan exponiendo una de las desventajas principales que tienen algunos métodos de reducción de dimensionalidad como *Principal Component Analysis* (PCA): la dificultad para interpretar los resultados obtenidos. Para ilustrar esta idea, se describe un problema presentado por Flury y Riedwyl en [16]: se tienen las medidas de las cabezas de 200 soldados del ejército suizo, y se quiere generar las medidas de unas pocas cabezas “principales” para fabricar máscaras de gas válidas para todo el ejército. Este conjunto de datos está disponible en R. Véase [22, 3].

Con este objetivo, como estos patrones principales deben representar a la totalidad de los datos de la muestra, un primer enfoque es que cada dato pueda ser aproximado como combinación de los patrones principales. Para esto, una opción viene dada por una variante de PCA. Sean x_1, \dots, x_n los datos de la muestra, definimos los patrones principales como:

$$z_1^*, \dots, z_p^* = \underset{z_1, \dots, z_p}{\operatorname{argmin}} \sum_{i=1}^n \left\| x_i - \sum_{k=1}^p \alpha_{ik} z_k \right\|^2,$$

donde $p \ll n$.

Se puede demostrar que estos z_1^*, \dots, z_p^* corresponden a las componentes principales del conjunto de datos. Para ello, se usan los mismos argumentos que en la definición de PCA. Se recomienda consultar [11, Sección 1], junto a [29, Sección 4.1.1].

Sin embargo, este procedimiento produce resultados difícilmente interpretables, además de poco verosímiles en muchos casos. Por ejemplo, como comentan los autores, se obtienen distancias negativas entre puntos de un mismo patrón principal. Esto es porque en ningún momento se ha forzado a que estos patrones sean verosímiles, es decir, que puedan pertenecer a la población real de la que provienen los datos.

Para solventar este problema, se impone que los patrones principales (*arquetipos* a partir de ahora) sean combinaciones convexas de los datos de la muestra. Es decir,

$$z_k = \sum_{j=1}^n \beta_{kj} x_j,$$

con $\sum_{j=1}^n \beta_{kj} = 1$ y $\beta_{kj} \geq 0$.

De nuevo, el objetivo es estudiar qué combinaciones de estos arquetipos puedan representar a la totalidad de los datos de la muestra. Se buscan entonces α_{ik} que minimicen

$$\left\| x_i - \sum_{k=1}^p \alpha_{ik} z_k \right\|^2.$$

Vuelve a tener sentido imponer que estas combinaciones sean convexas ($\sum_{k=1}^p \alpha_{ik} = 1$ y $\alpha_{ik} \geq 0$), dado que una vez más obtenemos resultados interpretables. Por ejemplo, podemos ver los coeficientes resultantes como probabilidades:

$$\alpha_{ik} = \mathbb{P}(x_i \in [z_k]),$$

donde $[z_k]$ denota una subpoblación o clase representada por dicho arquetipo.

Podemos ya definir formalmente los arquetipos de un conjunto de datos.

Definición 2.1. Sea x_1, \dots, x_n un conjunto de datos. Entonces, los *arquetipos* z_1^*, \dots, z_p^* correspondientes son aquellos que minizan

$$(2.1) \quad \text{RSS} = \sum_{i=1}^n \left\| x_i - \sum_{k=1}^p \alpha_{ik} z_k \right\|^2 = \sum_{i=1}^n \left\| x_i - \sum_{k=1}^p \alpha_{ik} \sum_{j=1}^n \beta_{kj} x_j \right\|^2,$$

con $\sum_{k=1}^p \alpha_{ik} = 1$, $\sum_{j=1}^n \beta_{kj} = 1$, $\alpha_{ik}, \beta_{kj} \geq 0$, $p \ll n$.

A partir de ahora, para expresar la condición de que los pesos forman una combinación convexa, usaremos el concepto de n -simplex (Definición 1.8). Es decir, en la definición anterior se tiene $\alpha_i \in \Delta^{p-1}$, $\beta_k \in \Delta^{n-1}$, con $\alpha_i = (\alpha_{i1}, \dots, \alpha_{in})$ y $\beta_k = (\beta_{k1}, \dots, \beta_{kn})$, $i = 1, \dots, n$, $k = 1, \dots, p$.

Por último, cabe observar que el conjunto de arquetipos no es necesariamente único.

2.2. Localización de los arquetipos

Los Teoremas 1.9 y 1.10 nos dan una primera intuición (acertada) de cuáles son y dónde se localizan los arquetipos de un conjunto de datos. Ambos nos dicen que un conjunto convexo se puede expresar como la combinación convexa de únicamente sus puntos extremos. Por ello, si transformamos nuestro conjunto de datos en un conjunto convexo C (vía cierre convexo) y tomamos $p = \text{card}(\text{Ext}(C))$, los arquetipos serán los

puntos extremos del mismo, ya que son capaces de expresar a través de combinaciones convexas cualquier punto del conjunto de datos con RSS = 0.

En el artículo original, los autores enuncian la siguiente proposición, en línea con lo que acabamos de razonar.

Proposición 2.2. *Sean $x_1, \dots, x_n \in \mathbb{R}^d$ y $C = \text{conv}(\{x_1, \dots, x_n\})$ y $p \in \mathbb{N}$ el número de arquetipos elegido. Consideramos $N = \text{card}(\text{Ext}(C))$. Se tiene:*

1. Si $p = 1$, $z_1 = \bar{x}$ (media de $\{x_1, \dots, x_n\}$) es el único arquetipo.
2. Si $1 < p < N$, entonces existe un conjunto de arquetipos $\{z_1, \dots, z_p\} \subset \partial C$.
3. Si $p = N$, entonces $\text{Ext}(C) = \{z_1, \dots, z_p\}$ es un conjunto de arquetipos.

Demostración. En primer lugar, para los tres casos es trivial que los arquetipos propuestos son combinaciones convexas del conjunto de datos.

1. Es bien sabido que la media, \bar{x} , es el minimizador sin restricciones de RSS cuando $p = 1$.
2. Sea $\{z_1, \dots, z_p\}$ un conjunto de arquetipos (que no tiene que ser único). Supongamos que $z_1 \in \text{int } C$. Tomamos z_j con $j \neq 1$ otro arquetipo cualquiera y definimos

$$z(t) = tz_1 + (1 - t)z_j, \text{ para } t > 1.$$

Consideramos $t_0 > 1$ tal que $z(t_0) \in \partial C$. Observamos que

$$z_1 = \frac{1}{t_0}z(t_0) + \left(1 - \frac{1}{t_0}\right)z_j \in \text{conv}(\{z(t_0), z_j\}).$$

Por tanto, $\text{conv}(\{z_1, \dots, z_p\}) \subseteq \text{conv}(\{z(t_0), \dots, z_p\})$, con $z(t_0) \in \partial C$. Además, $\{z(t_0), \dots, z_p\}$ minimizará también RSS, luego es un conjunto de arquetipos. Podemos repetir este procedimiento con cualquier arquetipo del conjunto $\{z_1, \dots, z_p\}$ que esté en $\text{int } C$ y obtenemos un nuevo conjunto de arquetipos en ∂C .

3. Por el Teorema 1.9 sabemos que $C = \text{conv}(\text{Ext}(C))$, luego RSS = 0 y $\text{Ext}(C)$ es un conjunto de arquetipos como queríamos.

□

Obsérvese que en el punto 2 no se puede intercambiar ∂C por $\text{Ext}(C)$, ya que si bien tenemos garantizado que con esa recta llegaremos a un punto de ∂C , no podemos asegurar que este punto esté en $\text{Ext}(C)$.

Por otro lado, en el artículo [11] se enuncia la proposición sobre $\{x_1, \dots, x_n\} \cap \partial(C)$, pero nos es suficiente con $\text{Ext}(C)$ para conseguir RSS = 0. Por las Proposiciones 1.6 y 1.7, $\text{Ext}(C) \subseteq \{x_1, \dots, x_n\} \cap \partial(C)$, pero esta inclusión puede ser estricta si, por ejemplo, hay tres puntos en $\{x_1, \dots, x_n\} \cap \partial(C)$ contenidos en una misma recta. Por tanto, nuestra nueva condición requiere de menos puntos.

Con el fin de visualizar la localización que acabamos de describir, se han calculado tres arquetipos de una nube de puntos usando el paquete de R [14]. Los resultados se recogen en la Figura 2.1. En primer lugar, en la Figura 2.1(a) observamos que los arquetipos (puntos rojos) están situados en la frontera y que su cierre convexo intenta aproximar el cierre convexo del conjunto total. Por otro lado, en la Figura 2.1(b) podemos ver cómo los arquetipos aproximan el conjunto de datos. Recordemos que el cierre convexo de C se define como el conjunto de combinaciones convexas de elementos de C , por lo que cualquier punto que pertenezca al cierre convexo de los arquetipos será expresado sin error por una combinación convexa de los mismos. Sin embargo, los puntos fuera del cierre convexo deberán ser aproximados por el punto más cercano del cierre convexo (proyección). De hecho, RSS es la suma de distancias de estos puntos a sus respectivas proyecciones. En la Figura 2.1(b) vemos en verde la representación que otorgan los arquetipos. Para puntos ya pertenecientes al cierre convexo, el punto verde coincide con el original; mientras que para puntos fuera del mismo vemos la línea que une el punto original con el punto verde proyectado.

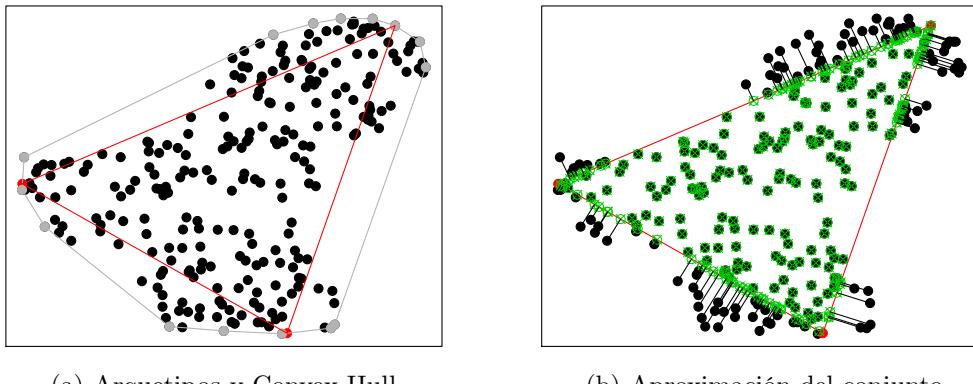


Figura 2.1: Ejemplo de arquetipos y sus aproximaciones para una nube de puntos. A la izquierda, en gris, el cierre convexo de los puntos. En rojo, los arquetipos y su cierre convexo. A la derecha, en verde, los puntos proyectados.

Para incidir en la idea de que cuántos más arquetipos consideremos, mejor aproximaremos el *Convex Hull* y por tanto menor será RSS, se ha calculado el error sobre el conjunto de datos de la Figura 2.1 al utilizar entre 1 y 5 arquetipos. La curva decreciente del error se observa en la Figura 2.2.

2.3. Cálculo de los arquetipos

Como hemos comentado anteriormente, para obtener los arquetipos debemos minimizar la suma RSS en (2.1) buscando los valores óptimos de α y β . Este problema es resoluble por optimizadores generales de mínimos cuadrados, pero en la práctica resulta impráctico. A continuación, describimos el método de cálculo de 1994, basándonos en las ideas propuestas en el artículo original, así como en los detalles que comentan en [13] los desarrolladores de la librería de arquetipos de R [14].

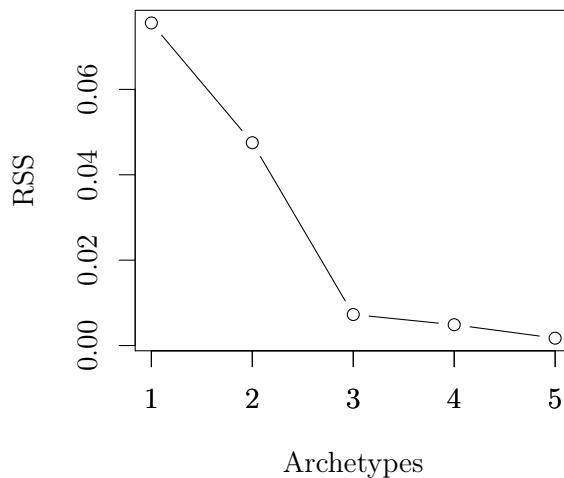


Figura 2.2: Evolución de RSS en función del número de arquetipos.

En primer lugar, para facilitar la comprensión e inspirándonos en [13, 5, 30, 4] usaremos notación matricial. Sea $X \in M_{n \times m}$ la matriz de datos (n datos en dimensión m). Sean $A \in M_{n \times p}$, $B \in M_{p \times n}$ las matrices cuyas filas son α_i y β_k , respectivamente. Con esto, los arquetipos son $BX = Z \in M_{p \times m}$ y el objetivo a minimizar es

$$\text{RSS} = \|X - ABX\|^2,$$

donde la norma matricial es la de Frobenius.

Se observa que RSS no es convexa en el producto AB , ya que en la bibliografía se presentan varios ejemplos en los que hay mínimos locales distintos (lo que contradice el Teorema 1.15). Sin embargo, RSS sí es convexa en A y en B por separado (fijando la otra matriz). Es decir, tal y como enunciamos en la Definición 1.19, RSS es *biconvexa*. La prueba se basa en obtener las matrices hessianas de RSS para A y para B , demostrando que son semidefinidas positivas y usando el Teorema 1.13 para concluir que RSS es convexa en ambos casos. Se sugiere consultar [11, Sección 5] y/o [30, Sección 2] para la demostración completa.

Como comentamos en la Sección 1.2.4, un algoritmo para optimizar una función biconvexa se basa en iterar sobre problemas de optimización convexa en cada variable (Definición 1.21). Esta metodología de optimización alternante es la que propusieron en [11], que es la siguiente:

1. Para un conjunto fijo de arquetipos Z , encontrar α_i que minimicen

$$\|x_i - \alpha_i Z\|^2,$$

con $\alpha_i \in \Delta^{p-1}$.

2. Recalcular los arquetipos \tilde{Z} para los α_i encontrados, resolviendo $X = A\tilde{Z}$.
3. Para \tilde{Z} , encontrar los β_k que minimicen

$$\|\tilde{z}_k - \beta_k X\|^2,$$

con $\beta_k \in \Delta^{n-1}$.

4. Redefinir los arquetipos $Z = BX$ para los β_k encontrados.
5. Recalcular RSS. Si es suficientemente pequeña o se ha alcanzado el límite de iteraciones, terminar. Si no, volver al punto 1.

Para resolver los problemas de optimización convexa 1 y 3, los autores proponen usar un algoritmo de optimización de mínimos cuadrados [28], añadiendo una constante de penalización que fuerce a que se cumpla $\alpha_i \in \Delta^{p-1}$, $\beta_k \in \Delta^{n-1}$. Es decir, si queremos resolver

$$\|u - wT\|^2,$$

con $w \in \Delta$, se resuelve, bajo restricciones sobre w de no negatividad,

$$\|u - wT\|^2 + M^2 \|1 - w\|^2.$$

La bibliografía posterior también comparte la técnica de la optimización alternante, aunque, como veremos en el siguiente capítulo, se proponen métodos más eficientes para resolver los dos subproblemas de optimización convexa.

CAPÍTULO 3

Métodos mejorados de cálculo de arquetipos

En este capítulo estudiaremos dos métodos que calculan los arquetipos de forma más eficiente. Como se comentó en el capítulo anterior, los autores propusieron resolver el problema de optimización con métodos de mínimos cuadrados imponiendo las restricciones convexas por penalización. Esta idea funciona, pero es evidente que no es eficiente imponer las restricciones de esa manera.

3.1. Cálculo del gradiente

Ambos métodos se basan en descenso por gradiente, por lo que el primer paso es obtener el gradiente de la función a optimizar, RSS.

Usando de nuevo notación matricial, recordemos que

$$\text{RSS} = \|X - ABX\|^2.$$

Para poder derivar esta expresión, usaremos propiedades de la derivada de la traza, por lo que la reescribimos como sigue:

$$\begin{aligned} \text{RSS} &= \|X - ABX\|^2 \\ &= \text{Tr}((X - ABX)^T(X - ABX)) \\ &= \text{Tr}(X^T X - X^T B^T A^T X - X^T ABX + X^T B^T A^T ABX) \\ (3.1) \quad &= \text{Tr}(X^T X - 2X^T ABX + X^T B^T A^T ABX) \end{aligned}$$

donde en el paso (3.1) hemos usado que la traza es un operador lineal invariante a trasponer.

Las propiedades de la derivada de la traza que vamos a usar han sido obtenidas de [33, Sección 2.5] y son las siguientes:

$$(3.2) \quad \frac{\partial}{\partial X} \text{Tr}(AXB) = A^T B^T$$

$$(3.3) \quad \frac{\partial}{\partial X} \text{Tr}(A^T X^T B X A) = B^T X A A^T + B X A A^T.$$

Aplicando estas propiedades sobre la derivada de (3.1) obtenemos los gradientes buscados:

$$(3.4) \quad \nabla_A \text{RSS} = \nabla_A = 2(ABXX^T B^T - XX^T B^T) = 2(AZZ^T - XZ^T)$$

$$(3.5) \quad \nabla_B \text{RSS} = \nabla_B = 2(A^T ABXX^T - A^T XX^T).$$

Con esta información, seríamos capaces de intentar resolver el problema de optimización sin restricciones con descenso por gradiente. Sin embargo, como explicamos en la Sección 1.2.5, es necesario realizar ciertas adaptaciones para mantener las restricciones convexas.

3.2. Descripción de los métodos mejorados

3.2.1. Gradiente proyectado

El primer método es el de gradiente proyectado. Esta idea, que ya dimos en la Definición 1.24, es en la que se basa la propuesta de [30].

En primer lugar, es necesario definir la proyección al simplex Δ correspondiente, de forma que los parámetros cumplan las restricciones de formar una combinación convexa. Dicha proyección, para $x \in \mathbb{R}^n$, se define como:

$$y = P_{\Delta^{n-1}}(x) = \frac{\tilde{x}}{\sum_{i=1}^n \tilde{x}_i},$$

con $\tilde{x}_i = \max(x_i, 0)$. De esta forma, aseguramos que $y_i \geq 0$, $\sum_{i=1}^n y_i = 1$ (i.e., $y \in \Delta^{n-1}$).

Definimos además la proyección sobre una matriz como la proyección aplicada a cada una de sus columnas.

Siguiendo lo descrito en la Definición 1.24 y el gradiente que hemos obtenido en (3.4) y (3.5), podemos ya definir el algoritmo de descenso por gradiente proyectado. Partiendo de una inicialización A_0 y B_0 , se opera como sigue:

1. $A_{t+1} = A_t - \mu_A \nabla_{A_t}$,
2. $A_{t+1} = P_{\Delta^{p-1}}(A_{t+1})$,
3. $B_{t+1} = B_t - \mu_B \nabla_{B_t}$,
4. $B_{t+1} = P_{\Delta^{n-1}}(B_{t+1})$,
5. $Z_{t+1} = B_{t+1}X$,
6. Recalcular RSS. Si es suficientemente pequeña o se ha alcanzado el límite de iteraciones, terminar. Si no, volver al punto 1.

En [30] se comentan además distintas formas de inicializar A_0 y B_0 . En nuestro caso, como estamos interesados en estudiar la eficiencia de los métodos de cálculo, dejamos al margen la inicialización y la mantendremos lo más simple posible con matrices diagonales con 1s.

Por otro lado, cabe comentar que en [30] la implementación que usan del gradiente proyectado es ligeramente distinta. Esto es porque adaptan el gradiente de RSS usando la regla de la cadena para tener en cuenta las operaciones realizadas en la proyección. En [30, Sección 2.2] pueden encontrarse los detalles de esta ligera modificación.

3.2.2. Adaptación del Algoritmo de Frank-Wolfe

El otro método que mejora la propuesta inicial es hacer uso del algoritmo de Frank-Wolfe, descrito en la Definición 1.25. Los autores de [4] detallan la adaptación de este algoritmo al problema del cálculo de arquetipos.

En primer lugar, recordemos que era necesario calcular la mejor *dirección factible* en cada paso. Resulta que en el caso del simplex este cálculo es muy sencillo. Tal y como se demuestra en [9], la dirección de máximo decrecimiento es siempre hacia el vértice del simplex que minimiza el gradiente.

Es decir, tomando la base canónica $\{e(i) : e(i)_i = 1 \wedge e(i)_j = 0 \text{ } i \neq j\}$, la dirección factible en el paso k es $e(i')$, con $i' = \operatorname{argmin}_i (\nabla f(x_k))_i$.

Con esto, podemos definir el algoritmo propuesto. En primer lugar, como describen en [9], se inicializa el vector a optimizar a uno de los vértices del simplex. En nuestro caso, inicializaremos todas las filas de la matriz que estemos optimizando a $e(1)$. Entonces, si queremos optimizar la matriz A con filas α_i , el procedimiento es el siguiente:

1. $\alpha_i = e(1)$, para todo i .
2. Se recalcula ∇_A .
3. Para cada $i \in \{1, \dots, n\}$:
 - a) $j = \operatorname{argmin}_j (\nabla_A)_{ij}$.
 - b) $\alpha_i = \alpha_i + \frac{2}{t+2} (e(j) - \alpha_i)$.
4. Si las actualizaciones son suficientemente pequeñas o se ha alcanzado el límite de iteraciones, terminar. Si no, volver al punto 2.

El procedimiento es el mismo para la optimización de la matriz B . Siguiendo el método de optimización alternante, conseguimos minimizar RSS.

3.3. Comparación de los métodos

Para hacernos una idea de la mayor eficiencia de los métodos descritos en la sección anterior con respecto al método original, se han realizado pruebas computacionales

para diversos tamaños de datos de entrada. Para ello, se han implementado los tres algoritmos tanto en R como en Python. Dicha implementación puede consultarse en el repositorio público [10]. Las pruebas de tiempo que presentamos a continuación se han realizado con la implementación en Python, dado que los tiempos de ejecución de los algoritmos son entre dos y tres veces más rápidos que en R. Diversas pruebas se han ejecutado para determinar la causa, llegando a la conclusión de que es la multiplicación de cadenas de matrices dónde se encuentran las mayores diferencias (de hasta x2). Al parecer esto se debe a que R usa por defecto una versión de BLAS (Basic Linear Algebra Subprograms), la librería encargada de multiplicar matrices, no optimizada, como se comenta en [37].

Las pruebas han consistido en ejecutar los tres algoritmos para matrices aleatorias con todas las combinaciones de los siguientes valores de los parámetros:

- $n_samples$ (n , número de muestras) $\in \{100, 1000, 10000\}$
- $n_features$ (m , número de características) $\in \{5, 10, 25, 50\}$
- $n_archetypes$ (p , número de arquetipos) $\in [1, 10]$

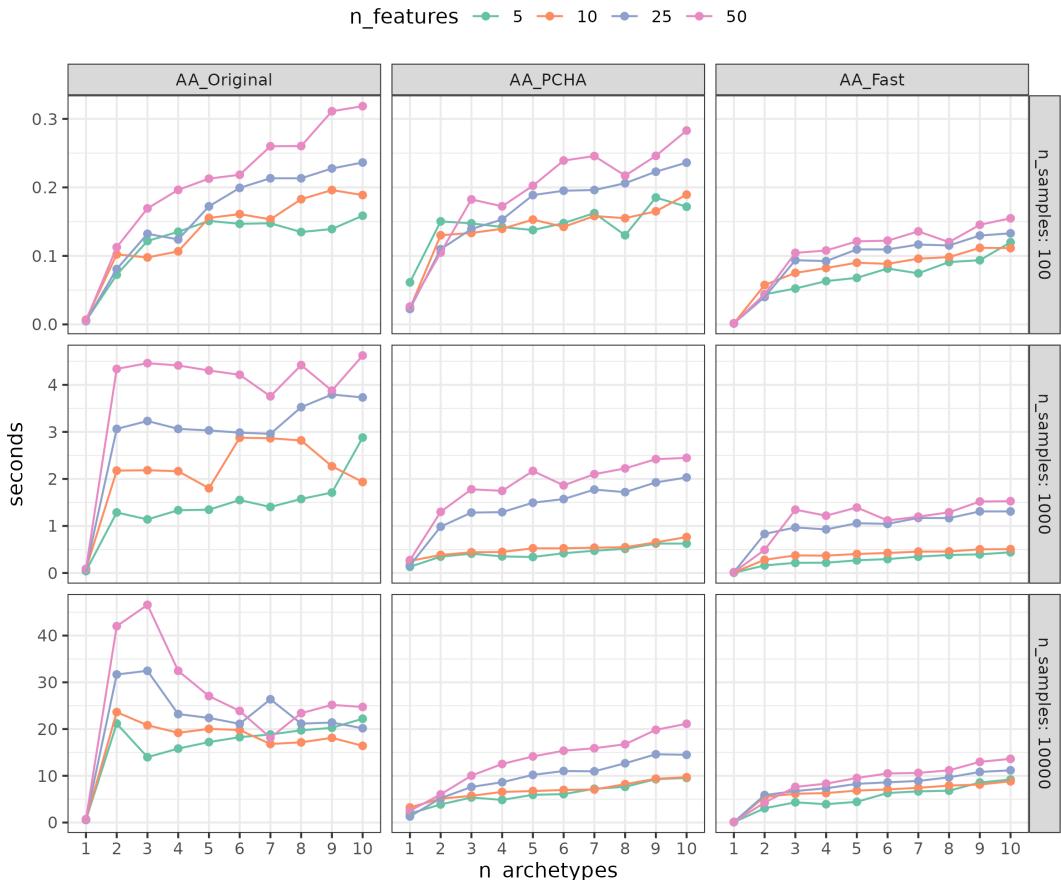


Figura 3.1: Comparación de los tiempos de ejecución.

En la Figura 3.1 observamos que el algoritmo original es muy poco eficiente en comparación con los algoritmos mejorados, demostrando entonces que imponer las restricciones convexas a través de la penalización no es una buena opción. En cuanto a los dos algoritmos mejorados, es evidente que la versión de gradiente proyectado *AA_PCHA* es más lenta en todos los casos que la versión basada en Frank-Wolfe *AA_Fast*. Como dejamos entrever en la definición de ambos métodos, para ciertos conjuntos la proyección podía ser más costosa que encontrar *direcciones factibles*. Parece claro que este es el caso del simplex Δ .

Si bien el tiempo de ejecución es algo a tener en cuenta, también es necesario que estos métodos converjan a una buena solución. Por ello, evaluamos a continuación la evolución de RSS (normalizado por el número de datos) para los mismos experimentos que hemos realizado para el tiempo.

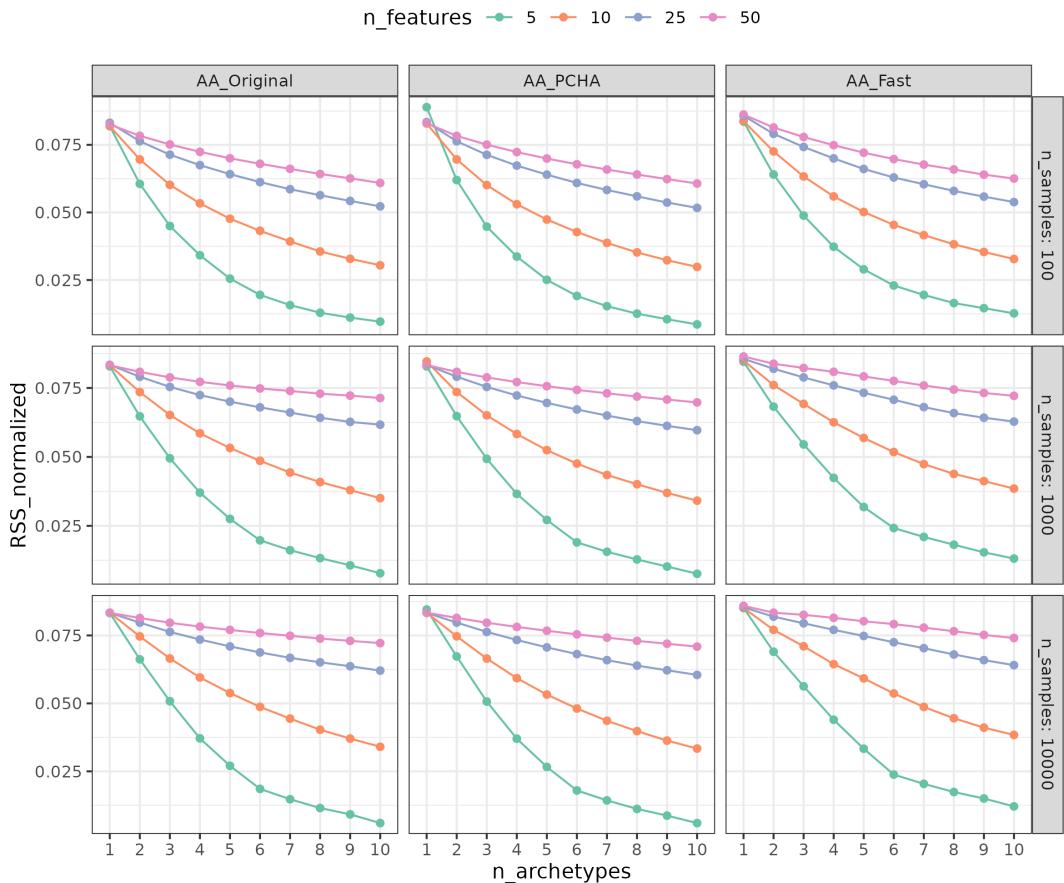


Figura 3.2: Comparación de RSS normalizado.

En primer lugar, observamos en la Figura 3.2 cómo, en todos los algoritmos y para todos los tamaños, el RSS se reduce según aumentamos el número de arquetipos. Esto es lógico, ya que cuántos más arquetipos tengamos, mejor vamos a poder aproximar el cierre convexo del conjunto. Por otro lado, también vemos cómo, por lo general, la versión basada en Frank-Wolfe *AA_Fast* alcanza ligeramente peores resultados (mayor

RSS). Podemos atribuir esto a que, si bien es más rápido ahorrarse la proyección, considerar sólo *direcciones factibles* durante la optimización limita la búsqueda de mínimos. No obstante, la diferencia observada no es ni mucho menos tan significativa como la de tiempos de ejecución (véase la Figura 3.1).

A la luz de los resultados presentados, podemos considerar *AA_Fast* como el método de cálculo más conveniente. De hecho, los arquetipos de los distintos ejemplos que se exponen en el siguiente capítulo han sido obtenidos con este algoritmo.

CAPÍTULO 4

Ejemplos reales de análisis de arquetipos

Una vez hemos descrito la base que los sustenta y comprobado cuál es la forma más eficiente de obtenerlos, vamos a comprobar las ventajas que ofrecen los arquetipos frente a otras técnicas en ejemplos reales.

4.1. Reconocimiento facial

Un problema que podemos intentar resolver con el análisis de arquetipos es el de reconocimiento facial. Este se basa en, dadas dos imágenes, determinar si se trata de la misma persona. Técnicas no supervisadas como PCA han sido usadas para este fin, por ejemplo en [43, 42]. El uso de estos métodos busca definir un espacio en el que representar el par de imágenes para poder comparar sus respectivas proyecciones. En el caso de las tres técnicas que vamos a probar, cada una construirá su espacio en base a una serie de imágenes principales, con vistas a que cualquier imagen pueda ser aproximada como combinación de estas.

Comprobaremos a continuación la selección de 25 imágenes principales que propone cada una de los tres métodos a probar: PCA, k-means y análisis de arquetipos. Para ello, ejecutaremos los tres algoritmos sobre el conjunto de datos [23, 36], el cual contiene más de 13 000 imágenes de caras como las que se observan en la Figura 4.1



Figura 4.1: Ejemplos de imágenes contenidas en el conjunto de datos.

En primer lugar, en la Figura 4.2 vemos los resultados de PCA. Las imágenes mostradas son las 25 componentes principales. Las componentes recogen información

sobre los rasgos que más caracterizan a los rostros, como son la nariz, la boca y los ojos. Sin embargo, como se observa, los rostros obtenidos han dejado de ser inteligibles y por tanto interpretables. No se niega que esta representación no sea apropiada para resolver el problema, pero queda patente que perdemos el control sobre la interpretabilidad de los resultados.



Figura 4.2: Resultado de ejecutar PCA sobre el conjunto de datos.

En segundo lugar, en la Figura 4.3 vemos los resultados de k-means. Esta vez, los rostros parecen más humanos que en el caso anterior. No obstante, como siempre sucede cuando calculamos la media de unos datos, esta es una representación “suavizada” de los mismos. De esta forma, al estar obteniendo los rostros medios de la población, perdemos información sobre los detalles. Además, k-means no obtiene las imágenes principales (centroídes) con el objetivo de que su combinación aproxime lo mejor posible al resto de imágenes, como sí hacen PCA y el análisis de arquetipos.

Por último, en la Figura 4.4 vemos los resultados del análisis de arquetipos. En contraste con los dos métodos anteriores, esta vez sí que observamos rostros perfectamente humanos, aunque algo desfigurados. Podemos observar claramente distintos

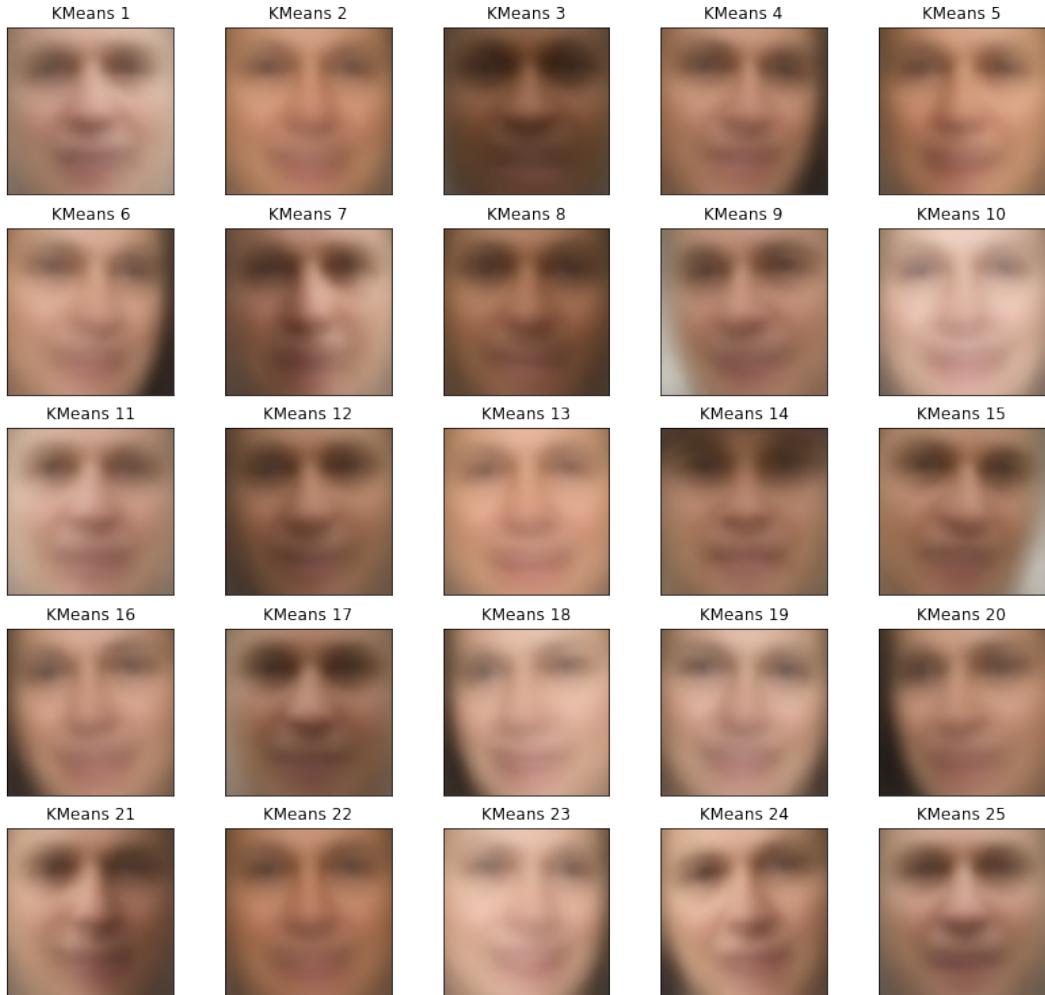


Figura 4.3: Resultado de ejecutar k-means sobre el conjunto de datos.

rasgos faciales entre ellos: bocas abiertas y cerradas, cejas pobladas y finas, bigotes, gafas, etc. Al igual que PCA, las imágenes principales han sido obtenidas para que su combinación aproxime al resto de imágenes. Pero, además, como comentamos en la Sección 2.1 de motivación, el hecho de que estas combinaciones sean convexas las hace interpretables. Para ilustrar este hecho, se ha desarrollado una función que dibuja la distribución de pesos obtenida sobre los arquetipos para un dato dado. Un par de ejemplos pueden verse en la Figura 4.5. Estas gráficas nos dan información de cuánto se asemeja el dato de entrada a cada uno de los arquetipos, pudiendo entonces interpretar las decisiones que tome nuestro modelo de detección facial. Esto es porque, tras obtener los espacios definidos por las imágenes principales, el modelo determinará qué son rostros similares si el conjunto de pesos que expresan cada rostro como combinación de las imágenes principales es similar. Como hemos podido com-

probar, el análisis de arquetipos es el único método que nos permite interpretar tanto las imágenes principales como la distribución de pesos obtenida sobre ellas.

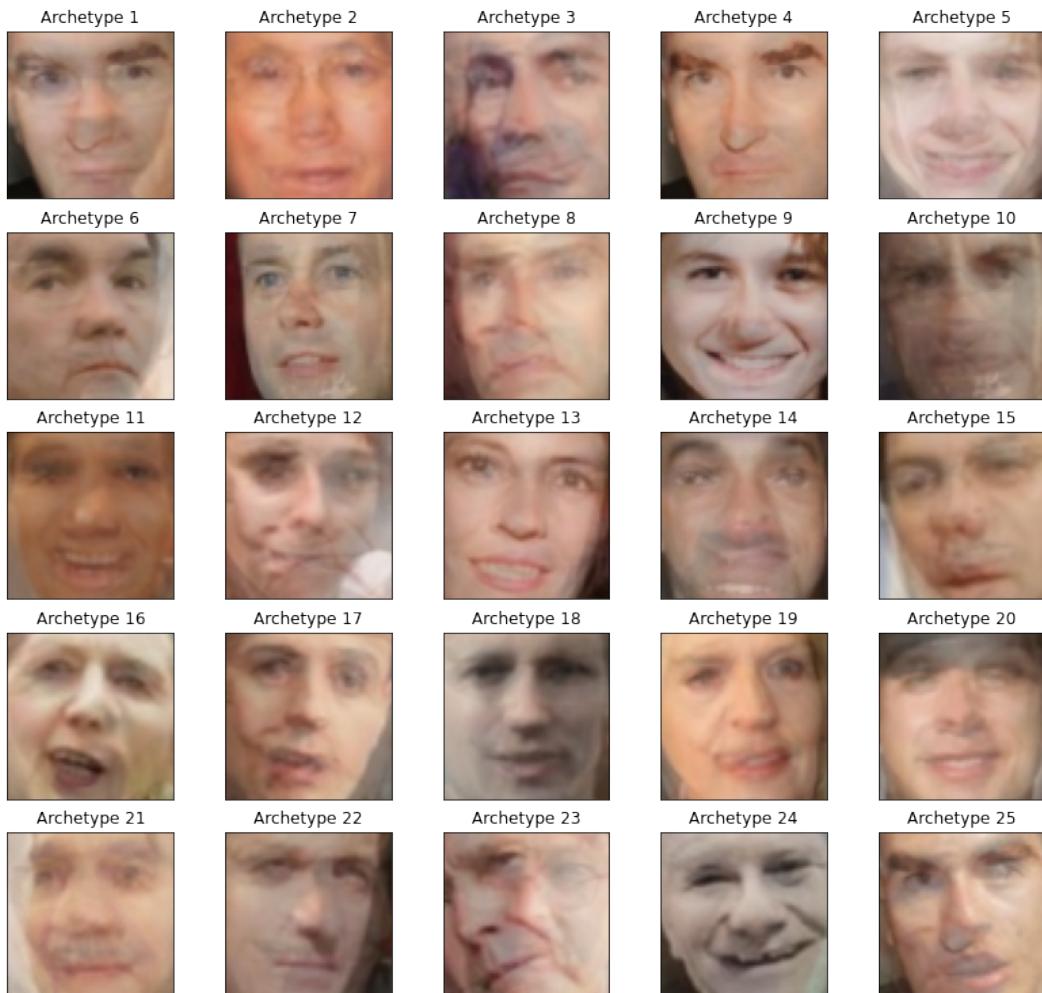


Figura 4.4: Resultado de ejecutar análisis de arquetipos sobre el conjunto de datos.

4.2. Segmentación de clientes

Otro ejemplo dónde podemos observar las ventajas del análisis de arquetipos es en el problema de segmentación de clientes. Este consiste en agrupar los clientes de una empresa que tienen conductas similares con el objetivo de usar esta información para crear campañas de marketing dirigidas, mejorar el diseño de productos específicos o identificar tendencias de consumo. En este caso, vamos a aplicar k-means y análisis de arquetipos sobre el conjunto de datos [32]. Tras una limpieza de los datos, se han obtenido las siguientes variables:

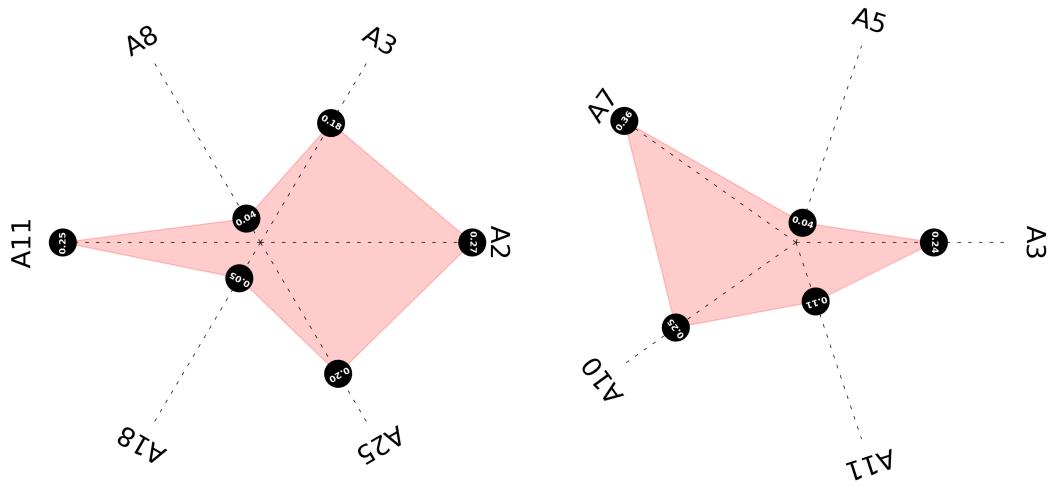


Figura 4.5: Distribuciones de pesos obtenidas sobre las imágenes principales 4.4

- *Age*: edad.
 - *Education_Level*: entero entre 0 y 4 que representa el nivel de educación.
 - *Has_Partner*: variable binaria que indica si el cliente tiene pareja.
 - *Has_Children*: variable binaria que indica si el cliente tiene hijos.
 - *Income*: renta anual del cliente en dólares.
 - *Spending*: cantidad total gastada en dólares.
 - *Food*: cantidad total gastada en comida en dólares.
 - *Wine*: cantidad total gastada en vino en dólares.
 - *Gold*: cantidad total gastada en oro en dólares.
 - *Discounts*: número de compras hechas con descuento.
 - *Seniority*: número de días de antigüedad como cliente.
 - *Recency*: número de días desde la última compra del cliente.

En la Figura 4.6 podemos ver los tres arquetipos y los tres centroídes obtenidos tras la ejecución de sus respectivos algoritmos sobre los datos. Nos referiremos a cada uno de ellos por su índice precedido de A si es arquetipo o K si es centroide. Como las unidades de medida son distintas en cada caso, se presentan los percentiles obtenidos para cada variable.

Observamos ciertas similitudes entre los resultados, aunque también algunas diferencias clave. En primer lugar, parece que ambos algoritmos han obtenido representantes muy similares para las variables demográficas. Así, $A1$ y $K1$ son prácticamente

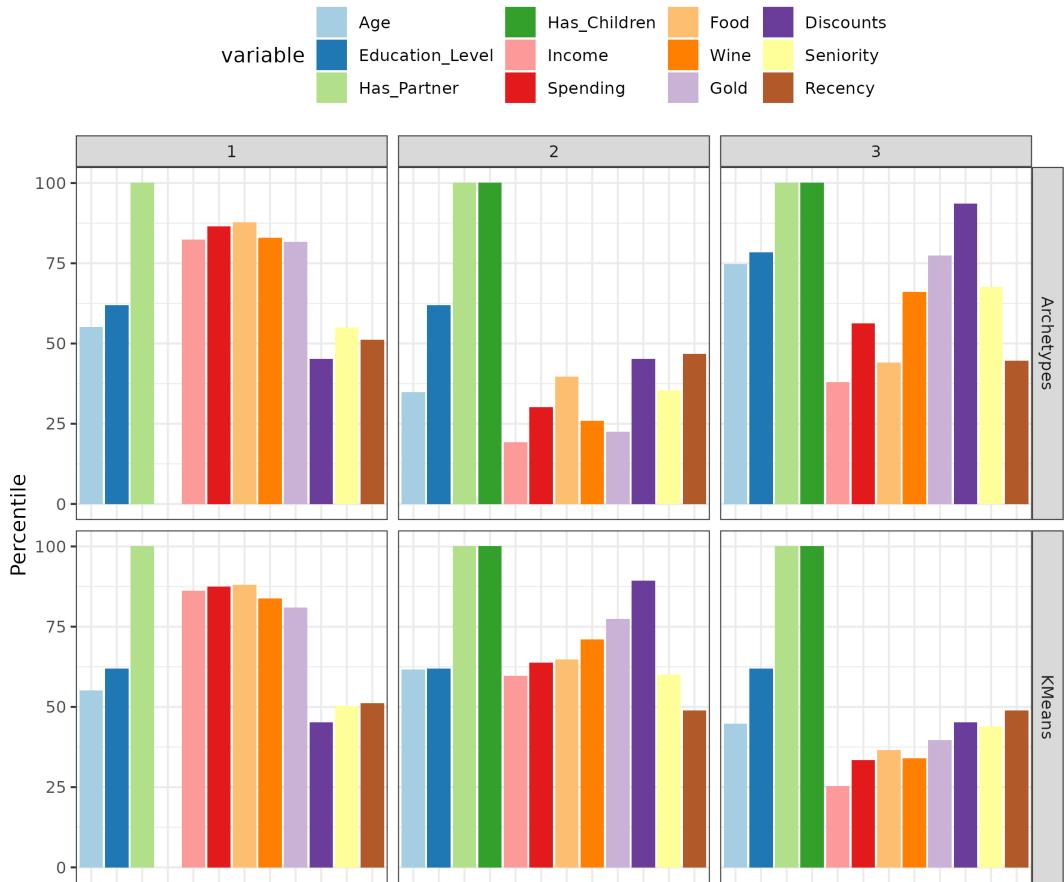


Figura 4.6: Resultado de ejecutar análisis de arquetipos y k-means.

equivalentes en todas las variables, representando a clientes de mediana edad y formación, con pareja pero sin hijos, con elevados ingresos y gastos. Por otro lado, en términos demográficos, A_2 parece estar relacionado con K_3 , y A_3 con K_2 . En el primer caso, nos encontramos a un cliente más joven que la media pero con un nivel de formación medio-alto, con pareja y con hijos. Debido a su juventud, los ingresos y los gastos son bajos. En el segundo caso, son personas de más edad, también con pareja y con hijos, y con mayor poder económico. Comenzamos a observar en ambos casos como los arquetipos contienen valores más extremos, lo cual es sin duda una ventaja para poder realmente distinguir comportamientos entre grupos. Esto queda patente en las variables de consumo. Si bien todas ellas son muy similares en cada centroide, en los arquetipos observamos diferencias importantes que aportan valor. Por ejemplo, A_2 , debido a su menor edad y por tanto menor presupuesto, centra sus gastos mayoritariamente en comida. Por otro lado, A_3 , de mayor edad y poder económico, tiene gastos mucho mayores en artículos menos indispensables como el vino y el oro. Estas conclusiones tan relevantes no pueden ser obtenidas en el caso de los centroides, ya que, como en el ejemplo de las imágenes, la media suaviza los detalles.

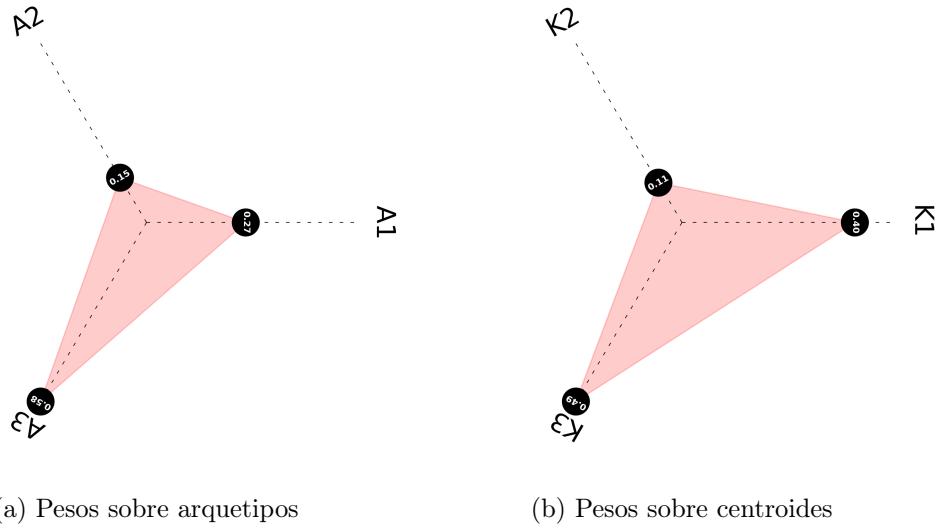


Figura 4.7: Distribuciones de pesos obtenidas para un dato aleatorio de [32].

Los arquetipos no solo nos permiten capturar mejor detalles importantes, sino que también, como ya hemos comentado, nos dan la opción de obtener porcentajes de semejanza fiables e interpretables con otros datos a través de combinaciones convexas. Este hecho permitiría a los equipos de marketing determinar el perfil, o combinaciones de perfiles, al que pertenece un nuevo cliente de la empresa; pudiendo personalizar los productos que se le ofrecen o las estrategias que se le aplican. En la Figura 4.7 mostramos los pesos obtenidos al minimizar la combinación convexa que mejor aproxima un dato aleatorio de la muestra con respecto a arquetipos y centroides. Observamos que los pesos de 4.7(b) son más suaves que los de 4.7(a), posiblemente debido, una vez más, al suavizado de la media implícito en los centroides. Por último, pero no menos importante, el error de la aproximación de k-means es un 11% mayor que el de la aproximación de los arquetipos.

CAPÍTULO 5

Conclusiones y trabajo futuro

A lo largo de este documento, hemos podido comprender profundamente el análisis de arquetipos, tanto sus fundamentos teóricos, como sus ventajas sobre otros métodos similares en situaciones reales. Por el camino, hemos explorado también diversos algoritmos de optimización convexa que nos han llevado a mejorar con creces la propuesta original.

Visto el potencial que ofrece esta técnica, merece la pena comentar posibles líneas de trabajo futuro. En primer lugar, existen ya trabajos que extienden el análisis de arquetipos desde perspectivas variadas. Por un lado, [26] combina el análisis de arquetipos con las redes neuronales profundas. En [39] se propone utilizar suposiciones distribucionales sobre el vector observado para encontrar los arquetipos; mientras [38] adapta el análisis de arquetipos a otro tipo de datos, en este caso cualitativos. Por otro lado, [12] añade robustez al método de cálculo modificando la función de coste, reduciendo así el efecto de *outliers*. Relacionado con esto, [8] usa las proyecciones al espacio generado por el cierre convexo de los arquetipos para detectar *outliers*.

Queda patente que han surgido múltiples líneas de investigación derivadas del concepto del análisis de arquetipos. Por nuestra parte, se propone lo siguiente. Si bien la convexidad nos aporta buenas propiedades para definir los arquetipos, dar por hecho que el soporte de nuestros datos es un conjunto convexo es una suposición demasiado exigente. Buscando relajar esta hipótesis, se pretende explorar formas más generales de definir los arquetipos que nos permitan aproximar soportes más generales. Para ello, proponemos dos posibles puntos de partida. En primer lugar, se puede explorar el concepto de los conjuntos α -convexos, una generalización de la convexidad. En particular, intentaríamos definir los arquetipos generalizados de forma que approximen el α -convex hull, propuesto en [31]. Por otro lado, [18] describe una aproximación local del cierre convexo (*Local convex hull*) por vecinos próximos que también sería interesante tener en cuenta. En cualquier caso, la principal tarea pendiente de la generalización que proponemos es definir una función de error que mida la calidad de la aproximación generada por unos arquetipos dados, ahora que no tenemos combinaciones y cierres convexos a nuestra disposición. Minimizando esta función de error, obtendríamos una serie de puntos que permitirían entender soportes más generales de nuestro conjunto de datos.

Bibliografía

- [1] ANGEL, T. Convex functions. http://www.math.udel.edu/~angell/Opt/conv_fcn.pdf, Accedido 2022-01-03.
- [2] ARORA, S. Cos 521 advanced algorithm design: Lecture19. <https://www.cs.princeton.edu/courses/archive/fall13/cos521/lecnotes/lec19.pdf>, Accedido 2022-01-03, 2013.
- [3] ATKINSON, A. C., RIANI, M., Y CERIOLI, A. Swiss heads R dataset. <https://www.rdocumentation.org/packages/fsdaR/versions/0.4-8/topics/swissheads>, Accedido 2022-01-07.
- [4] BAUCKHAGE, C., KERSTING, K., HOPPE, F., Y THURAU, C. Archetypal analysis as an autoencoder. In *Workshop New Challenges in Neural Computation* (2015), pp. 8–16.
- [5] BAUCKHAGE, C., Y THURAU, C. Making archetypal analysis practical. In *Pattern Recognition, 31st DAGM Symposium* (09 2009), Springer, pp. 272–281.
- [6] BERRENDERO, J. R. Investigación operativa: Conjuntos convexos. <https://verso.mat.uam.es/~joser.berrendero/cursos/Matematicas-I0/io-tema2-16.pdf>, Accedido 2021-11-20.
- [7] BERRENDERO, J. R. Investigación operativa: Funciones convexas y optimización convexa. <https://verso.mat.uam.es/~joser.berrendero/cursos/Matematicas-I0/io-tema4-16.pdf>, Accedido 2022-01-03.
- [8] CABERO, I., EPIFANIO, I., PIÉROLA, A., Y BALLESTER, A. Archetype analysis: A new subspace outlier detection approach. *Knowledge-Based Systems* 217 (2021), 106830.
- [9] CLARKSON, K. L. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Trans. Algorithms* 6, 4 (sep 2010).
- [10] COBO, G. G. Archetypal analysis implementation. <https://github.com/atmguille/archetypal-analysis>, Accedido 2022-05-17.
- [11] CUTLER, A., Y BREIMAN, L. Archetypal analysis. *Technometrics* 36, 4 (1994), 338–347.

- [12] EUGSTER, M. J., Y LEISCH, F. Weighted and robust archetypal analysis. *Computational Statistics & Data Analysis* 55, 3 (2011), 1215–1225.
- [13] EUGSTER, M. J. A., Y LEISCH, F. From Spider-Man to Hero – archetypal analysis in R. *Journal of Statistical Software* 30, 8 (2009), 1–23.
- [14] EUGSTER, M. J. A., LEISCH, F., Y SETH, S. archetypes: Archetypal Analysis R package. <https://cran.r-project.org/package=archetypes>, Accedido 2022-01-07, 2019.
- [15] FERNANDEZ-GRANDA, C. Optimization-based data analysis: Convex optimization. https://cims.nyu.edu/~cfgranda/pages/0BDA_fall17/notes/convex_optimization.pdf, Accedido 2022-01-03, 2017.
- [16] FLURY, B., Y RIEDWYL, H. *Multivariate Statistics: A Practical Approach*. Chapman & Hall, Ltd., London, 1988.
- [17] FRANK, M., Y WOLFE, P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly* 3 (1956), 95–110.
- [18] GETZ, W., Y WILMERS, C. A local nearest-neighbor convex-hull construction of home ranges and utilization distributions. *Ecography* 27 (08 2004), 489–505.
- [19] GORDON, G. Optimization: Convex sets. https://www.cs.cmu.edu/~ggordon/10725-F12/scribes/10725_Lecture3.pdf, Accedido 2021-11-20, 2012.
- [20] GORSKI, J., PFEUFFER, F., Y KLAMROTH, K. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research* 66, 3 (Dec 2007), 373–407.
- [21] GU, Q. Optimization: Lecture 7. [https://piazza.com/class_profile/get_resource/is58gs5cfya7ft/it3isd93pmp5ft#:~:text=For%20convex%20function%2C%20we%20can,is%20also%20a%20global%20minimum.&text=Theorem%201%20\(Local%20Minimum%20is,over%20a%20convex%20set%20D](https://piazza.com/class_profile/get_resource/is58gs5cfya7ft/it3isd93pmp5ft#:~:text=For%20convex%20function%2C%20we%20can,is%20also%20a%20global%20minimum.&text=Theorem%201%20(Local%20Minimum%20is,over%20a%20convex%20set%20D), Accedido 2022-01-03, 2016.
- [22] HEWSON, P. Swiss heads R dataset. <https://rdrr.io/cran/Flury/man/swiss.heads.html>, Accedido 2022-01-07.
- [23] HUANG, G. B., RAMESH, M., BERG, T., Y LEARNED-MILLER, E. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.
- [24] JAGGI, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning* (Atlanta, Georgia, USA, 17–19 Jun 2013), S. Dasgupta y D. McAllester, Eds., vol. 28 of *Proceedings of Machine Learning Research*, PMLR, pp. 427–435.
- [25] JPAYANSOMET. How one can draw a convex function? <https://tex.stackexchange.com/questions/394923/how-one-can-draw-a-convex-function>, Accedido 2022-01-03, 2017.

- [26] KELLER, S. M., SAMARIN, M., WIESER, M., Y ROTH, V. Deep archetypal analysis. In *Pattern Recognition* (Cham, 2019), G. A. Fink, S. Frintrop, y X. Jiang, Eds., Springer International Publishing, pp. 171–185.
- [27] LAVROV, M. Math 484 nonlinear programming: Convexity. <https://faculty.math.illinois.edu/~mlavrov/docs/484-spring-2019/ch2lec1.pdf>, Accedido 2021-11-20, 2019.
- [28] LAWSON, C. L., Y HANSON, R. J. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974.
- [29] MATHAR, R., ALIREZAEI, G., BALDA, E., Y BEHBOODI, A. *Fundamentals of Data Analytics With a View to Machine Learning*. Springer, 2020.
- [30] MØRUP, M., Y HANSEN, L. K. Archetypal analysis for machine learning and data mining. *Neurocomputing* 80 (2012), 54–63. Special Issue on Machine Learning for Signal Processing 2010.
- [31] PATEIRO-LÓPEZ, B., Y CASAL, A. Generalizing the convex hull of a sample: The r package alphahull. *Journal of Statistical Software* 34 (04 2010), 1–28.
- [32] PATEL, A. Customer personality analysis. analysis of company's ideal customers. <https://www.kaggle.com/imakash3011/customer-personality-analysis>, Accedido 2022-03-03, 2021.
- [33] PETERSEN, K. B., Y PEDERSEN, M. S. The matrix cookbook, Oct. 2008. Version 20081110.
- [34] PÉREZ, J. Geometría: Conjuntos convexos. <https://www.ugr.es/~jperez/docencia/GeomConvexos/cap1.pdf>, Accedido 2021-11-20.
- [35] ROCKYROCK. Using pfg plots to plot unit simplex in 3 dimensions. <https://tex.stackexchange.com/questions/251264/using-pfg-plots-to-plot-unit-simplex-in-3-dimensions>, Accedido 2021-11-20, 2018.
- [36] SANDERSON, C., Y LOVELL, B. C. Multi-region probabilistic histograms for robust and scalable identity inference. In *Advances in Biometrics* (Berlin, Heidelberg, 2009), M. Tistarelli y M. S. Nixon, Eds., Springer Berlin Heidelberg, pp. 199–208.
- [37] SANTILLAN, C. Improving R perfomance by installing optimized BLAS/LAPACK libraries. <https://csantill.github.io/RPerformanceWBLAS/>, Accedido 2022-02-25, 2018.
- [38] SETH, S., Y EUGSTER, M. J. Archetypal analysis for nominal observations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 5 (2016), 849–861.
- [39] SETH, S., Y EUGSTER, M. J. A. Probabilistic archetypal analysis. *Machine Learning* 102, 1 (Jan 2016), 85–113.

- [40] SIMON, B. *Convexity: An Analytic Viewpoint (Cambridge Tracts in Mathematics)*. Cambridge University Press, Cambridge, 2011.
- [41] TORBJØRN. Alignment of tikz pictures in subfigures. <https://tex.stackexchange.com/questions/302589/alignment-of-tikz-pictures-in-subfigures>, Accedido 2021-11-20, 2016.
- [42] TURK, M., Y PENTLAND, A. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* 3, 1 (01 1991), 71–86.
- [43] ÇARIKÇI, M., Y ÖZEN, F. A face recognition system based on eigenfaces method. *Procedia Technology* 1 (2012), 118–123. First World Conference on Innovation and Computer Sciences (INSODE 2011).