

vectores-componentes-principales

November 22, 2024

1 Vectores

1.1 Origen de los vectores

Los vectores surgen históricamente para representar magnitudes físicas que no pueden ser descritas completamente con un único valor numérico (escalares), ya que poseen tanto **magnitud** como **dirección**. Este concepto fue fundamental en la física para modelar fenómenos como la fuerza, la velocidad y la aceleración. Con el tiempo, los vectores adquirieron un papel central en la matemática, particularmente en la **geometría analítica** y el **álgebra lineal**, extendiéndose a aplicaciones en computación gráfica, ingeniería y mecánica cuántica.

Ejemplo: - En física, una fuerza de 10 N actuando hacia el este se representa como un vector $\mathbf{F} = (10, 0)$ en un sistema bidimensional.

1.2 Operaciones con vectores

1.2.1 Campo

Un campo es un conjunto \mathbb{F} con dos operaciones, suma (+) y multiplicación (\cdot), que satisfacen las siguientes propiedades fundamentales:

Propiedades de la suma (+): - **Asociatividad:** $(a + b) + c = a + (b + c)$ para todo $a, b, c \in \mathbb{F}$. - **Conmutatividad:** $a + b = b + a$ para todo $a, b \in \mathbb{F}$. - **Elemento neutro aditivo:** Existe un $0 \in \mathbb{F}$ tal que $a + 0 = a$. - **Inverso aditivo:** Para cada $a \in \mathbb{F}$, existe un $-a \in \mathbb{F}$ tal que $a + (-a) = 0$.

Propiedades de la multiplicación (\cdot): - **Asociatividad:** $a \cdot (b \cdot c) = (a \cdot b) \cdot c$. - **Elemento neutro multiplicativo:** Existe un $1 \in \mathbb{F}$ (con $1 \neq 0$) tal que $a \cdot 1 = a$. - **Inverso multiplicativo:** Para cada $a \in \mathbb{F}$ con $a \neq 0$, existe $a^{-1} \in \mathbb{F}$ tal que $a \cdot a^{-1} = 1$. - **Conmutatividad:** $a \cdot b = b \cdot a$.

Propiedad distributiva: - $a \cdot (b + c) = a \cdot b + a \cdot c$ para todo $a, b, c \in \mathbb{F}$.

Ejemplos: - El conjunto de números reales \mathbb{R} con las operaciones usuales de suma y multiplicación es un campo. - Los números complejos \mathbb{C} también forman un campo.

Aplicaciones: - Los campos son fundamentales en muchos aspectos de las matemáticas y las ciencias aplicadas. Se utilizan en la **teoría de números**, la **física cuántica**, y en la resolución de ecuaciones algebraicas, donde las operaciones básicas de suma y multiplicación deben cumplir propiedades estructurales específicas.

1.2.2 Escalares

Un **escalar** es cualquier elemento de un campo \mathbb{F} . En el contexto de vectores, los escalares se utilizan para cambiar la magnitud de un vector y, potencialmente, invertir su dirección.

Si $\mathbf{v} = (v_1, v_2, v_3)$ es un vector y $\lambda \in \mathbb{F}$ es un escalar, la multiplicación escalar se define como:

$$\lambda \mathbf{v} = (\lambda v_1, \lambda v_2, \lambda v_3)$$

Ejemplos: - Para $\mathbf{v} = (2, 3, -1)$ y $\lambda = 4$, se tiene:

$$\lambda \mathbf{v} = (4 \cdot 2, 4 \cdot 3, 4 \cdot -1) = (8, 12, -4)$$

Aplicaciones: - El uso de escalares es crucial en la **transformación de coordenadas** en álgebra lineal, **ajuste de escalas** en imágenes digitales, y en la **física** para ajustar la intensidad de vectores como la fuerza, el desplazamiento, etc.

1.2.3 Producto Escalar

El **producto escalar** es una operación que asocia a dos vectores un número escalar, representando su relación en términos de magnitud y dirección.

Para $\mathbf{u} = (u_1, u_2, u_3)$ y $\mathbf{v} = (v_1, v_2, v_3)$, el producto escalar se define como:

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + u_3 v_3$$

Propiedades: - **Conmutatividad:** $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$. - **Distributividad:** $\mathbf{u} \cdot (\mathbf{v} + \mathbf{w}) = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{w}$.
- **Relación con el ángulo:** Si θ es el ángulo entre \mathbf{u} y \mathbf{v} :

$$\mathbf{u} \cdot \mathbf{v} = |\mathbf{u}| |\mathbf{v}| \cos \theta$$

Ejemplo: - Dados $\mathbf{u} = (1, 2, 3)$ y $\mathbf{v} = (4, 5, 6)$, su producto escalar es:

$$\mathbf{u} \cdot \mathbf{v} = (1 \cdot 4) + (2 \cdot 5) + (3 \cdot 6) = 4 + 10 + 18 = 32$$

Aplicaciones: - En **física**, el producto escalar se usa para calcular el trabajo realizado por una fuerza sobre un objeto. - En **aprendizaje automático**, el producto escalar se utiliza para medir la similitud entre vectores de características en tareas como la clasificación y la regresión. - En **gráficos computacionales**, el producto escalar ayuda a determinar el ángulo entre vectores de superficie y la luz, lo que es crucial en el sombreado.

1.2.4 Producto Cruz

El **producto cruz** es una operación definida en \mathbb{R}^3 que asocia a dos vectores un nuevo vector perpendicular a ambos. Se calcula como:

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix}$$

Donde $\mathbf{i}, \mathbf{j}, \mathbf{k}$ son los vectores unitarios en \mathbb{R}^3 .

Propiedades: - **No conmutatividad:** $\mathbf{u} \times \mathbf{v} = -(\mathbf{v} \times \mathbf{u})$. - **Distributividad:** $\mathbf{u} \times (\mathbf{v} + \mathbf{w}) = (\mathbf{u} \times \mathbf{v}) + (\mathbf{u} \times \mathbf{w})$. - **Magnitud:** El módulo del producto cruz está relacionado con el área del paralelogramo formado por \mathbf{u} y \mathbf{v} :

$$|\mathbf{u} \times \mathbf{v}| = |\mathbf{u}| |\mathbf{v}| \sin \theta$$

Donde θ es el ángulo entre \mathbf{u} y \mathbf{v} .

Ejemplo: - Dados $\mathbf{u} = (1, 0, 0)$ y $\mathbf{v} = (0, 1, 0)$, su producto cruz es:

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix} = \mathbf{i}(0 \cdot 0 - 0 \cdot 1) - \mathbf{j}(1 \cdot 0 - 0 \cdot 0) + \mathbf{k}(1 \cdot 1 - 0 \cdot 0) = \mathbf{k}$$

Aplicaciones: - El producto cruzado es útil en **mecánica**, especialmente en el cálculo de momentos de fuerza y en la determinación de torques. - En **gráficos computacionales**, se usa para calcular normales a superficies en el modelado 3D.

1.2.5 Producto de Hadamard

El **producto de Hadamard**, también conocido como **producto elemento a elemento**, es una operación que se realiza únicamente en vectores o matrices de igual dimensión. El resultado es otro vector o matriz donde cada elemento se obtiene multiplicando las entradas correspondientes de los operandos.

Para dos vectores $\mathbf{u} = (u_1, u_2, u_3)$ y $\mathbf{v} = (v_1, v_2, v_3)$, el producto de Hadamard se define como:

$$\mathbf{u} \circ \mathbf{v} = (u_1 \cdot v_1, u_2 \cdot v_2, u_3 \cdot v_3)$$

Propiedades: - **Conmutatividad:** $\mathbf{u} \circ \mathbf{v} = \mathbf{v} \circ \mathbf{u}$. - **Asociatividad:** $(\mathbf{u} \circ \mathbf{v}) \circ \mathbf{w} = \mathbf{u} \circ (\mathbf{v} \circ \mathbf{w})$. - **Compatibilidad dimensional:** Sólo está definido si los vectores tienen la misma dimensión. - **No distributivo respecto a la suma usual:** $\mathbf{u} \circ (\mathbf{v} + \mathbf{w}) \neq (\mathbf{u} \circ \mathbf{v}) + (\mathbf{u} \circ \mathbf{w})$ en general.

Ejemplo: - Dados $\mathbf{u} = (1, 2, 3)$ y $\mathbf{v} = (4, 5, 6)$, su producto de Hadamard es:

$$\mathbf{u} \circ \mathbf{v} = (1 \cdot 4, 2 \cdot 5, 3 \cdot 6) = (4, 10, 18)$$

Aplicaciones: - En aprendizaje automático y análisis de datos, el producto de Hadamard se usa frecuentemente en operaciones de normalización y cálculo de similitudes. - En álgebra, tiene aplicaciones en la manipulación de matrices, especialmente en computación paralela.

1.3 Definición de estructuras vectoriales

1.3.1 Semigrupo

Un semigrupo es una estructura algebraica (S, \cdot) que consiste en un conjunto S junto con una operación binaria \cdot que cumple la propiedad de **asociatividad**:

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c, \quad \forall a, b, c \in S$$

Propiedades: - **Asociatividad:** La operación \cdot se aplica de la misma forma sin importar cómo se agrupan los elementos. Por ejemplo:

$$(2 + 3) + 4 = 2 + (3 + 4)$$

Ejemplo: - El conjunto de números naturales \mathbb{N} con la operación suma $(+)$ es un semigrupo: - Asociatividad: $(a + b) + c = a + (b + c)$. - Sin elemento neutro para todos los casos (por lo que no es un grupo).

Aplicaciones del semigrupo: - **Procesos de agregación:** El semigrupo es útil en el modelado de procesos donde los elementos se agregan de manera secuencial, como la acumulación de recursos o el tiempo de espera en sistemas de colas. - **Teoría de lenguajes formales:** En autómatas y gramáticas formales, los semigrupos ayudan a describir cómo se combinan las cadenas de caracteres.

1.3.2 Grupo

Un grupo es una estructura algebraica (G, \cdot) donde G es un conjunto y \cdot es una operación binaria que satisface las siguientes propiedades: - **Asociatividad:** $(a \cdot b) \cdot c = a \cdot (b \cdot c)$, $\forall a, b, c \in G$. - **Elemento neutro:** Existe un elemento $e \in G$ tal que $a \cdot e = e \cdot a = a$, $\forall a \in G$. - **Inversos:** Para cada $a \in G$, existe un elemento $a^{-1} \in G$ tal que $a \cdot a^{-1} = a^{-1} \cdot a = e$.

Propiedades adicionales (si es un grupo abeliano): - **Conmutatividad:** $a \cdot b = b \cdot a$, $\forall a, b \in G$.

Ejemplo: - El conjunto de enteros \mathbb{Z} con la operación suma $(+)$: - Asociatividad: $(a + b) + c = a + (b + c)$. - Elemento neutro: 0, ya que $a + 0 = a$. - Inverso: Para cualquier $a \in \mathbb{Z}$, el inverso aditivo es $-a$, ya que $a + (-a) = 0$. - Este es un grupo abeliano porque $a + b = b + a$.

Aplicaciones del grupo: - **Criptografía:** Los grupos son fundamentales en algoritmos de cifrado y firmas digitales, donde las operaciones de grupo aseguran la seguridad y fiabilidad de los sistemas. - **Simetría en física:** Los grupos son clave en el estudio de simetrías en física teórica, como en la teoría de partículas y la física cuántica, donde se usan para describir las simetrías de las leyes físicas.

1.3.3 Anillo

Un anillo $(R, +, \cdot)$ es una estructura algebraica con dos operaciones (suma y multiplicación) que cumple las siguientes propiedades: - **$(R, +)$ es un grupo abeliano:** - La operación $+$ es asociativa, conmutativa, tiene un elemento neutro 0 y cada elemento $a \in R$ tiene un inverso aditivo $-a$. - **(R, \cdot) es un semigrupo:** - La operación \cdot es asociativa. - **Distributividad:** La multiplicación es distributiva con respecto a la suma:

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(a + b) \cdot c = (a \cdot c) + (b \cdot c)$$

Ejemplo: - El conjunto de números enteros \mathbb{Z} con las operaciones suma $(+)$ y multiplicación (\cdot) : - $(\mathbb{Z}, +)$ es un grupo abeliano. - (\mathbb{Z}, \cdot) es un semigrupo. - La multiplicación es distributiva sobre la suma. - Nota: \mathbb{Z} no es un campo porque no todos los elementos tienen inverso multiplicativo.

Aplicaciones del anillo: - **Teoría de códigos:** Los anillos son útiles en el desarrollo de códigos correctores de errores, donde las operaciones en anillos permiten una corrección eficiente de errores en la transmisión de datos. - **Geometría algebraica:** En geometría algebraica, los anillos de polinomios son herramientas fundamentales para estudiar variedades algebraicas.

Ejemplo de anillo conmutativo: - El conjunto de números polinomiales $P[x]$ con coeficientes en \mathbb{R} y las operaciones usuales de suma y multiplicación.

1.3.4 Espacio Vectorial

Un espacio vectorial $(V, \mathbb{F}, +, \cdot)$ es una estructura algebraica que consta de: - Un conjunto V cuyos elementos se llaman **vectores**. - Un campo \mathbb{F} cuyos elementos se llaman **escalares**. - Una operación de suma vectorial $(+)$ definida en V . - Una operación de multiplicación escalar (\cdot) entre elementos de \mathbb{F} y V .

Estas operaciones satisfacen las siguientes propiedades:

Propiedades de la suma vectorial $(+)$: - **Asociatividad:** $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$, $\forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in V$. - **Conmutatividad:** $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$, $\forall \mathbf{u}, \mathbf{v} \in V$. - **Elemento neutro:** Existe $\mathbf{0} \in V$ tal que $\mathbf{u} + \mathbf{0} = \mathbf{u}$. - **Inverso aditivo:** Para cada $\mathbf{u} \in V$, existe $-\mathbf{u} \in V$ tal que $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$.

Propiedades de la multiplicación escalar (\cdot) : - **Asociatividad con escalares:** $\alpha(\beta\mathbf{v}) = (\alpha\beta)\mathbf{v}$, $\forall \alpha, \beta \in \mathbb{F}, \mathbf{v} \in V$. - **Elemento neutro multiplicativo:** $1 \cdot \mathbf{v} = \mathbf{v}$, $\forall \mathbf{v} \in V$, donde 1 es el elemento neutro del campo \mathbb{F} . - **Distributividad con escalares y vectores:** - $\alpha(\mathbf{u} + \mathbf{v}) = \alpha\mathbf{u} + \alpha\mathbf{v}$, $\forall \alpha \in \mathbb{F}, \mathbf{u}, \mathbf{v} \in V$. - $(\alpha + \beta)\mathbf{v} = \alpha\mathbf{v} + \beta\mathbf{v}$, $\forall \alpha, \beta \in \mathbb{F}, \mathbf{v} \in V$.

Ejemplo: - El conjunto de vectores en \mathbb{R}^n , con las operaciones de suma vectorial y multiplicación escalar usuales: - **Suma:** $\mathbf{u} + \mathbf{v} = (u_1 + v_1, u_2 + v_2, \dots, u_n + v_n)$. - **Multiplicación escalar:** $\alpha\mathbf{v} = (\alpha v_1, \alpha v_2, \dots, \alpha v_n)$. - Este espacio cumple todas las propiedades mencionadas y tiene aplicaciones en geometría, física y computación.

Aplicaciones del espacio vectorial: - **Grafos y redes:** Los espacios vectoriales son útiles en el análisis y representación de redes, como en la optimización de flujos en redes de transporte. - **Gráficas computacionales:** En gráficos por computadora, los espacios vectoriales modelan transformaciones geométricas como rotaciones, escalados y traslaciones.

Ejemplo aplicado: - El conjunto de funciones continuas $C[a, b]$ en un intervalo $[a, b]$ con las operaciones: - Suma: $(f + g)(x) = f(x) + g(x)$. - Multiplicación escalar: $(\alpha f)(x) = \alpha f(x)$. - Este es un espacio vectorial sobre el campo \mathbb{R} .

1.3.5 Subespacio Vectorial

Un **subespacio vectorial** es un subconjunto W de un espacio vectorial V que cumple las siguientes tres condiciones: 1. **Cerradura bajo la suma vectorial:** Si $\mathbf{u}, \mathbf{v} \in W$, entonces $\mathbf{u} + \mathbf{v} \in W$. 2. **Cerradura bajo la multiplicación escalar:** Si $\mathbf{u} \in W$ y $\alpha \in \mathbb{F}$, entonces $\alpha\mathbf{u} \in W$. 3. **Contiene el vector nulo:** $\mathbf{0} \in W$.

Ejemplo: - El conjunto de vectores en \mathbb{R}^3 de la forma $(x, y, 0)$, donde $x, y \in \mathbb{R}$, es un subespacio de \mathbb{R}^3 . - Es cerrado bajo la suma: Si $(x_1, y_1, 0)$ y $(x_2, y_2, 0)$ son vectores de W , entonces $(x_1 + x_2, y_1 + y_2, 0)$ también es un vector de W . - Es cerrado bajo la multiplicación escalar: Si $(x, y, 0) \in W$ y $\alpha \in \mathbb{R}$, entonces $(\alpha x, \alpha y, 0) \in W$. - Contiene el vector nulo: El vector $(0, 0, 0)$ está en W .

Aplicaciones del subespacio vectorial: - **Sistemas de ecuaciones lineales:** Los subespacios vectoriales son esenciales en la resolución de sistemas de ecuaciones lineales, donde se buscan soluciones dentro de subespacios específicos. - **Control y robótica:** En el análisis y diseño de sistemas de control y en la robótica, los subespacios vectoriales se usan para modelar trayectorias y movimientos en el espacio tridimensional.

1.4 Vectores, valores propios, y subespacios generados

La historia de los vectores, los valores propios y los subespacios generados tiene sus raíces en el desarrollo del álgebra lineal, una rama fundamental de las matemáticas que estudia los espacios vectoriales y las transformaciones lineales. Estas herramientas matemáticas, desarrolladas inicialmente por figuras clave como Carl Friedrich Gauss, Arthur Cayley y David Hilbert, permitieron profundizar en el entendimiento de las estructuras algebraicas y sus aplicaciones. A lo largo de los siglos, estos conceptos se extendieron desde la geometría y la física hasta otras disciplinas como la economía, la computación y la ingeniería. En la actualidad, el álgebra lineal es uno de los pilares fundamentales de la ciencia de datos y el aprendizaje automático, jugando un papel esencial en la manipulación y análisis de grandes volúmenes de datos.

Los **vectores propios** y **valores propios** son conceptos claves en álgebra lineal que surgen cuando se estudian transformaciones lineales en espacios vectoriales. Estos conceptos no solo son fundamentales para la teoría matemática, sino que también tienen aplicaciones directas en áreas como la computación, la física, la estadística y, especialmente, en la ciencia de datos.

1.5 Definición matemática de vectores propios y valores propios

Dado un operador lineal representado por una matriz cuadrada A , un vector no nulo \mathbf{v} se denomina **vector propio** de A si existe un valor escalar λ tal que:

$$A\mathbf{v} = \lambda\mathbf{v}$$

Donde λ es el **valor propio** asociado al vector propio \mathbf{v} . En otras palabras, la acción de la matriz sobre el vector solo cambia su magnitud (por un factor λ) y no su dirección. La importancia de estos vectores radica en que describen las direcciones invariantes de una transformación, es decir, las direcciones en las que los datos no se “doblan” ni se “desplazan”, solo se escalan.

1.5.1 Cálculo de los valores propios y vectores propios

Para encontrar los valores propios y los vectores propios de una matriz A , resolvemos la ecuación característicamente conocida como:

$$\det(A - \lambda I) = 0$$

Donde I es la matriz identidad de la misma dimensión que A , y λ es el valor propio que estamos buscando. Una vez encontrados los valores propios, los vectores propios asociados se encuentran sustituyendo cada valor propio λ en la ecuación:

$$(A - \lambda I)\mathbf{v} = 0$$

Esta es un sistema de ecuaciones lineales homogéneas que se resuelve para encontrar los vectores propios.

1.6 Origen histórico de los vectores y valores propios

El concepto de **valor propio** fue introducido por el matemático alemán **David Hilbert** en el contexto de la teoría de matrices y ecuaciones diferenciales. En sus trabajos sobre operadores lineales y ecuaciones diferenciales, Hilbert y otros pioneros comenzaron a estudiar las soluciones de sistemas lineales de ecuaciones en los que las matrices actuaban sobre vectores, pero en lugar de alterar su dirección, simplemente los escalaban. Este descubrimiento fue crucial para entender el comportamiento de sistemas físicos complejos, como los sistemas cuánticos y la mecánica estadística.

Sin embargo, la verdadera importancia de los valores propios y los vectores propios fue completamente reconocida a medida que avanzaba la teoría de matrices. A lo largo del siglo XX, el desarrollo de computadoras y algoritmos eficientes permitió aplicar estos conceptos en una variedad de áreas, desde la ingeniería estructural hasta la informática. Con el aumento de la digitalización de datos, las técnicas matemáticas basadas en álgebra lineal, como el **Análisis de Componentes Principales (PCA)** y la **Descomposición en Valores Singulares (SVD)**, adquirieron una relevancia aún mayor, facilitando el procesamiento y análisis de grandes volúmenes de información.

1.7 Aplicaciones en Ciencia de Datos

Los conceptos de **vectores propios** y **valores propios** tienen múltiples aplicaciones en ciencia de datos. Estos conceptos no solo son fundamentales para simplificar y organizar grandes volúmenes de datos, sino que también permiten mejorar la eficiencia de los algoritmos, obtener insights valiosos y reducir el tiempo de procesamiento. En la ciencia de datos moderna, el análisis y manipulación de datos multidimensionales es una tarea común, y los vectores propios y valores propios desempeñan un papel crucial en la reducción de la complejidad de estos datos.

1.7.1 Reducción de Dimensionalidad: Análisis de Componentes Principales (PCA)

La **reducción de dimensionalidad** es una técnica que busca reducir la cantidad de variables (o dimensiones) de un conjunto de datos manteniendo la mayor parte de la información importante. Uno de los métodos más populares para la reducción de dimensionalidad es el **Análisis de Componentes Principales (PCA)**, que utiliza valores propios y vectores propios para proyectar los datos en un espacio de menor dimensión.

PCA comienza calculando la matriz de covarianza de los datos, que describe cómo las variables del conjunto de datos varían entre sí. Los valores propios y los vectores propios de esta matriz de covarianza indican las direcciones en las que los datos tienen la mayor varianza. Al ordenar los vectores propios según los valores propios, podemos seleccionar un subconjunto de las direcciones más importantes y proyectar los datos sobre ellas. Este proceso elimina la redundancia y reduce la dimensionalidad de manera efectiva.

Matemáticamente, si X es un conjunto de datos de n dimensiones, la matriz de covarianza C se calcula como:

$$C = \frac{1}{n} X^T X$$

Luego, los valores propios y vectores propios de C nos dan las direcciones (componentes principales) y las magnitudes de la varianza en esas direcciones.

Ejemplo de PCA en imágenes En la compresión de imágenes, PCA puede ser usado para reducir el tamaño de la imagen sin perder demasiada información. Dado un conjunto de píxeles representado como un conjunto de datos, PCA encuentra las direcciones de mayor varianza en el espacio de los píxeles y utiliza estas direcciones para representar la imagen de manera más eficiente. Este proceso reduce la cantidad de información necesaria para representar la imagen, sin perder los detalles más importantes.

1.7.2 Descomposición en Valores Singulares (SVD)

La **Descomposición en Valores Singulares (SVD)** es otra técnica que está estrechamente relacionada con los vectores propios y los valores propios. SVD descompone cualquier matriz A en tres matrices:

$$A = U\Sigma V^T$$

Donde U y V son matrices cuyas columnas son los vectores propios de AA^T y $A^T A$, respectivamente, y Σ es una matriz diagonal que contiene los valores singulares de A . Esta descomposición es útil para una variedad de aplicaciones, incluidas la compresión de datos, la resolución de sistemas de ecuaciones lineales, y la factorización de matrices en modelos de recomendación.

SVD se utiliza ampliamente en sistemas de recomendación como Netflix y Amazon para descomponer matrices de interacciones de usuarios y productos en factores latentes que pueden usarse para predecir qué productos un usuario podría preferir.

1.7.3 Aplicaciones de SVD en aprendizaje automático

En aprendizaje automático, SVD es útil para mejorar la eficiencia de algoritmos de clasificación y regresión. Al descomponer las matrices de características en componentes más simples, SVD puede ayudar a eliminar la redundancia, reducir el ruido y mejorar la calidad de los modelos de machine learning. En el caso de la **reducción de ruido**, los valores singulares más pequeños pueden ser descartados, lo que permite una representación más precisa de los datos.

1.7.4 Análisis de Redes y Grafos

Los valores propios y los vectores propios también tienen aplicaciones importantes en el análisis de redes y grafos. Un **grafo** es una estructura que consta de nodos (o vértices) y aristas (conexiones entre nodos). Los valores propios de la **matriz de adyacencia** de un grafo proporcionan información sobre las propiedades estructurales del grafo. Por ejemplo, los **vectores propios** asociados a los valores propios más grandes pueden ayudar a identificar nodos importantes dentro de la red, como aquellos que son más influyentes o centrales.

En las redes sociales, el análisis espectral de grafos se utiliza para estudiar cómo se propaga la información y cómo identificar comunidades dentro de grandes redes. Esta técnica puede ser aplicada en marketing viral, propagación de enfermedades, e incluso en la optimización de flujos de tráfico.

1.7.5 Subespacios Generados y sus Aplicaciones

Los **subespacios generados** por los vectores propios juegan un papel fundamental en la reducción de la dimensionalidad y la extracción de características importantes de los datos. Un subespacio

generado es el conjunto de todos los vectores que pueden formarse mediante combinaciones lineales de un conjunto de vectores propios.

Por ejemplo, si tenemos un conjunto de datos de n dimensiones, los vectores propios nos permiten generar un subespacio de menor dimensión que contenga la mayor parte de la varianza de los datos. Este subespacio facilita la visualización de datos en dimensiones más bajas y mejora la interpretación de los resultados de los algoritmos de aprendizaje automático.

En resumen, los conceptos de **vectores propios**, **valores propios** y **subespacios generados** son esenciales para el análisis y la manipulación de grandes volúmenes de datos en la ciencia de datos. Estos conceptos proporcionan herramientas poderosas para la **reducción de dimensionalidad**, la **compresión de datos**, y la **extracción de características**, mejorando la eficiencia de los algoritmos y facilitando la toma de decisiones informadas en una variedad de aplicaciones. A medida que los conjuntos de datos continúan creciendo en tamaño y complejidad, el uso de estos conceptos será cada vez más crucial en la optimización de modelos, la mejora de la visualización de datos, y la toma de decisiones informadas en una amplia variedad de aplicaciones.

1.8 Análisis de Vectores Propios, Valores Propios y Subespacios Generados en el Conjunto de Datos Iris

En este tutorial, exploraremos cómo los **vectores propios** y **valores propios** están profundamente conectados con el análisis de componentes principales (PCA) y la descomposición en valores singulares (SVD). Usaremos el conjunto de datos Iris de **scikit-learn** para aplicar estas técnicas, visualizar los resultados y entender cómo las matrices de covarianza y sus componentes contribuyen a la reducción de dimensionalidad.

1.8.1 Importación de librerías y carga del conjunto de datos Iris

Primero, importamos las librerías necesarias y cargamos el conjunto de datos Iris de **scikit-learn**. Este conjunto de datos es muy popular en la ciencia de datos, ya que contiene información sobre 150 muestras de flores de iris, con 4 características medidas: longitud y ancho de los sépalos y pétalos.

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import TruncatedSVD

# Cargar el conjunto de datos Iris
iris = load_iris()
X = iris.data # Características
y = iris.target # Etiquetas
feature_names = iris.feature_names
target_names = iris.target_names

# Mostrar las primeras filas del conjunto de datos
```

```
df = pd.DataFrame(X, columns=feature_names)
df['species'] = pd.Categorical.from_codes(y, target_names)
df.head()
```

```
[1]:      sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm) \
0                5.1           3.5           1.4           0.2
1                4.9           3.0           1.4           0.2
2                4.7           3.2           1.3           0.2
3                4.6           3.1           1.5           0.2
4                5.0           3.6           1.4           0.2

      species
0  setosa
1  setosa
2  setosa
3  setosa
4  setosa
```

1.8.2 Estandarización de los datos

Antes de aplicar PCA o cualquier técnica de reducción de dimensionalidad, es recomendable **estandarizar** los datos para que todas las características tengan la misma escala. Esto se hace utilizando **StandardScaler** de scikit-learn. La estandarización transforma los datos para que tengan una media de 0 y una desviación estándar de 1.

```
[2]: # Estandarizar los datos
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Mostrar las primeras filas de los datos estandarizados
pd.DataFrame(X_scaled, columns=feature_names).head()
```

```
[2]:      sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0          -0.900681           1.019004          -1.340227          -1.315444
1          -1.143017          -0.131979          -1.340227          -1.315444
2          -1.385353           0.328414          -1.397064          -1.315444
3          -1.506521           0.098217          -1.283389          -1.315444
4          -1.021849           1.249201          -1.340227          -1.315444
```

1.8.3 Cálculo de la matriz de covarianza y sus valores propios

La **matriz de covarianza** de un conjunto de datos es una matriz cuadrada que describe la varianza y covarianza entre las características. Esta matriz es fundamental para la comprensión de los vectores propios y valores propios. Los **valores propios** indican cuánta varianza en los datos se encuentra en la dirección de los **vectores propios**, que representan las direcciones principales en el espacio de características. En PCA, los vectores propios están alineados con los ejes principales de máxima varianza en los datos.

```
[3]: # Calcular la matriz de covarianza
cov_matrix = np.cov(X_scaled.T)

# Calcular los valores propios y los vectores propios
eig_values, eig_vectors = np.linalg.eig(cov_matrix)

# Mostrar los valores propios y los vectores propios
print("Valores propios:", eig_values)
print("\nVectores propios:\n", eig_vectors)
```

Valores propios: [2.93808505 0.9201649 0.14774182 0.02085386]

Vectores propios:

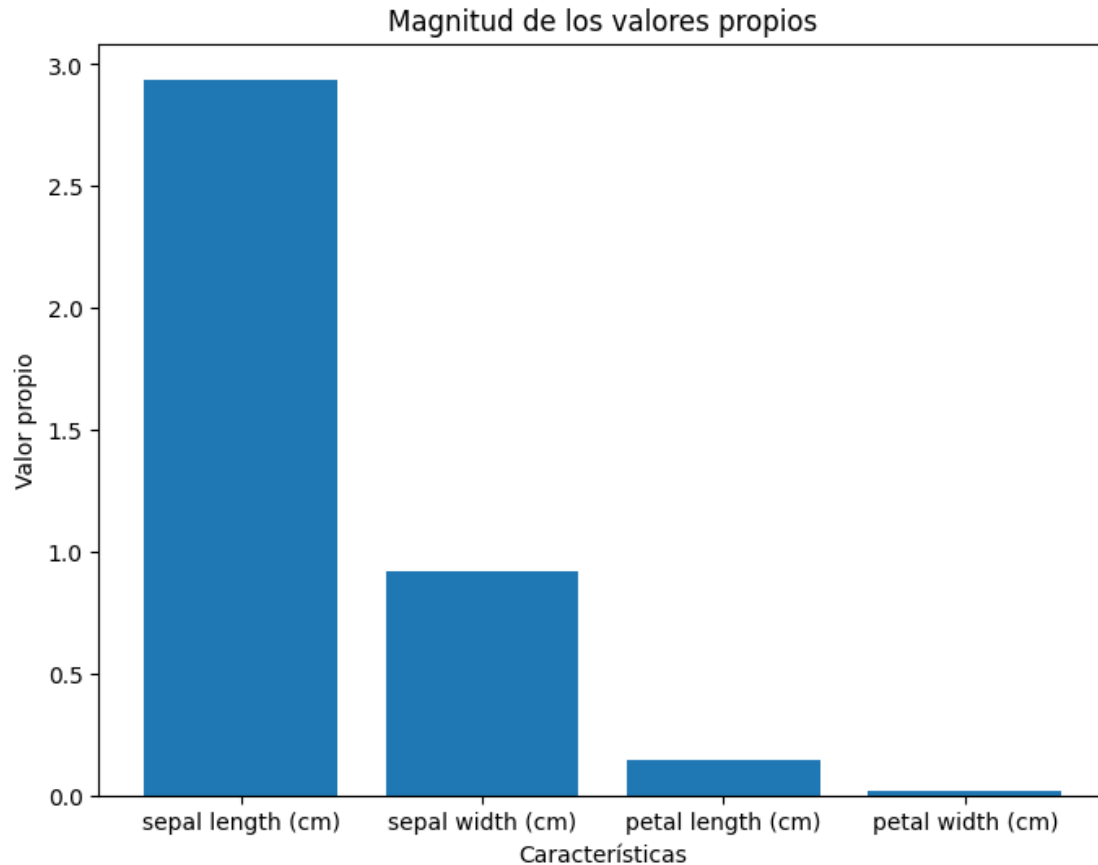
```
[[ 0.52106591 -0.37741762 -0.71956635 0.26128628]
 [-0.26934744 -0.92329566 0.24438178 -0.12350962]
 [ 0.5804131 -0.02449161 0.14212637 -0.80144925]
 [ 0.56485654 -0.06694199 0.63427274 0.52359713]]
```

En este punto, tenemos los **valores propios** que indican cuánta varianza está asociada con cada componente (eje principal) del conjunto de datos. Los **vectores propios** son los ejes que definen las direcciones principales de variabilidad en los datos.

1.8.4 Visualización de los valores propios y su magnitud

Para comprender mejor el impacto de cada valor propio, podemos graficar los valores propios y observar cómo se distribuye la varianza entre las características.

```
[4]: # Visualización de los valores propios (magnitud de varianza)
plt.figure(figsize=(8,6))
plt.bar(range(len(eig_values)), eig_values, tick_label=feature_names)
plt.xlabel('Características')
plt.ylabel('Valor propio')
plt.title('Magnitud de los valores propios')
plt.show()
```



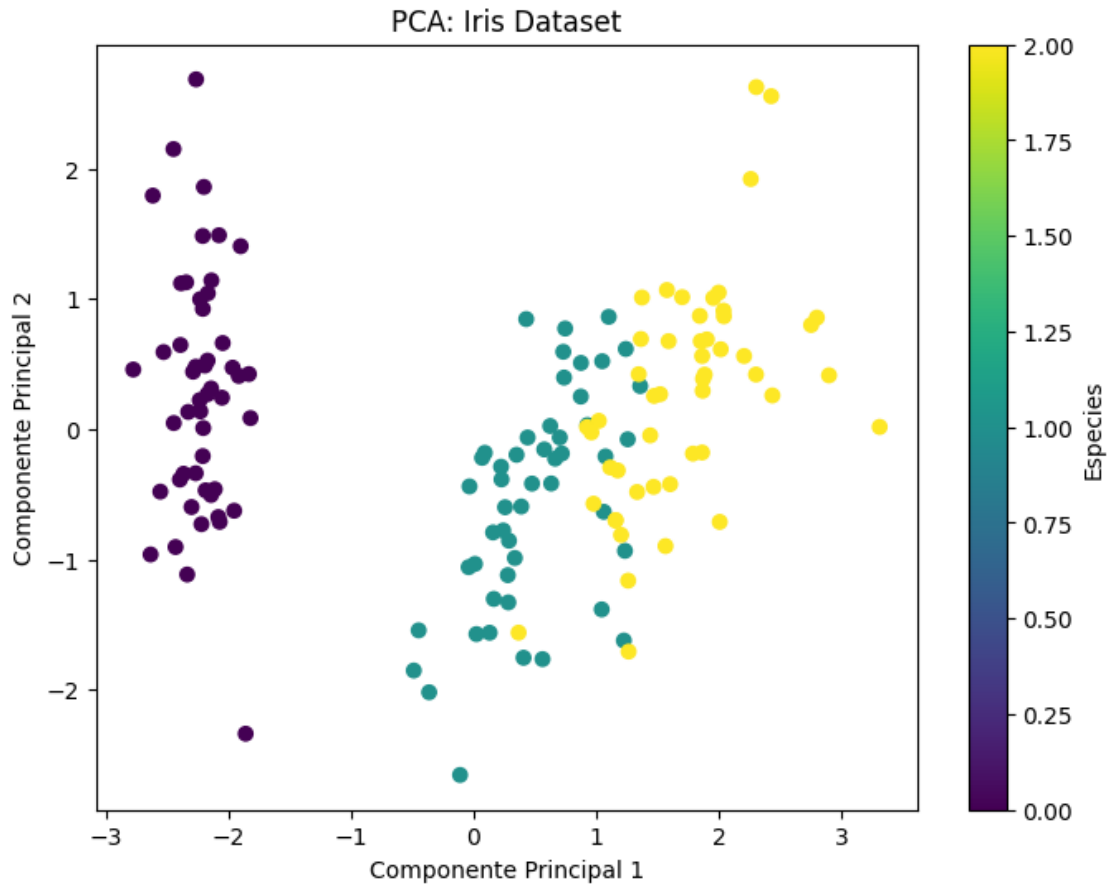
1.8.5 Análisis de Componentes Principales (PCA)

Ahora aplicamos el **Análisis de Componentes Principales (PCA)**, que utiliza los vectores propios de la matriz de covarianza para reducir la dimensionalidad de los datos. Vamos a proyectar los datos en las dos primeras componentes principales y visualizarlos en un gráfico.

```
[5]: # Aplicar PCA
n_components = len(eig_values)
pca = PCA(n_components)
X_pca = pca.fit_transform(X_scaled)

# Visualizar los resultados de PCA
plt.figure(figsize=(8,6))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='viridis')
plt.xlabel('Componente Principal 1')
plt.ylabel('Componente Principal 2')
plt.title('PCA: Iris Dataset')
plt.colorbar(label='Especies')
plt.show()
```

```
# Mostrar la varianza explicada por cada componente
print(f"Varianza explicada por los primeros dos componentes: {sum(pca.
↪ explained_variance_ratio_[0:2])}")
```



Varianza explicada por los primeros dos componentes: 0.9581320720000166

La **varianza explicada** por los primeros componentes principales nos indica cuánta de la variabilidad original de los datos se conserva al reducir las dimensiones. Esta es una de las razones por las que PCA es útil: permite representar los datos de manera más simple sin perder demasiada información.

1.8.6 Graficar los vectores propios en el espacio de características

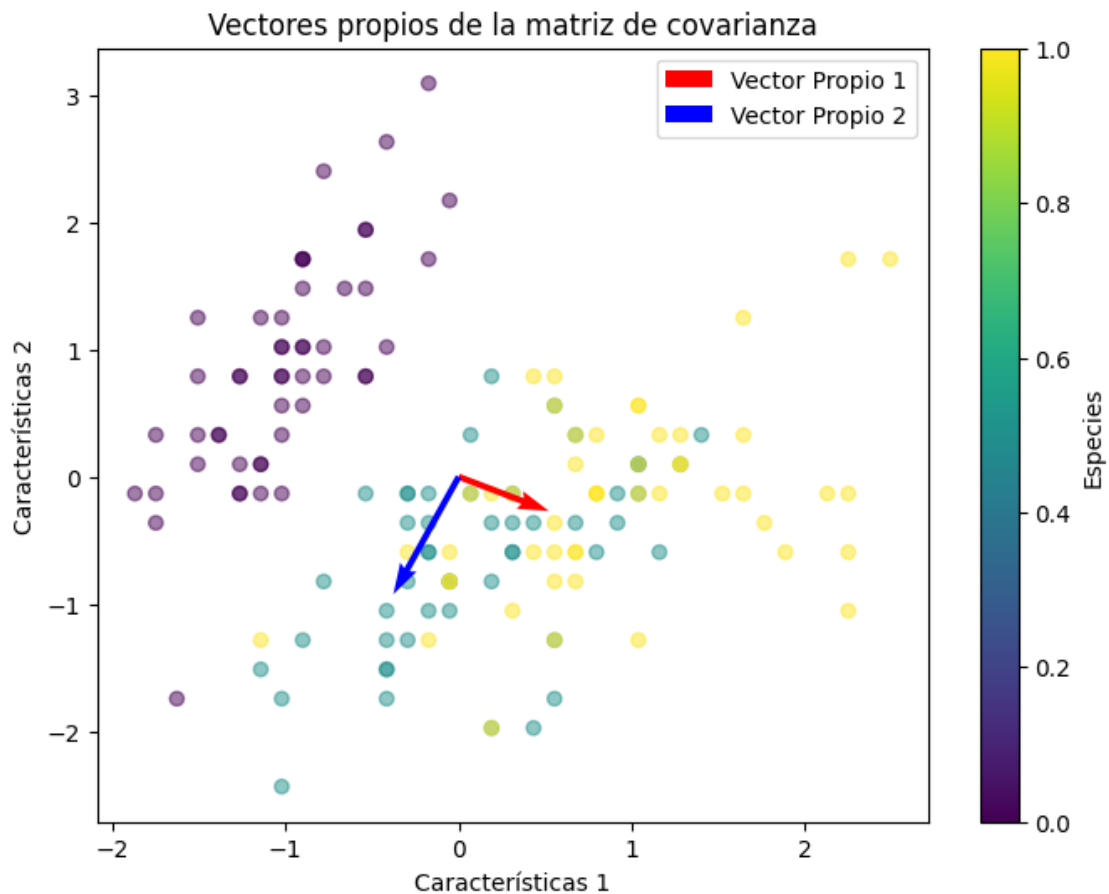
Los vectores propios son las direcciones principales a lo largo de las cuales los datos varían. Vamos a graficar estos vectores en el espacio de características para entender cómo se proyectan los datos a lo largo de estas direcciones principales. En esta gráfica, solo mostraremos los dos primeros vectores propios, que son los que se han utilizado para el PCA.

```
[6]: # Graficar los vectores propios en el espacio de características
plt.figure(figsize=(8,6))
```

```
plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=y, cmap='viridis', alpha=0.5)

# Graficar solo los primeros dos vectores propios de la matriz de covarianza
plt.quiver(0, 0, eig_vectors[0, 0], eig_vectors[1, 0], angles='xy',
          ↪scale_units='xy', scale=1, color='r', label='Vector Propio 1')
plt.quiver(0, 0, eig_vectors[0, 1], eig_vectors[1, 1], angles='xy',
          ↪scale_units='xy', scale=1, color='b', label='Vector Propio 2')

plt.xlabel('Características 1')
plt.ylabel('Características 2')
plt.title('Vectores propios de la matriz de covarianza')
plt.colorbar(label='Especies')
plt.legend()
plt.show()
```

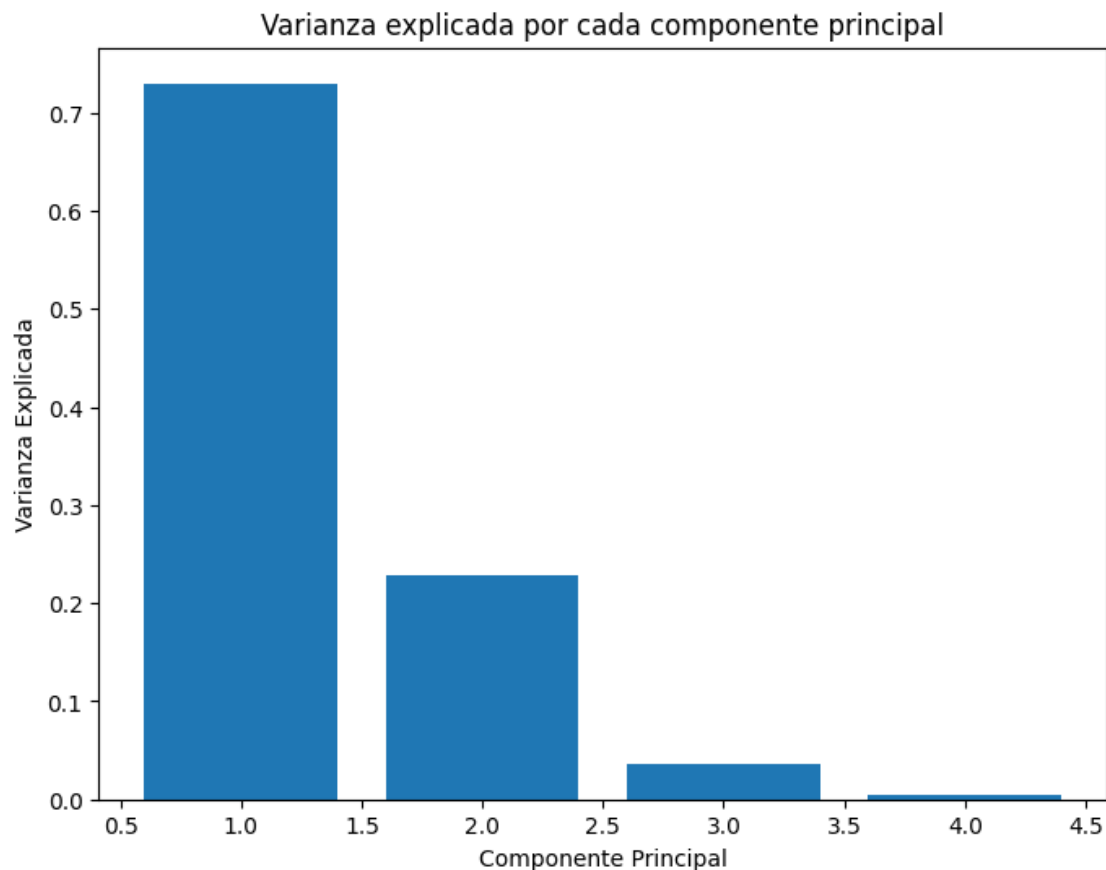


Los vectores rojos y azules en el gráfico indican las direcciones principales de máxima varianza (vectores propios). Al proyectar los datos en estas direcciones, obtenemos una representación más compacta de los mismos.

1.8.7 Visualización de la magnitud de la varianza de los componentes principales

A continuación, vamos a graficar la **magnitud de la varianza explicada** por cada componente principal, que nos permite entender cuánta información de los datos es capturada por cada componente.

```
[7]: # Visualización de la varianza explicada por cada componente principal
plt.figure(figsize=(8,6))
plt.bar(range(1, len(pca.explained_variance_ratio_) + 1), pca.
        explained_variance_ratio_)
plt.xlabel('Componente Principal')
plt.ylabel('Varianza Explicada')
plt.title('Varianza explicada por cada componente principal')
plt.show()
```



En este gráfico de barras, podemos observar cuánta varianza es explicada por cada uno de los primeros componentes principales. La altura de cada barra representa la proporción de varianza que el componente correspondiente captura de los datos.

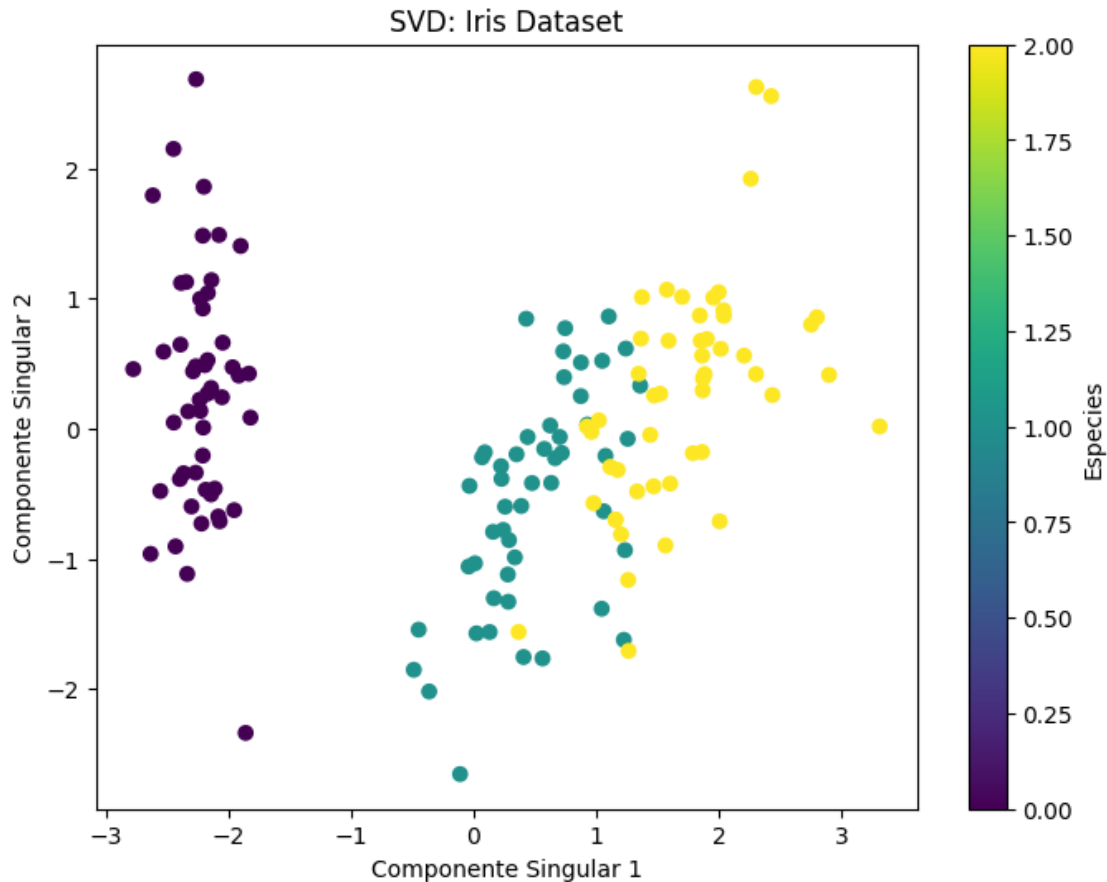
1.8.8 Descomposición en Valores Singulares (SVD)

La descomposición en valores singulares (SVD) es una técnica relacionada que descompone una matriz en tres matrices: una de vectores singulares izquierdos, una diagonal con los valores singulares y una de vectores singulares derechos. A continuación, aplicamos SVD a los datos estandarizados.

```
[8]: # Aplicar SVD
n_components = len(eig_values)
svd = TruncatedSVD(n_components)
X_svd = svd.fit_transform(X_scaled)

# Visualizar los resultados de SVD
plt.figure(figsize=(8,6))
plt.scatter(X_svd[:, 0], X_svd[:, 1], c=y, cmap='viridis')
plt.xlabel('Componente Singular 1')
plt.ylabel('Componente Singular 2')
plt.title('SVD: Iris Dataset')
plt.colorbar(label='Especies')
plt.show()

# Mostrar la varianza explicada por cada componente en SVD
print(f"Varianza explicada por los primeros dos componentes en SVD: {sum(svd.
    ↪ explained_variance_ratio_[0:2])}")
```

Varianza explicada por los primeros dos componentes en SVD: 0.9581320720000166

Al igual que PCA, SVD permite reducir la dimensionalidad de los datos, pero también tiene la ventaja de ser más generalizable y menos susceptible a ciertas irregularidades en los datos.

1.8.9 Graficar los vectores singulares en el espacio de características

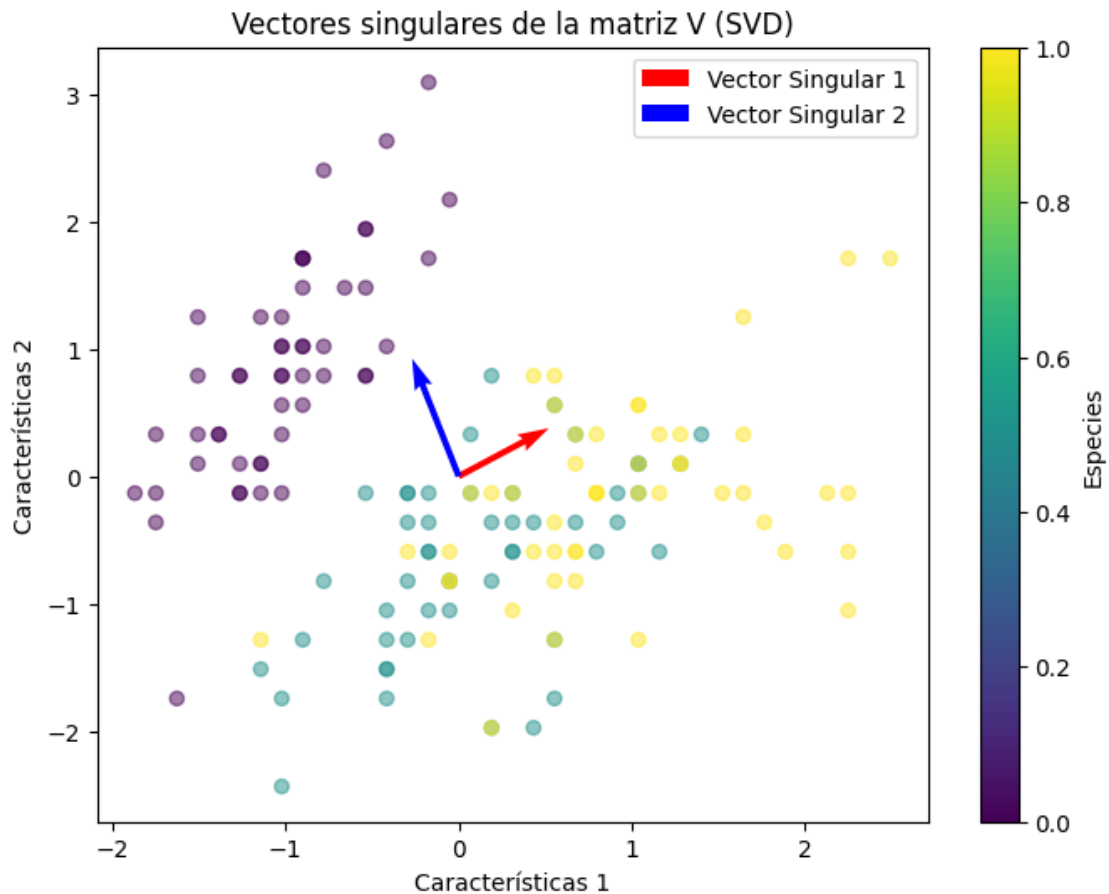
Los vectores singulares en la descomposición SVD son análogos a los vectores propios en PCA, pero en lugar de estar asociados con la matriz de covarianza, están relacionados con la matriz de datos. Vamos a graficar los vectores singulares en el espacio de características para visualizar cómo afectan la proyección de los datos. En esta gráfica, solo mostraremos los vectores singulares utilizados para la proyección de los primeros dos componentes singulares.

```
[9]: # Graficar los vectores singulares en el espacio de características
plt.figure(figsize=(8,6))
plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=y, cmap='viridis', alpha=0.5)

# Graficar solo los primeros dos vectores singulares de la matriz V (vectores_
↳ singulares derechos)
```

```
plt.quiver(0, 0, svd.components_[0, 0], svd.components_[1, 0], angles='xy',
↪scale_units='xy', scale=1, color='r', label='Vector Singular 1')
plt.quiver(0, 0, svd.components_[0, 1], svd.components_[1, 1], angles='xy',
↪scale_units='xy', scale=1, color='b', label='Vector Singular 2')

plt.xlabel('Características 1')
plt.ylabel('Características 2')
plt.title('Vectores singulares de la matriz V (SVD)')
plt.colorbar(label='Especies')
plt.legend()
plt.show()
```

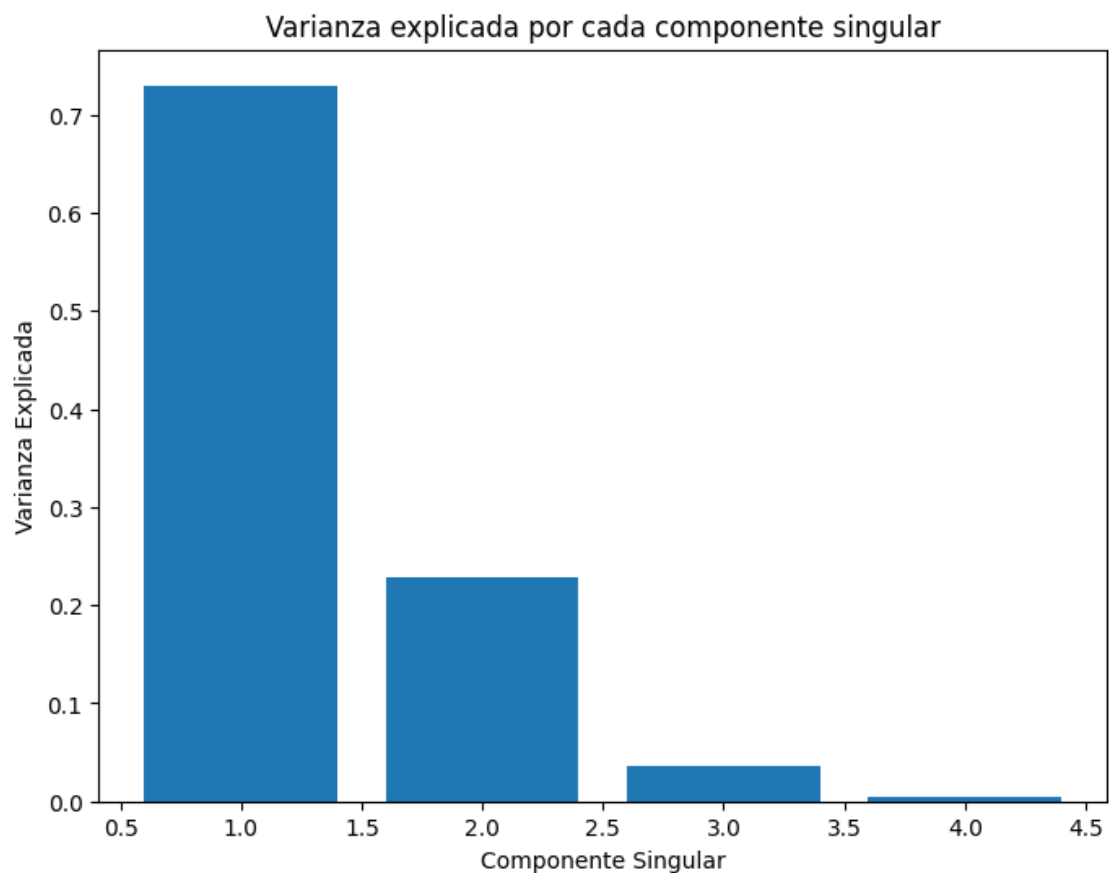


Los vectores rojos y azules en el gráfico indican las direcciones principales de máxima varianza, que son los vectores singulares. Estos vectores afectan la proyección de los datos hacia un espacio de menor dimensión.

1.8.10 Visualización de la varianza explicada por cada componente singular

Ahora, vamos a graficar la **magnitud de la varianza explicada** por cada componente singular, lo cual nos da una visión clara de qué tan bien cada componente captura la variabilidad de los datos.

```
[10]: # Visualización de la varianza explicada por cada componente singular
plt.figure(figsize=(8,6))
plt.bar(range(1, len(svd.explained_variance_ratio_) + 1), svd.
        ↪explained_variance_ratio_)
plt.xlabel('Componente Singular')
plt.ylabel('Varianza Explicada')
plt.title('Varianza explicada por cada componente singular')
plt.show()
```



En este gráfico de barras, podemos observar cuánta varianza es explicada por cada uno de los componentes singulares. La altura de cada barra muestra la proporción de varianza que el componente singular correspondiente captura de los datos.

1.8.11 Comparación de PCA y SVD

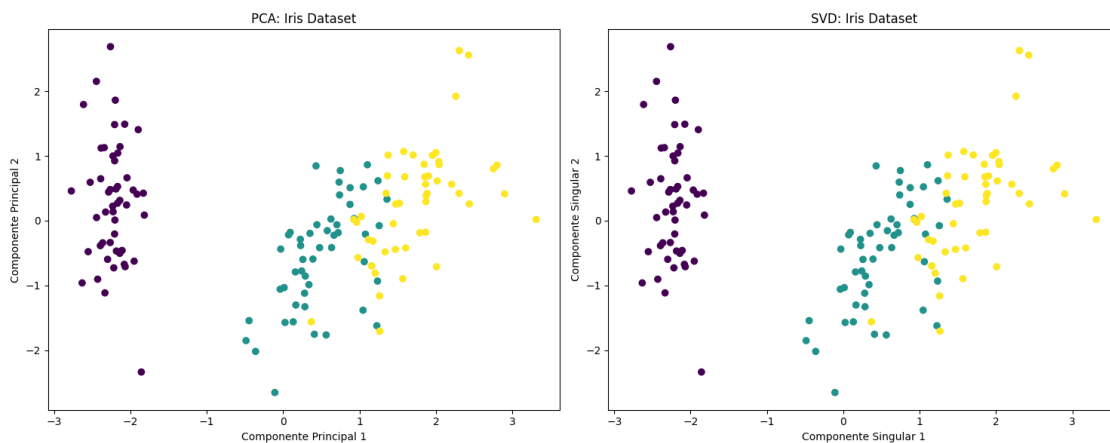
Finalmente, comparamos las representaciones de los datos obtenidas con PCA y SVD en términos de reducción de dimensionalidad. Ambas técnicas deberían proporcionar representaciones similares de los datos, pero cada una tiene sus características y aplicaciones.

```
[11]: # Comparar las proyecciones de PCA y SVD en un gráfico
fig, axes = plt.subplots(1, 2, figsize=(15,6))

# PCA
axes[0].scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='viridis')
axes[0].set_title('PCA: Iris Dataset')
axes[0].set_xlabel('Componente Principal 1')
axes[0].set_ylabel('Componente Principal 2')

# SVD
axes[1].scatter(X_svd[:, 0], X_svd[:, 1], c=y, cmap='viridis')
axes[1].set_title('SVD: Iris Dataset')
axes[1].set_xlabel('Componente Singular 1')
axes[1].set_ylabel('Componente Singular 2')

plt.tight_layout()
plt.show()
```



A lo largo de este análisis, hemos aplicado las técnicas de **Análisis de Componentes Principales (PCA)** y **Descomposición en Valores Singulares (SVD)** al conjunto de datos Iris, utilizando **valores propios** y **vectores propios** para reducir la dimensionalidad de los datos y obtener representaciones más simples. Ambas técnicas permiten entender las relaciones subyacentes entre las características del conjunto de datos y visualizarlas en un espacio de menor dimensión.

Además, la relación entre los vectores propios y la matriz de covarianza nos proporciona una intuición clara sobre cómo los datos se distribuyen en el espacio de características y cómo podemos proyectarlos de manera que conserven la mayor parte de la información relevante.