Análisis de componentes principales y sus aplicaciones en el campo del análisis de malware

Carmen Johana Calderón Chona

June 03, 2025

Contents

1. Introducción

En la era digital actual, los ciberataques representan una de las amenazas más significativas para la seguridad de los sistemas informáticos. Es así como la detección temprana de malware —software malicioso diseñado para dañar o comprometer dispositivos— se ha convertido en una tarea crítica, tanto para empresas como para usuarios individuales. En este contexto, las técnicas de aprendizaje automático han demostrado ser herramientas prometedoras para anticipar ataques basándose en grandes volúmenes de datos sobre el comportamiento y las características del sistema.

Sin embargo, estos conjuntos de datos suelen tener una alta dimensionalidad, lo que puede introducir ruido, dificultar el entrenamiento de modelos eficientes y aumentar el riesgo de sobreajuste. Una solución a este problema es el uso del Análisis de Componentes Principales (PCA), una técnica estadística que permite transformar el espacio de variables originales en un conjunto reducido de componentes no correlacionados, preservando la mayor parte de la varianza de los datos.

Este trabajo se enfoca en responder la siguiente pregunta de investigación: ¿Es posible construir un modelo de aprendizaje automático que, utilizando PCA como paso de preprocesamiento, logre predecir correctamente si un dispositivo será infectado por malware en la mayoría de los casos? Para ello, se utiliza el conjunto de datos de la competencia "Microsoft Malware Prediction" disponible en Kaggle, con el objetivo de evaluar si PCA mejora el rendimiento del modelo y facilita el análisis de las variables más relevantes.

2. Conceptos de álgebra lineal y probabilidad

Antes de adentrarnos en el tema de estudio central en este documento, es fundamental repasar algunos conceptos básicos tanto de álgebra lineal como de probabilidad. Esta sección tiene como objetivo proporcionar las herramientas teóricas necesarias para comprender con claridad los resultados que se presentan más adelante.

2.1. Álgebra lineal:

2.1.0.1. Matrices

Una matriz es una disposición rectangular de números (llamados elementos) organizados en filas y columnas. Por ejemplo, una matriz de m filas y n columnas se llama matriz de tamaño $m \times n$, y se representa comúnmente como:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \tag{1}$$

Si tienes una matriz A de tamaño $m \times n$, su traspuesta, denotada como A^T, es una matriz de tamaño $n \times m$ tal que:

$$(A^T)_{ij} = A_{ji} \tag{2}$$

Por ejemplo, dada la matriz

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \tag{3}$$

su traspuesta es

$$A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \tag{4}$$

Algunas propiedades de la traspuesta, cuya demostración se puede consultar en [1] son:

- $(A^T)^T = A$
- $(A + B)^T = A^T + B^T$
- $(kA)^T = kA^T$, para cualquier escalar k
- $(AB)^T = B^T A^T$

Es decir, el elemento en la fila i y columna j de la traspuesta es igual al elemento en la fila j y columna i de la matriz original.

Una matriz cuadrada es una matriz con el mismo número de filas y columnas, es decir, de dimensión $n \times n$. Estas matrices son especialmente importantes porque permiten definir operaciones como el determinante, la traza, o conceptos como autovalores y autovectores.

2.1.0.2. Determinante de una matriz cuadrada

El determinante de una matriz cuadrada es un número que resume ciertas propiedades de la matriz, como si es invertible o no. El determinante de una matriz A de orden n se denota como $\det(A)$ o $|A|$.

Para matrices grandes, una forma de calcular el determinante es usando la expansión por cofactores (o fórmula de Laplace). Esta se hace usualmente a lo largo de una fila o una columna. Por ejemplo, para una matriz 3×3:

$$\det(A) = a_{11} \cdot C_{11} - a_{12} \cdot C_{12} + a_{13} \cdot C_{13} \tag{5}$$

donde C_{ij} es el **cofactor**, que se calcula como el determinante del menor (la submattriz que resulta al eliminar la fila i y la columna j), multilpicado por $(-1)^{i+j}$.

Algunas propiedades importantes del determinante son las siguientes:

- Si $\det(A) \neq 0$, entonces A es invertible.
- El determinante de una matriz triangular (superior o inferior) es el producto de sus elementos diagonales.
- $\det(AB) = \det(A)\det(B)$
- $\det(A^T) = \det(A)\det(A^T) = \det(A)$

Para ver una demostración de estas propiedades y otras, consulte [1].

2.1.0.3. Inversa de una matriz cuadrada

La inversa de una matriz cuadrada A (de tamaño $n \times n$) es otra matriz A^{-1} tal que:

$$A \cdot A^{-1} = A^{-1} \cdot \cdot A = I \tag{6}$$

donde $I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$ es la matriz identidad de tamaño $n \times n$, es decir, una matriz con unos en la diagonal principal y ceros en el resto.

Una matriz cuadrada A tiene inversa (es invertible o no singular) si y solo si su determinante es distinto de cero.

Si $\det(A) = 0$, la matriz no tiene inversa y se llama singular o no invertible.

2.1.0.4. Algunas matrices importantes

Una matriz triangular es una matriz cuadrada en la que todos los elementos por encima o por debajo de la diagonal principal son cero. En el primer caso, a la matriz triangular se le denomina inferior y en el segundo, se dice que la matriz triangular es superior.

Por ejemplo, las matrices $\begin{bmatrix} 2 & -1 & 3 \\ 0 & 5 & 4 \\ 0 & 0 & 7 \end{bmatrix}$ y $\begin{bmatrix} 4 & 0 & 0 \\ -2 & 1 & 0 \\ 5 & 3 & 6 \end{bmatrix}$ son triangulares superior e inferior, respectivamente.

Una matriz diagonal, por otro lado, es una matriz cuadrada en la que todos los elementos fuera de la diagonal principal son cero:

$$D = \begin{pmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_n \end{pmatrix} \tag{7}$$

Una matriz simétrica es una matriz cuadrada $A \in R^{n \times n}$ tal que $A = A^T$

Es decir, sus elementos son simétricos respecto a la diagonal principal $a_{ij} = a_{ji}$.

Las matrices simétricas tienen propiedades muy importantes, especialmente en el análisis de sistemas físicos, optimización y más.

Adicionalmente, tenemos el concepto de matrices de permutación. Una matriz de permutación es una matriz cuadrada que se obtiene al permutar las filas (o columnas) de la matriz identidad. Ejemplo de matriz identidad I_3:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Si permutamos la primera y segunda fila, obtenemos:

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Las matrices de permutación cumplen con las siguientes propiedades:

- Son ortogonales, esto es, $P^{-1} = P^T$
- Al multiplicar una matriz A por una matriz de permutación P, se reordenan las filas o columnas de A. Por ejemplo, si M es una matriz de tamaño apropiado, el producto, PM es sólo una permutación de las filas de M y MP es sólo una permutación de las columnas de M.
- Se usan en algortimos de factorización(como factorización LU) y en métodos numéricos para mejorar la estabilidad de los cálculos.

2.1.0.5. Valores y vectores propios

Dado una matriz cuadrada A, un vector propio $v \neq 0$ y un valor propio $\lambda \in \mathbb{R}$ (o $\mathbb{C}$) cumplen:

$$Av = \lambda v \tag{10}$$

Esto significa que aplicar la matriz A al vector v no cambia su dirección, solo su escala.

Para encontrar los valores propios, se resuelve la ecuación característica:

$$\det(A - \lambda I) = 0 \tag{11}$$

Los λ que satisfacen esta ecuación son los valores propios. Para cada valor propio, se pueden encontrar los vectores propios resolviendo el sistema:

$$(A - \lambda I)v = 0 \tag{12}$$

2.1.0.6. Teorema espectral

El Teorema Espectral establece que toda matriz simétrica real $A \in R^{n \times n}$ puede ser **diagonalizada** por una matriz ortogonal. Es decir, existe una matriz ortogonal Q y una matriz diagonal D tal que:

$$A = QDQ^T \tag{13}$$

donde Q tiene como columnas a los vectores propios ortonormales de A y D contiene en su diagonal los valores propios reales de A.[1]

Para una ver una prueba del mismo, consulte [2]

2.2. Variables aleatorias, esperanza y medidas muestrales

Una variable aleatoria es una función que asigna un número real a cada resultado posible de un experimento aleatorio. Por ejemplo, al lanzar un dado, podemos definir una variable aleatoria X que tome el valor del número que sale. Esta función permite modelar situaciones inciertas de manera cuantitativa.

Una de las nociones fundamentales asociadas a una variable aleatoria es su esperanza matemática o valor esperado, denotado $E[X]$. Esta representa el promedio ponderado de los posibles valores de la variable, teniendo en cuenta su probabilidad de ocurrencia. Es decir, no es simplemente un promedio aritmético, sino un promedio ajustado por cuán probable es cada valor. La esperanza ofrece una primera aproximación del "centro" de la distribución de la variable aleatoria.

En la práctica, sin embargo, rara vez conocemos la distribución completa de una variable. En su lugar, disponemos de un conjunto finito de observaciones o datos, llamado muestra. A partir de esta muestra, estimamos la esperanza mediante la media muestral, que es el promedio aritmético de los datos observados. Si contamos con n observaciones $X_1, X_2, ..., X_n$ la esperanza muestral se calcula como:

---

[1] Note que los valores propios de una matriz simétrica real son reales. En efecto, suponga que $v \in \mathbb{C}^n$ es un vector propio de A, con valor propio $\lambda \in C$. Entonces:

$$Av = \lambda v \tag{14}$$

Ahora, tome el producto interno complejo v^*Av, donde v^* es el conjugado transpuesto de v (a saber, el transpuesto de v con sus elementos conjugados). Se tendría entonces que $v^*Av = \lambda v^*v$. Pero como A es simétrica real, también se cumple que $v^*Av = (Av)^*v = (\lambda v)^*v = \lambda^* v^*v$. Entonces

$$\lambda v^*v)\lambda^* v^*v \tag{15}$$

Y como $v^*v > 0$ (producto interno positivo si $v \neq 0$):

$$\lambda = \lambda^* \tag{16}$$

lo cual implica que $\lambda \in \mathbb{R}$.

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{17}$$

Otra medida esencial es la varianza, que cuantifica cuánto se dispersan los valores de la variable respecto a su media. Formalmente, la varianza de una variable aleatoria se define como el valor esperado del cuadrado de las desviaciones respecto a la media:

$$\mathrm{Var}(X) = E\big[(X - E[X])^2\big] \tag{18}$$

Nuevamente, como en la mayoría de los casos no conocemos $E[X]$, utilizamos la **varianza muestral**, que estima la dispersión a partir de los datos. Esta se calcula como el promedio de los cuadrados de las diferencias entre cada valor observado y la media muestral, pero dividiendo entre $n-1$ en lugar de n. Este ajuste, conocido como corrección de Bessel, compensa el sesgo introducido al estimar la media a partir de los datos:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n} \big(X_i - \bar{X}\big)^2 \tag{19}$$

La división entre $n-1$ asegura que el estimador sea insesgado, es decir, que en promedio produzca el valor correcto de la varianza poblacional.

Cuando se analizan dos variables aleatorias a la vez, interesa saber si existe alguna relación entre ellas. Para eso se utiliza la covarianza, una medida que describe cómo varían conjuntamente. Si al aumentar una variable la otra también tiende a aumentar, la covarianza es positiva; si al aumentar una variable la otra tiende a disminuir, es negativa. La covarianza entre dos variables aleatorias X y Y se define como:

$$\mathrm{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])] \tag{20}$$

En la práctica, la covarianza muestral se estima a partir de datos con la fórmula:

$$s_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}\big(X_i - \bar{X}\big)\big(Y_i - \bar{Y}\big) \tag{21}$$

La covarianza muestral permite detectar patrones de dependencia entre dos variables dentro de una muestra: si ambas tienden a aumentar o disminuir juntas, su covarianza será positiva; si una crece mientras la otra decrece, será negativa. Sin embargo, el valor de la covarianza depende de las unidades de medida de las variables, lo que dificulta su comparación entre distintos contextos o escalas.

Para solucionar este problema se utiliza la correlación, que es una versión normalizada de la covarianza. La correlación muestral entre dos variables X y Y se define como:

$$r_{XY} = \frac{s_{XY}}{s_X s_Y} \tag{22}$$

donde s_{XY} es la covarianza muestral entre X y Y, y s_X, s_Y son las desviaciones estándar muestrales de X y Y, respectivamente.

El resultado es un número entre -1 y 1 que se interpreta de la siguiente forma:

- Si $r_{XY} = 1$, hay una correlación perfectamente positiva (ambas variables aumentan juntas de manera proporcional).
- Si $r_{XY} = -1$, hay una correlación perfectamente negativa (una sube mientras la otra baja en proporción).
- Si $r_{XY} = 0$, no hay una relación lineal aparente entre las dos variables.

2.3. Vectores Aleatorios y Matriz de Covarianza

Un vector aleatorio de dimensión n es un vector cuyas entradas son variables aleatorias:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_n \end{bmatrix} \tag{23}$$

Si cada para cada componente X_i, $E[X_i]$ existe, se define **la esperanza del vector aleatorio** como:

$$E[X] = \begin{bmatrix} E[X_1] \\ E[X_2] \\ E[X_3] \\ \vdots \\ E[X_n] \end{bmatrix} \tag{24}$$

La **varianza del vector aleatorio** se generaliza a través de la siguiente

$$V(X) = E\big[(X - E[X])(X - E[X])^T\big] \tag{25}$$

Esta matriz se conoce como la matriz de covarianza del vector X y se denota por Σ. Sus entradas representan las covarianzas entre los componentes del vector:

$$\Sigma = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \dots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \dots & \mathrm{Cov}(X_2, X_n) \\ \vdots & \vdots & \dots & \vdots \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & \dots & \mathrm{Var}(X_n) \end{pmatrix} \tag{26}$$

Entre algunas de las propiedades de la matriz de covarianza se encuentran las de ser simétrica y definida positiva. Para ver más propiedades y su demostración son consulte [3].

3. Sobre PCA: Procedimiento y detalles importantes

3.1. Procedimiento para realizar el análisis de componentes principales

Sean X, Y variables aleatorias. Recordemos que si X, Y son variables aleatorias, su covarianza de define como $\text{Cov}(X,Y) = E[(X - E[X])(Y - E[Y])]$. Note que $\text{Cov}(X,X) = \text{Var}(X,X)$.

Con el fin de generalizar los conceptos anteriores a dimensiones más grandes se introduce el concepto de *vector aleatorio*. Como se mencionó en la sección 2, un vector aleatorio X de dimensión n se define como un vector cuyas componentes individuales $X_1, X_2, X_3, ..., X_n$ corresponden a variables aleatorias. De esta manera, los conceptos estadísticos anteriores se pueden generalizar a dichos vectores.

Sea $X = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix}$ un vector aleatorio. Si $E[X_i] < \inf$, para $i = 1, ..., n$ recordemos que la esperanza de X como el vector $E[X] := \begin{pmatrix} E[X_1] \\ \vdots \\ E[X_n] \end{pmatrix}$.

Entonces de manera análoga al caso de una variable aleatoria, se define el operador de varianza sobre un vector aleatorio X como:

$$V(X) = E\big[(X - E[X])(X - E[X])^T\big] \tag{27}$$

Adicionalmente, para PCA un concepto muy importante es el de matriz de covarianza. La matriz de covarianza es una mariz cuadrada que contiene la covarianza entre los elementos de un vector, más específicamente, la matriz de covarianza de un vector aleatorio X se define como:

$$\Sigma = \begin{pmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & ... & \text{Cov}(X_1, X_n) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & ... & \text{Cov}(X_2, X_n) \\ \vdots & \vdots & ... & \vdots \\ \text{Cov}(X_n, X_1) & \text{Cov}(X_n, X_2) & ... & \text{Var}(X_n) \end{pmatrix} \tag{28}$$

Proposición 3.1.

Sea $X = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix}$ un vector aleatorio tal que $E[X_i] < \infty$ para $i = 1, ..., n$ y Σ su matriz de covarianza, entonces

$$\Sigma = V(X) \tag{29}$$

Prueba

Note que $X - E[X] = \begin{pmatrix} X_1 - E[X_1] \\ X_2 - E[X_2] \\ X_3 - E[X_3] \\ \vdots \\ X_n - E[X_n] \end{pmatrix}$, entonces

$$(X - E[X])(X - E[X])^T = \begin{pmatrix} X_1 - E[X_1] \\ X_2 - E[X_2] \\ X_3 - E[X_3] \\ \vdots \\ X_n - E[X_n] \end{pmatrix} (X_1 - E[X_1] \;\; X_2 - E[X_2] \;\; X_3 - E[X_3] \;\; \ldots \;\; X_n - E[X_n]) \tag{30}$$

$$= \begin{pmatrix} (X_1 - E[X_1])^2 & (X_1 - E[X_1])(X_2 - E[X_2]) & \ldots & (X_1 - E[X_1])(X_n - E[X_n]) \\ (X_2 - E[X_2])(X_1 - E[X_2]) & (X_2 - E[X_2])^2 & \ldots & (X_2 - E[X_1])(X_n - E[X_n]) \\ \vdots & \vdots & \ldots & \vdots \\ (X_n - E[X_n])(X_1 - E[X_2]) & (X_n - E[X_n])(X_2 - E[X_2]) & \ldots & (X_n - E[X_n])^2 \end{pmatrix} \tag{31}$$

De esta manera,

$$E((X - E[X])(X - E[X])^T) \tag{32}$$

$$= \begin{pmatrix} E\left[(X_1 - E[X_1])^2\right] & E[(X_1 - E[X_1])(X_2 - E[X_2])] & \ldots & E[(X_1 - E[X_1])(X_n - E[X_n])] \\ E[(X_2 - E[X_2])(X_1 - E[X_2])] & E\left[(X_2 - E[X_2])^2\right] & \ldots & E[(X_2 - E[X_1])(X_n - E[X_n])] \\ \vdots & \vdots & \ldots & \vdots \\ E[(X_n - E[X_n])(X_1 - E[X_2])] & E[(X_n - E[X_n])(X_2 - E[X_2])] & \ldots & E\left[(X_n - E[X_n])^2\right] \end{pmatrix} \tag{33}$$

$$= \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \ldots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \ldots & \mathrm{Cov}(X_2, X_n) \\ \vdots & \vdots & \ldots & \vdots \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & \ldots & \mathrm{Var}(X_n) \end{pmatrix} [*] \tag{34}$$

Pero la expresión $[*]$ corresponde precisamente a la definición de matriz de covarianza, por lo que $V(X) \coloneqq E[(X - E[X])(X - E[X])^T] = \Sigma\blacksquare$.

Entre algunas de las propiedades de la matriz de covarianza se encuentra que es definida positiva y simétrica (recuerde que $\mathrm{Cov}(X_i, X_j) = \mathrm{Cov}(X_j, X_i)$). Para una prueba detallada de estas propiedades y otras diríjase a [3]

Recordemos que el teorema espectral nos dice que una matriz cuadrada con entradas en los reales es simétrica si y sólo si es diagonalizable ortogonalmente, esto es, una matriz cuadrada $n \times n$, A es simétrica si y sólo si podemos encontrar una **matriz ortogonal** Q y una **matriz diagonal** D tales que

$$A = QDQ^T = QDQ^{-1} \tag{35}$$

Por lo tanto, como la matriz de covarianza es simétrica, también es diagonalizable ortogonalmente. Sea $QDQ^{-}1$ su descomposición.

Las siguientes dos proposiciones serán relevantes en nuestro estudio de PCA:

Proposición 3.2.

Para la descomposición QDQ^T de la matriz de covarianza Σ anterior, si se reordenan los valores propios en D, por ejemplo, intercambiando λ_i con λ_j, y se hace exactamente el mismo

intercambio en las columnas correspondientes q_i y q_j de Q obteniendo nuevas matrices D' y Q', se tiene que Q' será ortogonal y $\Sigma = Q'D(Q')^T$.

Prueba

Una matriz es ortogonal si y sólo si sus columnas forman un conjunto de vectores ortonormales. Intercambiar dos de sus columnas no altera esta ortonormalidad (el conjunto seguirá siendo el mismo), así que Q' será también ortogonal.

Por otro lado, hacer ese intercambio descrito que afecta tanto a Q como a D básicamente corresponde a hacer una permutación coordinada. En efecto, sea P la **matriz de permutación** que intercambia las columnas i y j de la matriz Q produciendo Q'. Note que por ser matriz de permutación es ortogonal, entonces $P^T = P^{-1}$. El intercambio de λ_i con λ_j corresponde a un intercambio de las filas i y j y luego de las columnas i y j; en este orden, podemos expresar el intercambio de λ_i y λ_j como P^TDP. Luego, $D' = P^TDP$ y se tiene que

$$Q'D'(Q')^{-1} = (QP)(P^TDP)(P^TQ^T) = Q(PP^T)D(PP^T)Q^T = QDQ^T = \Sigma \quad (36)$$

Proposición 3.3.

Definimos $M \in \mathbb{R}^{n\times n}$ como una matriz diagonal ocn entradas en $\{+1, -1\}$, es decir:

$$M = \text{diag}(d_1, d_2, ..., d_n), d_i \in \{+1, -1\} \quad (37)$$

Sea

$$\hat{Q} = QM \quad (38)$$

, donde Q es la matriz ortogonal de la descomposición de Σ dada, entonces:

a) $\hat{Q}$ sigue siendo ortogonal

b) Puede elegirse M para que $\det\left(\hat{Q}\right) = 1$

c) $QDQ^T = \hat{Q}D\hat{Q}^T$

Prueba

a)

$$\hat{Q}^T\hat{Q} = M^TQ^TQM = M^TM = I \quad (39)$$

b) Recuerde que $\det\left(\hat{Q}\right) = \det(Q)\det(M)$ por propiedades de determinante. Como Q es ortogonal su determinante es 1 o –1. Note que independientemente del valor del determinante de Q, como $\det(M) = \prod_{i=1}^{n} d_i = \pm 1$, siempre se puede ajustar un signo en M para forzar que el producto total sea 1.

c)

$$(QM)D(QM)^T = QMDM^TQ^T \quad (40)$$

Como M es diagonal con ± 1 en la diagonal, entonces

$$MDM^T = MDM = M \quad (41)$$

pues $(MDM)_{ij} = m_i d_i m_j = d_i \delta_{ij}$ (sigue siendo D). Entonces $(QM)D(QM)^T = QDQ^T$

Por las dos proposiciones anteriores podemos seleccionar una descomposición ortogonal para Σ, QDQ^T tal que las entradas de la matriz diagonal D serán los valores propios ordenados de la forma $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ y adicionalmente, podemos escoger una matriz Q que cumpla que $\det(Q) = 1$. Más adelante veremos que el ordenar los valorEs propios facilita el identificar las direcciones de mayor varianza en PCA, mientras tanto nos centraremos en la suposición de que $\det(Q) = 1$. Bajo esta hipótesis, el cambio de variables $\mathrm{x}' = Q^T \mathrm{x}$ corresponde a una **rotación de ejes**. Básicamente, al hacer $\mathrm{x}' = Q^T \mathrm{x}$ estamos expresando el vector x en la base ortonormal (respecto al producto escalar euclídeo) formada por los vectores columna de Q, denotados como $q_1, ..., q_n$. Es decir, proyectamos x sobre cada *dirección principal* (este es el nombre que le daremos a los vectores q_i) haciendo $\mathrm{x}_i{}' = \langle q_i, \mathrm{x} \rangle$. Así habremos cambiado el sistema de coordenadas.

3.2. Matriz de covarianza de un conjunto de datos

Para ir concretando las ideas de cómo funciona PCA sobre un conjunto de datos, suponga que tenemos m características o atributos que se desean medir para n trabajadores. Estas características asumen valores en los reales, entonces cada muestra x_i(dada por una medición de las m características para un trabajador i) será un vector en $\mathbb{R}^m$, esto es, $\mathrm{x} = \begin{pmatrix} \mathrm{x}_{1i} \\ \mathrm{x}_{2i} \\ ... \\ \mathrm{x}_{mi} \end{pmatrix}$ donde cada x_{ji} corresponde al valor que se obtuvo para el trabajador i en la característica j. Denotaremos por X a la matriz $m \times n$ cuyas columnas serán los vectores x_i para $i = 1, ..., n$, es decir, $X = [\mathrm{x}_1 \mid \mathrm{x}_2 \mid ... \mid \mathrm{x}_n]$ donde cada fila j estará formada por las mediciones de los trabajadores $1, 2, 3, ..., n$ para la caracaterística j. Ahora consideraremos la matriz

$$Y \coloneqq [\ \mathrm{x}_1 - \bar{\mathrm{x}} \mid \ \mathrm{x}_2 - \bar{\mathrm{x}} \mid ... \mid \mathrm{x}_n - \bar{\mathrm{x}}\ ] \tag{42}$$

donde $\bar{x} \coloneqq \frac{1}{n} \sum_{i=1}^{n} \mathrm{x}_i$. Básicamente, aquí habremos centrado los datos (note que la suma de cada fila de Y será cero). Más adelante se explicará detalladamente el porqué es importante centrar los datos, pero por el momento, puede decirse a grandes rasgos que en un conjunto de datos no centrados, la matriz de covarianza también incluye las contribuciones de las medias, lo que podría confundir la **varianza** y el **sesgo**. En ese sentido, el centrado elimina el **sesgo sistemático** y garantiza que las componentes principales halladas con PCA (que se definirá a continuación) representen fielmente la variabilidad subyacente.

Definiremos ahora la matriz $S \coloneqq \frac{1}{n} YY^T$. Note que esta matriz está dada por

$$\frac{1}{n} YY^T = \frac{1}{n} \begin{pmatrix} (\mathrm{x}_{11} - \bar{\mathrm{x}}) & (\mathrm{x}_{12} - \bar{\mathrm{x}}) & ... & (\mathrm{x}_{1n} - \bar{\mathrm{x}}) \\ (\mathrm{x}_{21} - \bar{\mathrm{x}}) & (\mathrm{x}_{22} - \bar{\mathrm{x}}) & ... & (\mathrm{x}_{2n} - \bar{\mathrm{x}}) \\ \vdots & \vdots & ... & \vdots \\ (\mathrm{x}_{m1} - \bar{\mathrm{x}}) & (\mathrm{x}_{m2} - \bar{\mathrm{x}}) & ... & (\mathrm{x}_{mn} - \bar{\mathrm{x}}) \end{pmatrix} \begin{pmatrix} (\mathrm{x}_{11} - \bar{\mathrm{x}}) & (\mathrm{x}_{21} - \bar{\mathrm{x}}) & ... & (\mathrm{x}_{m1} - \bar{\mathrm{x}}) \\ (\mathrm{x}_{12} - \bar{\mathrm{x}}) & (\mathrm{x}_{22} - \bar{\mathrm{x}}) & ... & (\mathrm{x}_{m2} - \bar{\mathrm{x}}) \\ \vdots & \vdots & ... & \vdots \\ (\mathrm{x}_{1n} - \bar{\mathrm{x}}) & (\mathrm{x}_{2n} - \bar{\mathrm{x}}) & ... & (\mathrm{x}_{mn} - \bar{\mathrm{x}}) \end{pmatrix} \tag{43}$$

$$= \frac{1}{n}\begin{pmatrix} \sum_{i=1}^{n}(\mathrm{x}_{1i}-\bar{\mathrm{x}})^2 & \sum_{i=1}^{n}(\mathrm{x}_{1i}-\bar{\mathrm{x}})(\mathrm{x}_{2i}-\bar{\mathrm{x}}) & \dots & \sum_{i=1}^{n}(\mathrm{x}_{1i}-\bar{\mathrm{x}})(\mathrm{x}_{mi}-\bar{\mathrm{x}}) \\ \sum_{i=1}^{n}(\mathrm{x}_{2i}-\bar{\mathrm{x}})(\mathrm{x}_{1i}-\bar{\mathrm{x}}) & \sum_{i=1}^{n}(\mathrm{x}_{2i}-\bar{\mathrm{x}})^2 & \dots & \sum_{i=1}^{n}(\mathrm{x}_{2i}-\bar{\mathrm{x}})(\mathrm{x}_{mi}-\bar{\mathrm{x}}) \\ \vdots & \vdots & \dots & \vdots \\ \sum_{i=1}^{n}(\mathrm{x}_{mi}-\bar{\mathrm{x}})(\mathrm{x}_{1i}-\bar{\mathrm{x}}) & \sum_{i=1}^{n}(\mathrm{x}_{mi}-\bar{\mathrm{x}})(\mathrm{x}_{2i}-\bar{\mathrm{x}}) & \dots & \sum_{i=1}^{n}(\mathrm{x}_{mi}-\bar{\mathrm{x}})^2 \end{pmatrix} \tag{44}$$

Pero recuerde que para dos variables aleatorias W, Z, su covarianza muestral está dada por $\mathrm{Cov}(W,Z) = \frac{1}{n}\sum_{i=1}^{n}(w_i-\bar{w})(z_i-\bar{z})$, entonces cada entrada i,j de la matriz S corresponde a la covarianza muestral de los vectores que aparecen en las filas i,j de la matriz Y o más precisamente, de las variables aleatorias que se muestrearon para las características i y j. Esta matriz es la matriz covarianza que usaremos en PCA.

Por un momento dejaremos la matriz de covarianza a un lado y nos centraremos en Y. Para cada columna y_i de Y, se tiene que si $a := \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$ es un vector en $\mathbb{R}^{n\times 1}$,

$$\sum_{i=1}^{n} a_j y_j = a^T Y \tag{45}$$

Nos interesa la varianza de cada $z_j = a^T y_j$, puesto que mide qué tan dispersos están los valores proyectados de nuestros datos sobre una dirección a en el espacio. a lo largo de esa dirección. Se puede decir, entonces que si la nube de puntos es muy larga en la dirección a, entonces las proyecciones z_i estarán muy separadas o en otras palabras, tendrán alta varianza y si la nube de puntos es muy estrecha en esa dirección las proyecciones estarán muy juntas, lo que significa baja varianza.

Consideramos entonces la varianza de $a^T Y$:

$$\mathrm{Var}(a^T X) = \mathrm{Var}\Big((a^T Y - E[a^T Y])(a^T Y - E[a^T Y])^T\Big) \tag{46}$$

Pero recordemos que la media de cada fila de Y es cero, luego, lo anterior es igual a

$$\mathrm{Var}(z_i) = \frac{1}{n}\sum_{i=1}^{n}(z_1)^2 = \frac{1}{n}\,\|z\|^2 = \frac{1}{n}\,\|a^T Y\|^2 = \frac{1}{n}(a^T Y)(a^T Y)^T \tag{47}$$

$$= \frac{1}{n} a^T Y Y^T a = a^T \Sigma a \tag{48}$$

lo cual nos dice que si queremos maximizar $\mathrm{Var}(a^T Y)$ necesitamos encontrar un vector $m-$ dimensional que maximice $a^T \Sigma a$.

Se impone una restricción adicional de requerir que $a^T a = 1$ (es decir, el vector a tiene norma unitaria) para garantizar que estamos buscando la dirección óptima en el espacio de vectores sin estar influenciados por la magnitud del vector. Sin esta restricción, podría lograr una forma cuadrática más grande $a^T S a$ simplemente escalando a para que tenga una norma más grande, lo que no proporcionaría una visión significativa de la varianza de los datos a lo largo de las direcciones en el espacio de características.

De este modo, esto se convierte en una problema de optimización donde usaremos un **multiplicador de Lagrange** para maximizar $a^T Sa - \lambda(a^T a - 1)$.

El lagrangiano para el problema está dado por

$$L(a, \lambda) = a^T Sa - \lambda(a^T a - 1) \tag{49}$$

con las condiciones $\max_a a^T Sa, \quad a^T a = 1$.

Tenemos que calcular $\nabla_a L(a, \lambda)$ lo cual se hace usando las derivadas matriciales:

$$\nabla_a L(a, \lambda) = 2Sa - 2\lambda a \tag{50}$$

porque $\frac{\partial a^T Sa}{\partial a} = 2Sa$ dado que S es simétrica, $\frac{\partial a^T a}{\partial a} = \frac{\partial \|a\|^2}{\partial a} = 2a$ y $\frac{\partial(1)}{\partial(a)} = \mathbf{0}$.

Y si igualamos a cero

$$2Sa - 2\lambda a = 0 \rightarrow Sa = \lambda a \tag{51}$$

Lo que nos permite concluir que a debe ser un valor propio de S y λ su valor propio.

Los m vectores propios $(a_1, a_2, ..., a_m)$ que obtendremos serán el conjunto completo de vectores propios de S. La combinación lineal $a_k^T Y$ será lo que denominaremos *componente principal* y para la obtención de cada una se tiene que el ordenamiento de los valores propios de la matriz de covarianza juega un papel importante, pues consideramos los valores propios con el valor más alto, ya que un valor propio más alto significará que se captura más varianza. Esto se debe a que un valor λ más alto resultará en un vector de magnitud mayor a lo largo del vector propio, es decir, estamos recuperando la máxima información de la matriz de covarianza. Se utilizan los vectores propios correspondientes para esos valores propios.

Más específicamente, haremos la rotación $Z = Q^T Y$ donde Q es la matriz ortogonal con restricción $\det(Q) = 1$ de la diagonalización ortogonal de Σ. Podría usarse una matriz Q ortogonal con determinante -1, puesto que esa inversión no afecta el subespacio generado y solo cambia la orientación como si de un espejo se tratase; sin embargo, aunque es equivalente estadísticamente, no se suele desear si uno quiere mantener consistencia geométrica. Por ejemplo, en aplicaciones como compresión, interpretación de componentes, o visualización, mantener la orientación es útil y evita confusión.

Note que si $Q = [q_1 \mid q_2 \mid q_3 \mid ...\mid q_m]$, entonces cada fila de $Z = \begin{bmatrix} q_1^T \\ q_2^T \\ q_3^T \\ \vdots \\ q_m^T \end{bmatrix} [y_1 \mid y_2 \mid ... \mid y_n] = \begin{bmatrix} q_1^T y_1 & q_1^T y_2 & \cdots & q_1^T y_n \\ q_2^T y_1 & q_2^T y_2 & \cdots & q_2^T y_n \\ \vdots & \vdots & \vdots & \vdots \\ q_m^T y_1 & q_m^T y_2 & \cdots & q_m^T y_n \end{bmatrix}$. Cada fila z_k^T de Z es la combinación lineal:

$$z_k = q_k^T y_i = \left[q_k^T y_1, q_k^T y_2, ..., q_k^T y_n\right] \tag{52}$$

donde cada $q_k^T y_i$ es la proyección del dato y_i sobre el eje q_k. Recordemos que los vectores q_k conforman una base ortonormal para $\mathbb{R}^m$, donde m es la cantidad de *features* o características

y que corresponden a vectores propios de Σ, por lo que las filas de Z serán las componentes principales.

Observe que los datos originales se pueden recuperar por medio de la matriz Z. En efecto, si z_i denota la i —ésima columna de la matriz Z, entonces para $i = 1, ..., n$ tenemos que

$$x_i = y_i + \bar{x} = Qz_i + \bar{x} \tag{53}$$

Para la reducción de dimensionalidad, recordemo que λ_i mide la varianza de los datos en la dirección del vector propio q_i, es decir, si proyectamos todos los datos sobbre la direción q_i, la varianza de esas proyecciones será exactamente λ_i. En este sentido, si λ_i es pequeño (cercano a cero), habrá muy poca varianza en la dirección q_i (los datos casi no se dispersan hacia esa dirección) y más aún, si $\lambda_i = 0$, todos los datos estarán contenidos en un **hiperplano ortogonal** a q_i, o en otras palabras, no habrá información útil (ni variación) en esa dirección. En conclusión, un λ_i pequeño quiere decir que esa dirección no aporta significativamente a describir la forma de la nube de datos.

Suponga que los valores propios $\lambda^{k+1}, \lambda_{k+2}..., \lambda_m$, son muy cercanos a cero, entonces la reducción de dimensionalidad se haría eliminando los vectorios propios asociados a esos valores propios de Q, de tal forma que $\bar{Z} = [q_1|q_2|...|q_k]^T Y$, es decir, solamente tomamos las primeras k componentes principales. Geométricamente, esto puede pensarse como la proyección de los datos rotados al subespacio generado por los vectores $q_1, ..., q_k$. Básicamente, la información de los m atributos originales la estamos resumiendo en los k atributos dados por las filas de $\bar{Z}$. Finalmente, la eliminación de características redundantes nos permitirá analizar de una mejor manera los datos iniciales.

3.3. Detalles importantes de PCA

PCA es una técnica basada en álgebra lineal que transforma las variables originales en combinaciones lineales (componentes principales). Recordemos que, para poder hacer esto, necesita: Calcular varianzas y covarianzas Construir una matriz de correlación o covarianza Aplicar descomposición en valores propios (eigen decomposition)

Pero estas operaciones sólo tienen sentido con datos numéricos. Sin embargo, en general, no es necesario que todas las variables numéricas se incluyan sin más. En efecto, existen algunas buenas prácticas al momento de seleccionar variables para PCA, entre las cuales se encuentran:

3.3.0.1. Eliminar columnas con varianza cercana a 0

Una variable con varianza cercana a cero significa que casi todos los datos tienen el mismo valor (por ejemplo, 99% de las filas tienen “5”). Por lo que, puede decirse que, estas variables no aportan variabilidad al conjunto de datos; además, como el PCA se basa en encontrar las direcciones de mayor varianza, una variable constante no influye en los componentes principales y finalmente, incluirlas puede introducir ruido innecesario o afectar la interpretación del resultado.

3.3.0.2. Filtrar variables numéricas que tengan sentido conjunto

Si se mezclan variables de naturaleza diferente (por ejemplo: datos financieros con características técnicas), los componentes pueden terminar siendo combinaciones arbitrarias de cosas

que no tienen relación lógica. Esto genera componentes difíciles de interpretar, lo cual limita su utilidad práctica. Así que es mejor aplicar PCA a grupos de variables que analicen aspectos similares del problema.

3.3.0.3. Asegurar que estén escaladas

El PCA se basa en varianza, y la varianza depende del orden de magnitud de la variable. Las variables con valores más grandes dominan la varianza total y, por tanto, influyen más en los componentes principales. Usar una técnica como StandardScaler (que pone media = 0 y desviación estándar = 1) es algo que se recomienda hacer para evitar este sesgo, y asegurar que todas las variables aporten de manera equilibrada al análisis.

3.3.0.4. Manejo de valores faltantes antes de aplicar PCA

Las operaciones que se realizan en PCA no pueden llevarse a cabo si hay valores nulos (NaN). Es por tal razón que los valores faltantes deben imputarse, usualmente con la media o mediana, o con algún método más sofisticado. Si no se hace, el PCA fallará directamente o se eliminarán automáticamente filas, lo cual puede distorsionar los resultados o reducir la muestra útil.

3.3.0.5. Consideraciones acerca de lo que es un valor propio "pequeño"

En análisis de componentes principales (PCA) y análisis factorial, un valor propio o eigenvalue representa la cantidad de varianza explicada por cada componente o factor. Tradicionalmente, según el criterio de Kaiser, se consideran relevantes solo aquellos factores con eigenvalues mayores a 1, ya que aportan más varianza que una variable original individual. Sin embargo, este criterio no es absoluto ni debe utilizarse como única regla para decidir la retención de factores.

Un eigenvalue se considera "pequeño" cuando es menor que 1, indicando que el factor o componente explica menos varianza que una variable original. Sin embargo, factores con eigenvalues ligeramente inferiores a 1 (por ejemplo, entre 0.95 y 0.99) pueden ser valiosos si cumplen ciertos criterios complementarios, como tener una carga factorial significativa, aportar una proporción relevante de la varianza total, ser consistentes con la teoría subyacente del estudio y ser apoyados por métodos como el análisis paralelo o la inspección del gráfico de sedimentación.

Por lo tanto, la decisión de retener factores con eigenvalues pequeños debe basarse en un análisis integral y no únicamente en un umbral fijo. Además, la retención de tales factores debe ser transparente y justificada, pues la inclusión de factores con eigenvalues bajos puede llevar a sobreextraer componentes o dificultar la interpretación.

Véase [4] para más información al respecto.

3.3.0.6. Escalado antes de PCA

Uno de los detalles cruciales a considerar antes de aplicar análisis de componentes principales es si se deben escalar o estandarizar los datos. La decisión de escalar afecta directamente el resultado del análisis, ya que modifica la matriz sobre la cual se realiza la descomposición espectral.

Cuando calculamos la matriz de covarianza Σ, estamos evaluando la variabilidad absoluta de los datos. Si una variable tiene una varianza mucho mayor que las demás, entonces su influencia numérica en la matriz será mucho mayor. Por ejemplo suponga que X_1 varía entre 1 y 10 y X_2 varía entre 1,000 y 10,000 . Una regla empírica bastante usada cuando no tenemos todos los datos dice que $\sigma \approx \frac{\text{valor máximo}-\text{valor mínimo}}{4}$ (véase [5]). Así que $\sigma_1 \approx 3$ y $\sigma_2 \approx 3000$, entonces la matriz de covarianza Σ estará "dominada" por los valores grandes en la dirección de X_2. Como consecuencia el primer componente principal (el de mayor varianza) apuntará casi en la dirección de X_2. Vemos que esto ocurre no porque X_2 sea más relevante, sino porque su escala es mayor. Entonces al aplicar PCA sobre Σ, estaremos maximizando la varianza en unidades absolutas.

Por lo anterior, se recomienda estandarizar los datos, transformando cada variable para que tenga media cero y desviación estándar uno:

$$Y_{ij} = \frac{X_{ij} - \mu_j}{\sigma_j} \tag{54}$$

$$R = \text{cor}(X) = \left(\frac{1}{n}\right) \cdot Y^T Y \tag{55}$$

Aquí, cada columna de Y tiene media cero y varianza uno.

Esta matriz tiene 1′s en la diagonal (porque cada variable estandarizada tiene varianza 1), y los coeficientes de correlación en las entradas fuera de la diagonal.

De este modo, al aplicar PCA sobre R, estaremos extrayendo componentes principales que maximizan la varianza relativa entre variables, sin que ninguna domine debido a su escala. Es decir, todas las variables contribuyen en igualdad de condiciones al cálculo.

Los autovalores λ_i de R nos dirán qué fracción de la varianza total estandarizada explican los componentes principales. Como la suma total de las varianzas estandarizadas es k (una por variable), se cumple que $\lambda_1 + \lambda_2 + ... + \lambda_k = k$ y así, cada λ_i indicará la proporción de información estructural explicada por el i —ésimo componente principal, sin sesgo por escalas.

4. PCA y análisis de malware

4.1. Modelos de predicción para analizar malware

Un malware (del inglés malicious software) es cualquier programa informático diseñado con la intención de dañar, interrumpir, infiltrarse o controlar sistemas informáticos, servidores o redes sin el consentimiento del usuario. Existen diversos tipos de malware que se clasifican según su comportamiento, objetivo o método de propagación (para más detalles, véase [6]).

El análisis de malware es el proceso de estudiar el comportamiento, las características y el código de software malicioso con el propósito de identificar infecciones, evaluar el alcance del daño, descubrir cómo se llevó a cabo la intrusión, determinar su origen, identificar las vulnerabilidades explotadas y prevenir futuras amenazas. En esencia, puede entenderse como el arte de disecar malware.

A lo largo del tiempo, se han desarrollado múltiples técnicas y metodologías que, combinadas, permiten realizar análisis de muestras de malware de manera sistemática y efectiva. Sin embargo, el panorama de amenazas evoluciona rápidamente. Los avances tecnológicos, el crecimiento exponencial del software y la creciente complejidad de los entornos computacionales han impulsado la aparición de nuevas variantes de malware. Algunas son versiones modificadas de amenazas antiguas que han sido adaptadas para evadir mecanismos de detección modernos; otras son completamente nuevas, diseñadas para explotar entornos actuales como servicios en la nube, dispositivos móviles o infraestructuras de inteligencia artificial.

Por ejemplo, hace algunos años bastaba con examinar las cadenas de texto embebidas en un ejecutable para detectar comportamientos maliciosos. Hoy en día, sin embargo, existen **malware polimórficos y metamórficos** que alteran su estructura internamente en cada ejecución, e incluso algunos que integran técnicas de aprendizaje automático para modificar dinámicamente su comportamiento en función del entorno en el que se ejecutan. Estas nuevas capacidades hacen que los enfoques tradicionales resulten insuficientes, y exigen la incorporación de técnicas avanzadas, como el análisis de comportamiento o el aprendizaje automático.

Básicamente, con el crecimiento exponencial del número de muestras de malware, especialmente a partir de la década de 2010, cuando se comenzaron a publicar artículos donde se hablaba sobre el uso de machine learning para hacer detección de malware , las técnicas tradicionales de detección —como las basadas en **firmas estáticas**— comenzaron a volverse ineficaces ante variantes cada vez más sofisticadas, como el malware polimórfico y metamórfico. Algunos artículos fundacionales pueden verse en las referencias [7], [8]. Y en ese sentido, fue a partir de 2010, cuando la industria (por ejemplo, AV-Test, Kaspersky y Symantec) reportó un crecimiento exponencial en el número de variantes de malware creadas automáticamente por herramientas de **obfuscation, packers y malware metamórfico** (vea [9]). Todo esto motivó el surgimiento de métodos basados en aprendizaje automático para automatizar y mejorar la detección.

A grandes rasgos, el proceso general para aplicar modelos predictivos en la detección de malware suele seguir las siguientes etapas:

1. Extracción de características (feature extraction): a partir de una muestra de malware o software benigno, se extraen características relevantes. Estas pueden ser estáticas (como

opcodes, llamadas al sistema, hashes, **cadenas de texto**, etc.) o dinámicas (como comportamientos observados en entornos **sandbox**).

2. Construcción de un dataset: cada muestra se representa como un vector de características. Se etiqueta con su clase correspondiente (por ejemplo, "malicioso" o "benigno").

3. Preprocesamiento: se limpian y normalizan los datos. Aquí pueden aplicarse técnicas de selección o reducción de dimensionalidad.

4. Entrenamiento de modelos: En esta fase se entrenan **clasificadores supervisados** (como **Random Forests, SVM, Redes Neuronales**) que aprenden a distinguir entre software malicioso y benigno en función de sus características.

5. Evaluación y despliegue: los modelos se validan con métricas como **precisión, recall y AUC**, y luego se despliegan para analizar nuevas muestras automáticamente.

4.2. Aplicación de PCA en la detección de malware

En general, los vectores de características extraídos de las muestras de malware suelen tener cientos o miles de dimensiones. En este contexto, el análisis de componentes principales (PCA) ha sido un enfoque que ha ganado relevancia como técnica previa al entrenamiento del modelo, por varias razones entre las cuales se encuentra la reducción dimensionalidad que permite reducir las características a usar conservando la información más significativa (es decir, la que explica mayor varianza), lo que mejora la eficiencia computacional y reduce el riesgo de sobreajuste, además de potenciar el rendimiento de los clasificadores y facilitar la representación gráfica y el análisis exploratorio permitiendo explorar la separación entre clases y detectar patrones ocultos.

El uso de PCA en modelos de predicción de infección por malware ha sido respaldado por estudios como *Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance* (2019) donde se propone un sistema robusto de clasificación de tráfico que utiliza PCA y Redes Neuronales Artificiales (RNA) para proporcionar una vigilancia extrema. Además, este método propuesto busca exponer diversos ciberataques basados en IA, su impacto actual y su futuro. La simulación se realiza mediante un agente autónomo de desarrollo propio que aprende por sí mismo. Los resultados experimentales confirmaron que los esquemas propuestos fueron eficientes para clasificar el tráfico de ataque con un 99% de precisión en comparación con los métodos de estado del arte.

En otro artículo titulado *Features Reduction with PCA Technique for Malware Detection Using Machine Learning* (2023) se usó PCA para reducción de características y luego se utilizaron cuatro clasificadores de aprendizaje automático (KNN, árbol de decisión Naïve Bayes y un bosque aleatorio) para la detección de malware. Los resultados demostraron que el mejor rendimiento provenía de la detección utilizando bosque aleatorio con una precisión promedio de 0,991688.

Existen más estudios que han evaluado la aplicabilidad de PCA en la detección de malware (vea [10], [11], [12]) y cuyos resultados han sido variados (algunos determinaron una mejora en el rendimiento de ciertos modelos, mientras que otros no vieron que se afectara la precisión de modelos de predicción de malware que usaban PCA).

4.3. Implementación: Microsoft Malware Prediction

4.3.0.1. Primer entregable

Dada la magnitud y complejidad del conjunto de datos original, cuya carga completa excede la capacidad de procesamiento eficiente en memoria, se optó por un enfoque por muestreo sistemático para realizar el análisis exploratorio inicial. Este enfoque no solo mitigó las restricciones computacionales, sino que también permitió obtener una perspectiva más representativa y estructurada del comportamiento general de los datos.

Se definió una estrategia de análisis basada en la extracción de cinco muestras aleatorias de 500 filas cada una. Las muestras fueron seleccionadas de manera aleatoria a partir del conjunto de datos ya filtrado, evitando aquellas entradas identificadas previamente como erróneas o ilegibles. Esto asegura que los datos analizados sean confiables y útiles para la posterior modelación y evaluación.

A cada una de estas muestras se le aplicó un Análisis de Componentes Principales (PCA), con el objetivo de obtener una primera visión de la estructura interna de los datos, identificar posibles agrupamientos, redundancias o direcciones principales de variabilidad. Si bien en esta fase aún no se ha definido una métrica formal para la interpretación cuantitativa de los componentes, la observación de las proyecciones resultantes permite evaluar de manera cualitativa ciertos aspectos relevantes, tales como:

- Distribución y dispersión de los puntos en el espacio reducido.
- Posibles patrones.
- Existencia de outliers u observaciones atípicas.
- Influencia de las dimensiones originales en los componentes principales.

Este análisis por bloques también proporciona un marco para comparar la estabilidad estructural del conjunto de datos: si las distintas muestras presentan comportamientos similares bajo PCA, se fortalece la hipótesis de que el conjunto tiene una distribución coherente. En cambio, variaciones abruptas entre muestras pueden indicar la presencia de subgrupos diferenciados, sesgos o la necesidad de normalización adicional.

El uso de múltiples muestras en lugar de una única instancia permite obtener una perspectiva más robusta y confiable del comportamiento general del dataset, sentando las bases para las siguientes etapas de limpieza, modelado y evaluación.

Como parte del proceso exploratorio, también se implementó un primer intento de modelo de predicción utilizando el algoritmo Random Forest. Este modelo no fue concebido como una solución final, sino como una herramienta preliminar para familiarizarse con el flujo de entrenamiento, validación y evaluación dentro del contexto del problema. La experimentación con Random Forest en esta fase proporcionó información valiosa sobre la factibilidad del modelado y sobre las transformaciones previas requeridas para optimizar el desempeño en fases posteriores.

Los resultados obtenidos en esta etapa y las conclusiones derivadas pueden consultarse en el archivo **E1Descripcion_Codigo_Analisis_Exploratorio**.

4.4. Conclusiones

[Sección a completar en el final...]

4.5. Glosario

- **AUC (Area Under the Curve):** Métrica de evaluación usada en clasificación binaria. Representa el área bajo la curva ROC y mide la capacidad del modelo para distinguir entre clases.
- **Aprendizaje automático (machine learning):** Subcampo de la inteligencia artificial que desarrolla algoritmos capaces de aprender patrones a partir de datos, con o sin supervisión humana.
- **Bosque aleatorio (Random Forest):** Algoritmo de clasificación que combina múltiples árboles de decisión para mejorar la precisión y controlar el sobreajuste.
- **Cadenas de texto:** Fragmentos legibles de caracteres encontrados en archivos ejecutables que pueden dar pistas sobre el comportamiento del software (por ejemplo, URLs, rutas o comandos).
- **Clasificadores supervisados:** Algoritmos de aprendizaje automático que aprenden a partir de ejemplos etiquetados para predecir clases futuras (e.g., benigno o malicioso).
- **Dataset:** Conjunto de datos estructurados que contienen ejemplos (muestras) con sus respectivas características y etiquetas para el entrenamiento y evaluación de modelos.
- **Feature extraction (extracción de características):** Proceso mediante el cual se identifican y seleccionan atributos relevantes de los datos (ya sea estáticos o dinámicos) para representar cada muestra.
- **Firmas estáticas:** Identificadores únicos (como hashes o patrones de código) usados tradicionalmente por antivirus para reconocer malware conocido.
- **Malware:** Programa o código malicioso diseñado para dañar, infiltrarse o controlar sistemas informáticos sin autorización.
- **Malware metamórfico:** Tipo de malware que reescribe su propio código internamente en cada ejecución para evitar su detección.
- **Malware polimórfico:** Malware que cifra o modifica su código externo (por ejemplo, por medio de packers u ofuscadores) para cambiar de apariencia sin alterar su funcionalidad.
- **Naïve Bayes:** Clasificador probabilístico basado en el teorema de Bayes. Asume independencia entre características, lo que lo hace rápido pero limitado en ciertos contextos.
- **Obfuscation (ofuscación):** Técnica para dificultar el análisis de un programa, alterando su código o estructura sin cambiar su funcionalidad.
- **Opcode (código de operación):** Instrucción que especifica la operación que debe realizar el procesador. Su análisis permite inferir patrones de ejecución de un binario.

- **PCA (Principal Component Analysis):** Técnica estadística de reducción de dimensionalidad que transforma los datos originales en un conjunto reducido de variables no correlacionadas (componentes principales) que retienen la mayor varianza posible.
- **Precisión (accuracy):** Métrica de evaluación que indica la proporción de predicciones correctas sobre el total de muestras.
- **Recall (sensibilidad):** Métrica que mide la proporción de verdaderos positivos identificados correctamente por el modelo respecto al total real de positivos.
- **Redes Neuronales Artificiales (RNA):** Modelos inspirados en la estructura del cerebro humano, capaces de aprender relaciones complejas entre datos de entrada y salida.
- **Sandbox:** Entorno controlado y aislado donde se ejecutan programas sospechosos para observar su comportamiento sin poner en riesgo el sistema real.
- **SVM (Support Vector Machine):** Algoritmo de clasificación que encuentra el hiperplano óptimo para separar clases en el espacio de características.

Bibliography

[1] G. Strang, *Introduction to Linear Algebra*, 5th ed. Wellesley-Cambridge Press, 2016.

[2] P. J. Davis, "The Theory of Spectral Representations." [Online]. Available: https://math.mit.edu/~dav/spectral.pdf°

[3] D. C. Lay, *Matrix Algebra Useful for Statistics*, 1st ed. Hoboken, NJ: Wiley-Interscience, 2012.

[4] S. Vatansever and R. K. Gupta, "Can I use principal component and factor even though eigenvalue is lower than 1?." [Online]. Available: https://www.researchgate.net/post/Can_I_use_principal_component_and_factor_even_though_eigenvalue_is_lower_than_1°

[5] M. Bleicher, "Range Rule for Standard Deviation." [Online]. Available: https://www.thoughtco.com/range-rule-for-standard-deviation-3126231°

[6] "Malware - Wikipedia." [Online]. Available: https://es.wikipedia.org/wiki/Malware°

[7] N. Autor, "A static malware detection system using data mining methods," *Nombre de la Revista o Conferencia*, no. Número, p. páginas, 2008, doi: DOI-si-lo-tienes°.

[8] F. Toolan and J. Carthy, "Feature selection for spam and phishing detection," *eCrime Researchers Summit (eCrime)*, pp. 1–12, 2010, doi: 10.1109/eCrime.2010.5668493°.

[9] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A Survey on Automated Dynamic Malware Analysis Techniques and Tools," *ACM Computing Surveys*, vol. 44, no. 2, pp. 1–42, 2012, doi: 10.1145/2089125.2089126°.

[10] D. Suhartono and others, "Neural Network with Principal Component Analysis for Malware Detection," in *Proceedings of the 16th International Conference on Security and Cryptography*, 2019, pp. 270–277. [Online]. Available: https://www.scitepress.org/Papers/2019/99089/99089.pdf°

[11] A. Mahindru *et al.*, "Enhancing malware detection with feature selection and scaling techniques," *Scientific Reports*, vol. 15, no. 1, p. 10724, 2025, doi: 10.1038/s41598-025-93447-x°.

[12] A. Kabakus and others, "Evaluation of Principal Component Analysis Variants to Assess Their Suitability for Mobile Malware Detection," *ResearchGate*, 2022, [Online]. Available: https://www.researchgate.net/publication/361523264_Evaluation_of_Principal_Component_Analysis_Variants_to_Assess_Their_Suitability_for_Mobile_Malware_Detection°

Otras fuentes

- Bleicher, M. (2020). Range rule for standard deviation. ThoughtCo. https://www.thoughtco.com/range-rule-for-standard-deviation-3126231
- Carballo R. (s.f.). Matrices y vectores aleatorios. https://personales.unican.es/carballor/curso18_19/Tema6_Lecc3_v2.pdf
- Ciencia de Datos. (s.f.). PCA con Python. https://cienciadedatos.net/documentos/py19-pca-python
- Davis, P. J. (2000). The theory of spectral representations [Lecture notes]. https://math.mit.edu/~dav/spectral.pdf
- Economipedia. (s.f.). Covarianza. https://economipedia.com/definiciones/covarianza.html
- Eigenvector. (2020). Effect of centering on PCA. https://eigenvector.com/wp-content/uploads/2020/06/EffectofCenteringonPCA.pdf
- Fortinet. (s.f.). Malware analysis. https://www.fortinet.com/lat/resources/cyberglossary/malware-analysis
- Hiberus. (s.f.). Análisis de componentes principales. https://www.hiberus.com/crecemos-contigo/analisis-de-componentes-principales/
- Hiru. (s.f.). Propiedades de los determinantes. https://www.hiru.eus/es/matematicas/propiedades-de-los-determinantes
- Math StackExchange. (s.f.). Covariance of a random vector. https://math.stackexchange.com/questions/1741161/covariance-of-a-random-vector
- Medium. (s.f.). How PCA correlates covariance and eigen vectors using Lagrange multiplier. https://medium.com/@saurabhpage1/how-pca-correlates-covariance-and-eigen-vectors-using-lagrange-multiplier-2e9d2df48903
- NumberAnalytics. (s.f.). Comprehensive data centering guide. https://www.numberanalytics.com/blog/comprehensive-data-centering-guide#google_vignette
- Pishro-Nik, H. (2020). Random vectors. ProbabilityCourse.com. https://www.probabilitycourse.com/chapter6/6_1_5_random_vectors.php
- ResearchGate. (s.f.). PCA with one eigenvalue. https://www.researchgate.net/post/Can-principal-component-analysis-PCA-be-explained-by-just-one-component-as-only-one-eigenvalue-is-1
- Saurabh, P. (s.f.). How PCA correlates covariance and eigen vectors using Lagrange multiplier. Medium. https://medium.com/@saurabhpage1/how-pca-correlates-covariance-and-eigen-vectors-using-lagrange-multiplier-2e9d2df48903
- Scitepress. (2019). Feature selection for malware detection using PCA. In Proceedings of the 16th International Conference on Security and Cryptography. https://www.scitepress.org/Papers/2019/99089/99089.pdf
- Shalizi, C. R. (2012). Advanced data analysis from an elementary point of view — Chapter 18. https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch18.pdf
- Universidad de Valencia. (s.f.). Operador varianza. https://www.uv.es/ceaces/normaMu/operadorvarianza/varianza.htm
- Universidad Nacional de Colombia. (s.f.). Clase 24 Parte 2 - Álgebra lineal. https://ciencias.medellin.unal.edu.co/cursos/algebra-lineal/clases/8-clases/126-clase-24-parte2.html

- Universidad Politécnica de Valencia. (s.f.). Variables multidimensionales. https://personales.upv.es/asala/DocenciaOnline/material/vblesMultidimensionales.pdf
- Universidad Veracruzana. (s.f.). Matrices simétricas y diagonalización ortogonal. https://www.uv.mx/personal/aherrera/files/2014/08/30c.-MATRICES-SIMETRICAS-Y-DIAGONALIZACION-ORTOGONAL.pdf
- UTN. (s.f.). Diagonalización ortogonal. https://aga.frba.utn.edu.ar/diagonalizacion-ortogonal-de-matrices-simetricas/
- Wikipedia. (s.f.). Cálculo matricial. https://es.wikipedia.org/wiki/C%C3%A1lculo_matricial
- Wikipedia. (s.f.). Covarianza. https://es.wikipedia.org/wiki/Covarianza
- Wikipedia. (s.f.). Correlación. https://es.wikipedia.org/wiki/Correlaci%C3%B3n
- Wikipedia. (s.f.). Matriz definida positiva. https://es.wikipedia.org/wiki/Matriz_definida_positiva
- Wikipedia. (s.f.). Matriz de covarianza. https://es.wikipedia.org/wiki/Matriz_de_covarianza
- Wikipedia. (s.f.). Matriz permutación. https://es.wikipedia.org/wiki/Matriz_permutaci%C3%B3n
- Wikipedia. (s.f.). Malware. https://es.wikipedia.org/wiki/Malware
- Wikipedia. (s.f.). Producto escalar. https://es.wikipedia.org/wiki/Producto_escalar