

UNIVERSIDAD DE COSTA RICA
SISTEMA DE ESTUDIOS DE POSGRADO

**MÉTODO DE SUBESPACIOS PARA DETECCIÓN
DE ANOMALÍAS EN SERIES DE TIEMPO**

*Tesis sometida a la consideración de la Comisión del Programa de
Estudios de Posgrado en Matemática para optar al grado y título de
Maestría Profesional en Métodos Matemáticos y Aplicaciones*

CARLOS JOSÉ VARGAS AGÜERO

Ciudad Universitaria Rodrigo Facio, San José, Costa Rica

2025

*A mi papá, mamá y hermano.
Nunca habría llegado tan lejos sin ustedes.*

Agradecimientos

Quiero agradecer a los profesores Fredy Vides y Esteban Segura por toda la guía que me dieron para desarrollar el proyecto de graduación final.

También quiero agradecer a mi familia que me apoyaron en todos los momentos de mi educación. Desde la escuela hasta la conclusión de mi maestría. Nunca hubiera llegado hasta aquí sin su apoyo.

Esta tesis fue aceptada por la Comisión del Programa de Estudios de Posgrado en Matemática de la Universidad de Costa Rica, como requisito parcial para optar al grado y título de Maestría Profesional en Métodos Matemáticos y Aplicaciones.



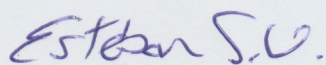
Dr. Javier Trejos Zelaya

Representante de la Decana del Sistema de Estudios de Posgrado



Dr. Fredy Antonio Vides Romero

Profesor Guía



Dr. Esteban Segura Ugalde

Lector



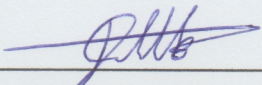
Dr. Jorge Arturo Gutiérrez Camacho

Lector



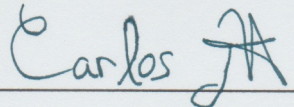
Dr. Juan Gabriel Calvo Alpízar

Lector



Dr. Darío Mena Arias

Representante del Programa de Posgrado en Matemática



Carlos José Vargas Agüero

Candidato

Índice general

Agradecimientos	III
1 Introducción	1
2 Preliminares y Notación	3
2.1. Preliminares y definiciones	3
2.2. Construcción de la matriz de Hankel	11
2.3. Construcción alternativa de la matriz de Hankel	13
3 Detección proyectiva de anomalías para series de tiempo	15
3.1. Teoría de subespacios para identificación de anomalías	15
3.2. Preliminares de la heurística del detector de anomalías	18
3.3. Heurística del PAD	23
4 Identificación de los puntos anómalos	27
4.1. Descomposición SVD de $\mathcal{H}_L(\Sigma_N)$	27
4.2. Descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$	28
4.3. Descomposición de valores propios sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$	29
4.4. Iteración IPMRQ sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$	29
4.5. Sobre la escogencia de ε	35
4.6. Complejidad de los métodos	36
4.7. Identificación de las anomalías	37
5 Experimentos y Resultados	42
5.1. Equipo y experimentos	42
5.2. Resultados de la detección de anomalías	46
5.3. Selección de distintas tolerancias	48
6 Conclusiones y futuro trabajo	56

6.1. Conclusiones	56
6.2. Disponibilidad de los datos	56
Bibliografía	57

Índice de figuras

2.1.	Señal a analizar y los puntos medidos.	11
2.2.	Construcción de la matriz de Hankel.	14
3.1.	Proyección de \mathbf{a} sobre los subespacios \mathcal{V} y \mathcal{V}^\perp usando los proyectores \mathbf{P}_r y \mathbf{Q}_r .	22
3.2.	Representación de un región de la señal $\mathbf{x}_L(t)$ sin anomalías y una región $\mathbf{y}_L(t)$ con anomalías, y las proyecciones en el subespacio $\text{colsp}(\mathbf{U}_r)$.	25
4.1.	Medición de la anomalía en el punto t_k con los segmentos antes y después incluyendo al punto. Los puntos naranjas representan anomalías.	38
4.2.	Asignación valores de $\mathbf{d}_\ell[k]$ y $\mathbf{d}_r[k]$ para el punto k .	39
4.3.	Asignación valores de $\mathbf{d}_\ell[k - L + 1]$ y $\mathbf{d}_r[k]$ para el punto k sin repetir cálculos.	39
4.4.	Asignación valores de $\mathbf{d}_\ell[k - L + 1]$ y $\mathbf{d}_r[k]$ para todos los puntos.	40
5.1.	Señal sintética con ruido añadido.	43
5.2.	Máquina vibradora de masajes con el celular para medir la aceleración.	44
5.3.	Señal de la máquina vibradora.	45
5.4.	Señal obtenida caminando.	45
5.5.	Señal sintética con las anomalías identificadas usando los valores de $L = 150$, $N = 1200$, tolerancia = 0.13 y función lógica AND.	46
5.6.	Señal de la máquina con las anomalías identificadas usando los valores de $L = 75$, $N = 1300$, tolerancia = 6.85 y función lógica OR.	47
5.7.	Señal de la caminata con las anomalías identificadas usando los valores de $L = 90$, $N = 900$, tolerancia = 0.19 y función lógica OR.	48
5.8.	Anomalías detectadas variando la tolerancia sobre la máquina vibradora, con $L = 75$, $N = 1300$ y función lógica OR.	49
5.9.	Anomalías detectadas variando la tolerancia sobre la caminata con valores $L = 900$, $N = 900$ y función lógica OR.	50
5.10.	Diagramas de caja de cada método para la señal de la máquina.	52
5.11.	Densidades de cada método para la señal de la máquina.	52

5.12. Diagramas de cada método para la señal de la caminata.	53
5.13. Densidades de cada método para la señal de la caminata.	53

Resumen

La detección de anomalías en series de tiempo es un proceso importante para el monitoreo de un sistema en tiempo real, ya que una anomalía puede indicar una situación crítica y requerir acción inmediata cuando el comportamiento anómalo ocurre, en particular para sistemas ciberfísicos, donde se pueden tener varias señales que se miden de manera constante en el tiempo.

En este trabajo se presenta un método de detección de anomalías en series de tiempo basado en subespacios de información de la propia serie temporal en comportamiento regular y un algoritmo rápido y de bajo costo computacional para detección de anomalías en series de tiempo, basado en la teoría presentada.

Además, se muestran resultados numéricos con señales reales en el tiempo generadas con las anomalías correctamente identificadas por los algoritmos propuestos. Se proporcionan todos los datos y códigos requeridos para replicar los resultados mostrados.

Abstract

The time series anomaly detection is an important process for the monitoring of a system in real time, because an anomaly may indicate a critical situation and require immediate action when an anomalous behavior happen, in particular on cyber-physical systems, where there can be various signals that being measured constantly in time.

In this document, it is presented a time series anomaly detection method based on subspaces of information of the time series in regular behavior, and a fast and low computational cost algorithm for anomaly detection for time series based on the theory presented.

Numerical results of real signal in time with the generated anomalies were correctly identified by the proposed algorithms. It is provided all the data and codes required to replicate the results shown.

Capítulo 1

Introducción

La detección de anomalías es el problema de encontrar patrones en los datos que no se conforman al comportamiento esperado [8, p.1]. Hay varios propósitos para la identificación de los puntos anómalos, por ejemplo, para hacer limpieza de los datos encontrando información que no representa medidas reales, o también para identificar eventos de interés, ya sea para analizar la anomalía o para tomar una decisión cuando se presenta alguna [4, p.2].

Dependiendo del sistema de interés, se podría necesitar tomar una acción inmediata cuando se presenta una situación anómala, por lo que se requieren métodos capaces de detectar comportamiento irregular de manera rápida. Un ejemplo de esto es detectar una transacción fraudulenta de una tarjeta de crédito, ya que muchas transacciones fraudulentas tienen un comportamiento anómalo. Así, estas transacciones tienen que ser rechazadas en el momento que se realizan en tiempo real y solo se dispone de unos segundos para decidir si aprobar la transacción o no.

En particular, si tenemos aparatos con baja capacidad computacional que miden la señal de entrada, en forma de una serie de tiempo, en los que queremos identificar comportamiento anómalo en tiempo real, es necesario desarrollar métodos que puedan usarse en equipos igualmente de baja capacidad computacional para poder identificar las anomalías. Así, el interés del trabajo es desarrollar un método para la identificación de anomalías en sistemas ciberfísicos que tienen sensores que van midiendo una señal en tiempo real, tales como los relojes inteligentes que tienen acelerómetros para medir el ritmo de una caminata o el pulso cardiaco de la persona. Comportamientos irregulares en estas señales podrían ser una situación que debe ser atendida de manera inmediata.

En este documento se desarrolla un método teórico y computacional de detección de

anomalías en series de tiempo basado en métodos de subespacio aplicados a matrices determinadas por dichas series y la idea es proyectar segmentos de la serie temporal en los subespacios para identificación de anomalías, junto con un algoritmo que no requiere capacidad grande de memoria y se corre con poca cantidad de cálculos para la identificación de estas anomalías de manera rápida.

El trabajo está organizado de la siguiente manera.

En el capítulo 2 se introduce la notación y conceptos básicos necesarios en el trabajo, junto con la formulación para la serie de tiempo de matrices de trayectoria tipo Hankel donde se obtienen segmentos con información relevante de la serie de tiempo en las columnas de las matrices de Hankel.

En el capítulo 3 se desarrolla la teoría para la detección de anomalías en series de tiempo, basándose en subespacios de información relevante de la señal y proyectando la información sobre el subespacio de información no relevante para la identificación de comportamiento anómalo y una heurística del método para reducir la cantidad de cálculos sobre la proyección en subespacios a un vector específico para identificar las anomalías.

Por otro lado, en el capítulo 4 se explican varios métodos para obtener el vector de identificación de anomalías. Además, se presenta una comparación entre estos métodos y el diseño del algoritmo para identificar los puntos anómalos sobre toda la serie de tiempo basado en los subespacios y cómo realizar el cálculo para ahorrar memoria ya habiendo encontrado este vector sobre todos los puntos.

En el capítulo 5 se explica el diseño experimental para obtener las señales a analizar, los resultados de aplicar el método de detección de anomalías sobre las señales generadas, efectos de tomar distintos umbrales para la identificación y una comparación de los distintos tiempos para obtener el vector que se usa para la identificación de regiones anómalas.

El capítulo 6 contiene las conclusiones del trabajo junto con un repositorio en GitHub con todos los códigos usados y datos disponibles para la replicación de los resultados reportados en este documento.

Capítulo 2

Preliminares y Notación

Se describe primero la notación y conceptos básicos a usar para desarrollar la teoría y el algoritmo de detección de anomalías. Además, se realiza un planteamiento para obtener la matriz de Hankel que se va a usar en todo el trabajo y un enfoque en la construcción de la matriz de Hankel con segmentos de la serie de tiempo como las columnas de la matriz, que contienen la información de la serie en comportamiento normal.

2.1. Preliminares y definiciones

En este trabajo, \mathbf{I}_n es la matriz identidad en $\mathbb{R}^{n \times n}$. Esta se usa en la Sección 4.4.

Definición 2.1 (Matriz ortogonal). [11, p. 107] Una matriz cuadrada $\mathbf{A} \in \mathbb{R}^{n \times n}$ es una matriz ortogonal si cumple con la relación

$$\mathbf{A}\mathbf{A}^T = \mathbf{I}_n,$$

donde \mathbf{A}^T es la traspuesta de \mathbf{A} . ◇

En este trabajo se utilizan dos normas, la norma euclidiana que se usan en todo el documento y la norma de Frobenius que es usada en el Teorema 3.5 y la Sec. 3.2.

Definición 2.2 (Norma Euclidea). [13, p. 270] Para un vector $\mathbf{x} \in \mathbb{R}^n$, la norma Euclidea para \mathbf{x} está definida por

$$\|\mathbf{x}\| = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}.$$

◇

Definición 2.3 (Norma de Frobenius). [13, p. 279] Para una matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$, la norma de Frobenius está definida por

$$\|\mathbf{A}\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2.$$

◇

Se utilizará el concepto de esfera de norma unidad en la Def. 2.20 para obtener un conjunto de vectores, donde lo importante es la dirección de estos vectores y no su magnitud.

Definición 2.4 (Esfera de norma unidad). El conjunto de esfera de norma unitaria, se denota con \mathbb{S}^{n-1} y está definida por

$$\mathbb{S}^{n-1} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| = 1\}.$$

◇

Para determinar si un punto de una serie de tiempo es anómalo, se usará una proyección como argumento de una función de activación, que indica si está sobre cierto umbral (Def. 4.3) y así definir el rango δ de una matriz (Def 2.14). Se usa la función $H_\delta(x)$ (Def. 2.5) para este propósito. [20, p. 2]

Definición 2.5 (Función de activación). La función $H_\delta(x)$ está definida por

$$H_\delta(x) = \begin{cases} 1 & \text{si } x > \delta, \\ 0 & \text{si } x \leq \delta. \end{cases}$$

◇

Para la detección de anomalías se generan subespacios generados por la información de la señal de entrenamiento sin anomalías. Para una señal de entrada, se proyecta la información en los subespacios de menor información (Sección 3.3) generados por los vectores izquierdos de la descomposición en valores singulares (Descomposición SVD) (Definición 2.12), por lo que se define el conjunto generador para generar estos subespacios.

Definición 2.6 (Conjunto generador). Sean L y r números enteros positivos y los vectores $\mathbf{v}_1, \dots, \mathbf{v}_r \in \mathbb{R}^L$. El conjunto generado [11, p. 242] denotado por $\text{gen}\{\mathbf{v}_1, \dots, \mathbf{v}_r\}$, es el conjunto de todas las combinaciones lineales de los vectores $\mathbf{v}_1, \dots, \mathbf{v}_r$; es decir,

$$\text{gen}\{\mathbf{v}_1, \dots, \mathbf{v}_r\} = \{\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \dots + \alpha_r \mathbf{v}_r \mid \alpha_1, \alpha_2, \dots, \alpha_r \in \mathbb{R}\}.$$

◇

En este trabajo, se usan varios métodos para encontrar un vector usado para la detección de anomalías. Entonces, se debe comparar la cantidad de cálculos (FLOPs) que requiere cada método (Sección 4.6) usando el O Grande y el qué tan rápido converge una iteración a la solución (Sección 4.4). Para esto, se recuerdan los siguientes conceptos.

Definición 2.7 (Orden de Convergencia). [6, p. 79] Sea $\{p_n\}_{n=0}^{\infty}$ una secuencia que converge a p con $p_n \neq p, \forall n$. Si las constantes positivas λ y α existen con

$$\lim_{n \rightarrow \infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lambda,$$

entonces se dice que $\{p_n\}_{n=0}^{\infty}$ converge a p con un orden α y error asintótico λ . \diamond

Definición 2.8 (FLOPs). [22, p. 3] Se le llama FLOPs a las operaciones de punto flotante (operaciones de suma, resta, multiplicación y división). La palabra viene del inglés *Floating-Point Operations*. \diamond

Definición 2.9 (O grande). [9, p. 54] Dada una función $g(n)$ positiva, se denota $O(g(n))$ al conjunto de funciones

$$O(g(n)) = \{f(n) : \text{si existen constantes positivas } c \text{ y } n_0 \text{ tal que} \\ 0 \leq f(n) \leq cg(n) \text{ para todo } n \geq n_0\}$$

\diamond

En la Sección 3.2, se discute como obtener la información más importante de una matriz con una de rango menor y obtener la información del espacio de las columnas de la matriz de menor rango.

Definición 2.10 (rango de \mathbf{A}). [22, p. 216] El rango de \mathbf{A} , que denotamos por $\text{rk}(\mathbf{A})$, es igual al número de columnas linealmente independientes de \mathbf{A} (rk de *rank* en inglés). \diamond

Definición 2.11 (Espacio de columnas y núcleo de \mathbf{A}). [22, p. 240] El espacio de columnas de $\mathbf{A} \in \mathbb{R}^{m \times n}$, se denota por $\text{colsp}(\mathbf{A})$, y está definido por

$$\text{colsp}(\mathbf{A}) = \{\mathbf{Ax} \mid \mathbf{x} \in \mathbb{R}^n\}.$$

Además, el núcleo (kernel en inglés) de \mathbf{A} se denota por $\text{ker}(\mathbf{A})$ y está definido por

$$\text{ker}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} = \mathbf{0}\}.$$

\diamond

Una descomposición importante en este trabajo, es la descomposición SVD. Esta se va a usar para desarrollar la teoría para la identificación de puntos anómalos, interpretaciones de los subespacios de información de la serie de tiempo y para obtener el vector para la identificación.

Definición 2.12 (Descomposición SVD). Sea $\mathbf{A} \in \mathbb{R}^{m \times n}$ y $r = \text{rk}(\mathbf{A})$. Se va a escribir la descomposición en valores singulares reducida dada por

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

donde $\mathbf{U} \in \mathbb{R}^{m \times r}$ y $\mathbf{V} \in \mathbb{R}^{r \times n}$ son matrices ortogonales, y $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$ es una matriz diagonal dada por $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$. Acá se utiliza la versión económica (o reducida) [17, p. 27] de la descomposición SVD con todos los $\sigma_i > 0$ para $i \in \{1, 2, \dots, r\}$. \diamond

Observación 2.13. Todas las matrices tienen una descomposición en valores singulares [17, p. 33].

Ahora, considere una matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ con valores singulares denotados por $\sigma_j(\mathbf{A})$ para $j = \{1, \dots, \min\{m, n\}\}$ y, así, la definición de rk_δ presentada en [18, p. 3].

Definición 2.14 (Rango δ de \mathbf{A}). Dado $\delta \geq 0$ y una matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\text{rk}_\delta(\mathbf{A})$ denota el entero no negativo determinado por la expresión

$$\text{rk}_\delta(\mathbf{A}) = \sum_{j=1}^{\min\{m,n\}} H_\delta(\sigma_j(\mathbf{A})) \quad (2.1)$$

donde los números $\sigma_j(\mathbf{A})$ representan los valores singulares que correspondan a una descomposición SVD de tamaño económico de \mathbf{A} . \diamond

La Def. 2.14 cumple la función de determinar el rango numérico de una matriz [22, p. 268], donde solo se suman la cantidad de valores singulares mayores a δ . La Def. 2.1 es consistente si se evalúa con $\delta = 0$ y se obtiene el rango de la matriz.

Lema 2.15. Si $\delta = 0$, entonces $\text{rk}_0(\mathbf{A}) = \text{rk}(\mathbf{A})$.

Prueba: De la Ec. 2.1, si se usa $\delta = 0$, se suma una función de activación sobre la cantidad de los valores singulares no cero de \mathbf{A} . Así, la cantidad de valores no cero de \mathbf{A} es igual al $\text{rk}(\mathbf{A})$ [17, p. 33], por lo tanto $\text{rk}_0(\mathbf{A}) = \text{rk}(\mathbf{A})$ \square .

En la Sección 3.1, se desarrolla la teoría para la identificación de puntos anómalos, que consiste en proyectar en subespacios, por lo que se define el Proyector Ortogonal.

Definición 2.16 (Proyector ortogonal \mathbf{P}). A una matriz $\mathbf{P} \in \mathbb{R}^{n \times n}$ se le llama proyector ortogonal cuando $\mathbf{P}^2 = \mathbf{P} = \mathbf{P}^T$, donde \mathbf{P}^T es la traspuesta de \mathbf{P} [17, p. 41-44]. \diamond

En el trabajo se utilizan varias series de tiempo. A continuación se escriben las notaciones para cada una de estas.

Definición 2.17 (Series de tiempo $\Sigma_\infty, \Sigma_N, \Sigma$). Para un muestreo ordenado de x_t , donde $t \in \mathbb{N}$ y $x_t \in \mathbb{R}$, se denota una serie de tiempo $\Sigma_\infty = \{x_t\}_{t \geq 1}$. Además, para el muestreo ordenado de puntos de $t = 1$ hasta N , se denota por $\Sigma_N = \{x_t\}_{t=1}^N \subset \Sigma_\infty$. También, se escribe $\Sigma \subset \Sigma_\infty$ para un muestreo de la serie de tiempo Σ_∞ y se considera de manera implícita el caso particular para $\Sigma = \{x_t\}_{t=1}^T$ para algún entero positivo T . \diamond

La manera que se va a usar la información de la serie de tiempo Σ_N de entrenamiento sin anomalías, es colocar segmentos de la serie de tiempo como columnas de matriz, por lo que se define los segmentos temporales $\mathbf{x}_L(t)$.

Definición 2.18 (Vector de segmento temporal $\mathbf{x}_L(t)$). Dado un entero positivo L y una serie de tiempo $\Sigma_\infty = \{x_t\}_{t \geq 1}$, define el vector $\mathbf{x}_L(t) \in \mathbb{R}^L$ dado por

$$\mathbf{x}_L(t) = \begin{bmatrix} x_{t-L+1} & x_{t-L+2} & x_{t-L+3} & \cdots & x_{t-1} & x_t \end{bmatrix}^T$$

para cualquier $t \geq L$. \diamond

La matriz $\mathcal{H}_L(\Sigma_N)$ que se va a usar durante el documento, está formada por los vectores segmentos temporales, al colocarlos como columnas de dicha matriz. La matriz $\mathcal{H}_L(\Sigma_N)$ sale de manera natural al plantear la evolución temporal del sistema (Sección 2.2), se puede obtener la información más importante de cada uno de los segmentos temporales para separar en subespacios (Sección 3.3) y todo el Capítulo 4.

Definición 2.19 (Matriz de trayectoria tipo Hankel $\mathcal{H}_L(\Sigma_N)$). Sea L un entero positivo y sea un muestreo ordenado $\Sigma_N = \{x_t\}_{t \geq 1}^N \subset \Sigma_\infty$. Se denota con $\mathcal{H}_L(\Sigma_N) \in \mathbb{R}^{L \times (N-L+1)}$ [18, p.3]

a la matriz de trayectoria tipo Hankel correspondiente a Σ_N , dada por la expresión

$$\begin{aligned} \mathcal{H}_L(\Sigma_N) &= \begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_{N-L} & x_{N-L+1} \\ x_2 & x_3 & x_4 & \cdots & x_{N-L+1} & x_{N-L+2} \\ x_3 & x_4 & x_5 & \cdots & x_{N-L+2} & x_{N-L+3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{L-1} & x_L & x_{L+1} & \cdots & x_{N-2} & x_{N-1} \\ x_L & x_{L+1} & x_{L+2} & \cdots & x_{N-1} & x_N \end{pmatrix} \\ &= \begin{pmatrix} | & & | & & & | \\ \mathbf{x}_L(L) & \mathbf{x}_L(L+1) & \cdots & \mathbf{x}_L(N) & & \\ | & & | & & & | \end{pmatrix} \end{aligned} \quad (2.2)$$

donde en la Ec. 2.2 se usa la Def. 2.18. \diamond

Para la detección de anomalías, idealmente se requiere un vector que pertenezca al kernel de $\mathcal{H}_L(\Sigma_N)^T$. Sin embargo, la matriz al tener datos de mediciones, puede tener un kernel no nulo, pero pueden haber vectores de norma unidad \mathbf{u} tales que $\mathcal{H}_L(\Sigma_N)^T \mathbf{u}$ sea cercano a cero. Se define el concepto de kernel aproximado de una matriz para denotar a los vectores que cumplan con esta propiedad para una matriz.

Definición 2.20 (Kernel aproximado ε de \mathbf{A}). [20, p. 2] Sea $\varepsilon > 0$. El kernel aproximado de \mathbf{A} denotado por $\ker_\varepsilon(\mathbf{A})$, se define por

$$\ker_\varepsilon(\mathbf{A}) = \{\mathbf{x} \in \mathbb{S}^{n-1} : \|\mathbf{A}\mathbf{x}\| \leq \varepsilon\} \cup \{\mathbf{0}\}.$$

\diamond

La Def. 2.20 es consistente con que todos los vectores de norma unitaria del $\ker(\mathbf{A})$ están contenidos en el $\ker_\varepsilon(\mathbf{A})$ para cualquier $\varepsilon > 0$.

Lema 2.21. Si $\mathbf{x} \in \ker(\mathbf{A})$ y $\|\mathbf{x}\| = 1$, entonces $\mathbf{x} \in \ker_\varepsilon(\mathbf{A}), \forall \varepsilon > 0$.

Prueba: De la definición de $\ker(\mathbf{A})$, si $\mathbf{x} \in \ker(\mathbf{A})$, entonces

$$\|\mathbf{A}\mathbf{x}\| = 0 < \varepsilon \Rightarrow \mathbf{x} \in \ker_\varepsilon(\mathbf{A}). \quad \square$$

Para la detección de anomalías, se necesitan vectores que pertenezcan a $\ker_\varepsilon(\mathcal{H}_L(\Sigma_N)^T)$, así que veamos cuales vectores de los cuatro subespacios fundamentales pueden ser candidatos para la detección de anomalías. Sea $\mathbf{A} \in \mathbb{R}^{m \times n}$ y $r = \text{rk}(\mathbf{A})$. Los cuatro subespacios

fundamentales de \mathbf{A} expresados en términos de los vectores \mathbf{v}_i y \mathbf{u}_i de la descomposición SVD de la descomposición no económica [22, p. 262] son

$$\begin{aligned}\text{colsp}(\mathbf{A}) &= \text{gen} \{ \mathbf{u}_1, \dots, \mathbf{u}_r \}, \\ \ker(\mathbf{A}) &= \text{gen} \{ \mathbf{v}_{r+1}, \dots, \mathbf{v}_n \}, \\ \text{colsp}(\mathbf{A}^T) &= \text{gen} \{ \mathbf{v}_1, \dots, \mathbf{v}_r \}, \\ \ker(\mathbf{A}^T) &= \text{gen} \{ \mathbf{u}_{r+1}, \dots, \mathbf{u}_m \}.\end{aligned}$$

Del Lema 2.21, se obtiene que los vectores que pertenecen a $\ker(\mathbf{A}^T)$ van a pertenecer a $\ker_\varepsilon(\mathbf{A}^T)$, por lo que cualquiera de los vectores $\mathbf{u}_{r+1}, \dots, \mathbf{u}_m$ podrían ser usados para la detección de anomalías. También se debe considerar cuáles vectores singulares izquierdos que pertenecen a $\text{colsp}(\mathbf{A})$ pertenecen a $\ker_\varepsilon(\mathbf{A}^T)$, y es una búsqueda que se requiere cuando $\ker(\mathbf{A})$ es vacío.

Se usa otra notación para las demostraciones y encontrar los otros vectores con los Teoremas 2.22 y 2.23.

Sea una matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$, $r = \text{rk}(\mathbf{A})$ y su descomposición económica SVD dada por

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

La descomposición se puede escribir [17, p. 27] relacionando cada uno de los vectores columna de \mathbf{U} y \mathbf{V} y los valores singulares de $\mathbf{\Sigma}$ de la forma

$$\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$$

para $i = \{1, \dots, r\}$.

Teorema 2.22. *Sea $\mathbf{A} \in \mathbb{R}^{m \times n}$ y $r = \text{rk}(\mathbf{A})$. Sean los vectores singulares izquierdos \mathbf{u}_i y derechos \mathbf{v}_i para $1 \leq i \leq r$, con $\mathbf{v}_i \in \text{colsp}(\mathbf{A}^T)$ y valores singulares σ_i de la descomposición SVD económica de \mathbf{A} relacionados por*

$$\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i.$$

Si $\sigma_i \leq \varepsilon$, entonces $\mathbf{v}_i \in \ker_\varepsilon(\mathbf{A})$.

Prueba: Esto se demuestra calculando la magnitud de $\mathbf{A}\mathbf{v}_i$. Usamos que en la descomposición SVD, los vectores \mathbf{u}_i están normalizados. Entonces

$$\begin{aligned}\|\mathbf{A}\mathbf{v}_i\| &= \|\sigma_i\mathbf{u}_i\| \\ &= \sigma_i\|\mathbf{u}_i\| \\ &= \sigma_i.\end{aligned}$$

Ya que $\sigma_i \leq \varepsilon$ por hipótesis, se cumple que $\|\mathbf{A}\mathbf{v}_i\| = \sigma_i \leq \varepsilon$. Por lo tanto $\mathbf{v}_i \in \ker_\varepsilon(\mathbf{A})$ para todos los \mathbf{v}_i con $\sigma_i \leq \varepsilon$. \square

Para la detección de anomalías, se debe obtener vectores que pertenezcan a $\ker_\varepsilon(\mathcal{H}_L(\Sigma_N)^T)$, y no a $\ker_\varepsilon(\mathcal{H}_L(\Sigma_N))$. Con el Teorema 2.22, se obtienen vectores que pertenecen a $\ker_\varepsilon(\mathbf{A})$. Además, en el Teorema 2.23, se obtiene una relación para obtener los vectores que pertenecen a $\ker_\varepsilon(\mathbf{A}^T)$ con la información de $\ker_\varepsilon(\mathbf{A})$ y la descomposición SVD de \mathbf{A} .

Teorema 2.23. Sea $\mathbf{A} \in \mathbb{R}^{m \times n}$ y sea los vectores singulares izquierdos \mathbf{u}_i y derechos \mathbf{v}_i relacionados por

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i,$$

para $1 \leq i \leq r$, donde se consideran solo los $\mathbf{v}_i \in \text{colsp}(\mathbf{A}^T)$ (para que $\sigma_i > 0$). Si $\mathbf{v}_i \in \ker_\varepsilon(\mathbf{A})$, entonces $\mathbf{u}_i \in \ker_\varepsilon(\mathbf{A}^T)$

Prueba: Considere la descomposición SVD de \mathbf{A}

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

Aplicando la transpuesta, se obtiene

$$\begin{aligned}\mathbf{A}^T &= (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^T \\ &= \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T.\end{aligned}\tag{2.3}$$

Así, la descomposición SVD de \mathbf{A}^T es $\mathbf{A}^T = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T$. Usando la notación de la Ec. 2.1, la descomposición se puede escribir como

$$\mathbf{A}^T\mathbf{u}_i = \sigma_i\mathbf{v}_i.\tag{2.4}$$

Los vectores izquierdos de la matriz \mathbf{A} son los vectores derechos de \mathbf{A}^T , con los mismos valores singulares respectivos. Se usa el mismo argumento del Teorema 2.22 con la Ec. 2.4. Si $\sigma_i \leq \varepsilon$, entonces $\mathbf{u}_i \in \ker_\varepsilon(\mathbf{A}^T)$. \square

2.2. Construcción de la matriz de Hankel

Considere una señal $x(t)$ de un sistema de interés. Cuando se mide la señal $x(t)$ y se almacena en memoria, no se guarda toda la señal continua, si no, puntos discretos puntos de cuando se realiza cada medición. Esto se ejemplifica en la Fig. 2.1, donde la línea continua representa la señal del sistema, y los círculos son los puntos medidos.

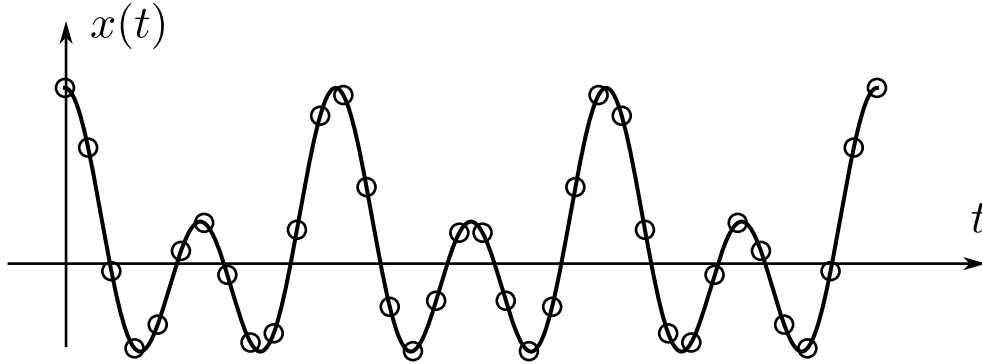


Figura 2.1. Señal a analizar y los puntos medidos.

Suponga que se miden N puntos de la señal en comportamiento regular: $x(t_1), x(t_2), \dots, x(t_N)$. Para efectos de notación, los puntos de la señal medidos $x(t_i)$ se escriben como $x_i = x(t_i)$. También suponga que el tiempo entre mediciones consecutivas $t_{j+1} - t_j$ es una constante $\forall j \geq 1$. Sea $\Delta t = t_{j+1} - t_j$ dicha constante.

Suponga además que las señales que se quieren analizar son de comportamiento periódico. Así, para una señal periódica, se podría esperar que sea posible escribir el punto x_{t+1} aproximadamente como una combinación lineal de los puntos pasados.

Si se usan $M \in \mathbb{N}$ puntos para escribir x_{M+1} , se expresa por

$$x_{M+1} = c_1 x_1 + c_2 x_2 + c_3 x_3 + \dots + c_{M-1} x_{M-1} + c_M x_M.$$

Note que se puede hacer esto para el punto x_{M+2} . Se usa el mismo vector de coeficientes

$$\mathbf{c} = (c_1, \dots, c_M)^T \quad (2.5)$$

y se obtiene que

$$x_{M+2} = c_1 x_2 + c_2 x_3 + c_3 x_4 \dots + c_{M-1} x_M + c_M x_{M+1}.$$

Continuando este proceso, se obtiene el sistema de ecuaciones

$$\begin{cases} c_1x_1 + c_2x_2 + c_3x_3 + \cdots + c_{M-1}x_{M-1} + c_Mx_M = x_{M+1} \\ c_1x_2 + c_2x_3 + c_3x_4 + \cdots + c_{M-1}x_M + c_Mx_{M+1} = x_{M+2} \\ c_1x_3 + c_2x_4 + c_3x_5 + \cdots + c_{M-1}x_{M+1} + c_Mx_{M+2} = x_{M+3} \\ \vdots + \vdots + \vdots + \cdots + \vdots + \vdots = \vdots \\ c_1x_{L-1} + c_2x_L + c_3x_{L+1} + \cdots + c_{M-1}x_{M+L-3} + c_Mx_{M+L-2} = x_{M+L-1} \\ c_1x_L + c_2x_{L+1} + c_3x_{L+2} + \cdots + c_{M-1}x_{M+L-2} + c_Mx_{M+L-1} = x_{M+L} \end{cases}$$

que al escribirlo de forma matricial obtenemos

$$\begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_{M-1} & x_M \\ x_2 & x_3 & x_4 & \cdots & x_M & x_{M+1} \\ x_3 & x_4 & x_5 & \cdots & x_{M+1} & x_{M+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{L-1} & x_L & x_{L+1} & \cdots & x_{M+L-3} & x_{M+L-2} \\ x_L & x_{L+1} & x_{L+2} & \cdots & x_{M+L-2} & x_{M+L-1} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{M-1} \\ c_M \end{pmatrix} = \begin{pmatrix} x_{M+1} \\ x_{M+2} \\ x_{M+3} \\ \vdots \\ x_{M+L-1} \\ x_{M+L} \end{pmatrix} \quad (2.6)$$

. La estructura de la matriz en la Ec. 2.6 es una matriz de Hankel [10, p. 688].

Usualmente hace referencia a que se tienen N puntos de entrenamiento para entrenar un modelo y construir un vector \mathbf{p} que se va a usar para la detección. Se quiere reescribir el último punto x_{L+M-1} como x_N . Despejando para M , se obtiene

$$\begin{aligned} N &= M + L - 1 \\ \Rightarrow M &= N - L + 1. \end{aligned} \quad (2.7)$$

Cambiando los términos con la Ec. 2.7 para que dependan de N y no M en la Ec. 2.6 se obtiene

$$\begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_{N-L} & x_{N-L+1} \\ x_2 & x_3 & x_4 & \cdots & x_{N-L+1} & x_{N-L+2} \\ x_3 & x_4 & x_5 & \cdots & x_{N-L+2} & x_{N-L+3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{L-1} & x_L & x_{L+1} & \cdots & x_{N-2} & x_{N-1} \\ x_L & x_{L+1} & x_{L+2} & \cdots & x_{N-1} & x_N \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{N-L} \\ c_{N-L+1} \end{pmatrix} = \begin{pmatrix} x_{N-L+2} \\ x_{N-L+3} \\ x_{N-L+4} \\ \vdots \\ x_N \\ x_{N+1} \end{pmatrix}. \quad (2.8)$$

Para uso del resto del documento, usando las Def. 2.18 y 2.19, se simplifican las notaciones. La Ec. 2.8 queda reescrita de la forma

$$\mathcal{H}_L(\Sigma_N)\mathbf{c} = \mathbf{x}_L(N+1). \quad (2.9)$$

Las series de tiempo a analizar para detectar las anomalías, son series donde la evolución temporal del sistema se puede modelar con una matriz de trayectoria tipo Hankel, como se muestra en la Ec. 2.9. Ejemplos de este modelaje matemático se puede usar, por ejemplo, para encontrar los coeficientes en sistemas dispersos [18] o también para obtener la contribución lineal de una señal eventualmente periódica [19].

Observación 2.24. Se menciona que tenemos N puntos para entrenamiento para encontrar el vector para la detección de anomalías y planteamos el sistema de la Ec. 2.8 usando un punto extra x_{N+1} . La solución al sistema no es tan importante, ya que la información necesaria para generar el vector para la detección de anomalías se encuentra en las columnas de la matriz $\mathcal{H}_L(\Sigma_N)$, que contiene todos los puntos disponibles para el entrenamiento.

Observación 2.25. El planteamiento inicial es para sistemas periódicos ya que hay una gran cantidad de sistemas con este comportamiento, pero en el planteamiento final, y la teoría que se describe en el Capítulo 3, solo es necesario que la evolución del sistema se pueda describir con una matriz de trayectoria tipo Hankel de la Ec. 2.9, por lo que la hipótesis que tenga que ser periódica no es necesaria.

2.3. Construcción alternativa de la matriz de Hankel

Otra manera de representar la construcción de la matriz de Hankel, en lugar de obtenerlo con los sistemas de ecuaciones, es enfocarse en las columnas. Una manera visual de representar a los vectores $\mathbf{x}_L(t)$ y la matriz $\mathcal{H}_L(\Sigma_N)$ se muestra en la Fig. 2.2. Suponga que tenemos N puntos de la señal y se escogen L puntos para formar los vectores $\mathbf{x}_L(t)$. En la Fig. 2.2 se muestra para $N = 24$ y $L = 7$.

Ahora se va a construir la matriz Hankel de otra manera. Para esto usamos segmentos desde el punto de la serie x_L hasta L elementos hacia atrás usando x_1 con

$$\mathbf{x}_L(L) = \begin{bmatrix} x_1 & x_2 & \cdots & x_{L-1} & x_L \end{bmatrix}^T.$$

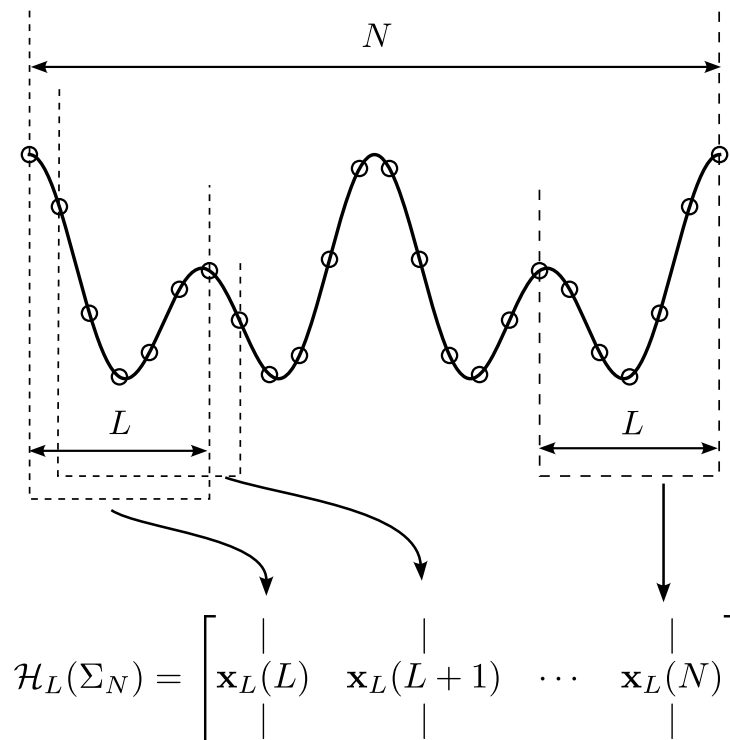


Figura 2.2. Construcción de la matriz de Hankel.

Considere este primer segmento $\mathbf{x}_L(L)$ y posicione este primer vector en la primera columna de $\mathcal{H}_L(\Sigma_N)$. Luego tomando x_{L+1} junto con L elementos hacia atrás hasta x_2

$$\mathbf{x}_L(L+1) = [x_2 \ x_3 \ \cdots \ x_L \ x_{L+1}]^T,$$

se coloca a $\mathbf{x}_L(L+1)$ en la segunda columna de $\mathcal{H}_L(\Sigma_N)$, y así, se continúa hasta el último elemento x_N con

$$\mathbf{x}_L(N) = [x_{N-L+1} \ x_{N-L+2} \ \cdots \ x_{N-1} \ x_N]^T.$$

Se posiciona a $\mathbf{x}_L(N)$ en la última columna de $\mathcal{H}_L(\Sigma_N)$. Se termina con una matriz donde cada una de sus columnas son segmentos de la serie de tiempo. Debido a la forma en que construye la matriz, esta tendrá L filas y $N - L + 1$ columnas.

La manera de construir la matriz de Hankel con la interpretación que las columnas son los segmentos temporales $\mathbf{x}_L(t)$ de toda la señal de entrenamiento es la manera preferencial que vamos a usar para obtener los resultados teóricos, ya que la identificación de los puntos anómalos del algoritmo desarrollado en este trabajo, se basa en la información de las columnas de la matriz $\mathcal{H}_L(\Sigma_N)$.

Capítulo 3

Detección proyectiva de anomalías para series de tiempo

En este capítulo se desarrolla la teoría para la detección proyectiva de anomalías para series de tiempo, basadas en subespacios con las definiciones y conceptos presentados en el Capítulo 2.

3.1. Teoría de subespacios para identificación de anomalías

Se define el L -normalidad de Σ , denotado por $\mathcal{N}_L(\Sigma)$ que corresponde a los segmentos temporales $\mathbf{x}_L(t)$ que no son anómalos.

Definición 3.1 (Región L -normalidad de Σ , $\mathcal{N}_L(\Sigma)$). Dado un entero positivo L y un muestreo de una serie de tiempo $\Sigma = \{x_t\}_{t=1}^T \subset \Sigma_\infty$ que corresponde a los primeros $T \geq L + 1$ elementos de $\Sigma_\infty = \{x_t\}_{t \geq 1}$, se denota con $\mathcal{N}_L(\Sigma)$ al conjunto de todos los $\mathbf{x}_L(t) \in \text{colsp}(\mathcal{H}_L(\Sigma))$. Al conjunto $\mathcal{N}_L(\Sigma)$ se le llama la región de L -normalidad de Σ . \diamond

A continuación, se define el siguiente concepto para la detección de anomalías [20, p. 2], usando la Def. 3.1.

Definición 3.2 (PAD). Dado $\varepsilon \geq 0$, un entero positivo $L > 0$ y un muestreo de una serie de tiempo $\Sigma \subset \Sigma_\infty$ que corresponde a los primeros $T \geq L + 1$ elementos de la serie de tiempo $\Sigma_\infty = \{x_t\}_{t \geq 1}$. Un Detector de Anomalías Proyectivo (L, ε) [(L, ε) -PAD] basado en Σ para Σ_∞ es determinado por un proyector \mathbf{P} tal que

$$\|\mathbf{P}\mathbf{x}_L(t) - \mathbf{x}_L(t)\| \leq \varepsilon \quad (3.1)$$

cuando $\mathbf{x}_L(t)$ no tiene un valor anómalo, con respecto a los datos de entrenamiento determinados por las columnas de $\mathcal{H}_L(\Sigma)$, dígame, cuando $\mathbf{x}_L(t) \in \mathcal{N}_L(\Sigma)$. \diamond

Observación 3.3. La notación (L, ε) -PAD de la Definición 3.2 viene de sus siglas en inglés Projective Anomaly Detector.

Basado en la Def.3.2 y la Def. 2.20, se puede obtener una cota como la de la Ec. 3.1 usando los primeros vectores singulares izquierdos asociados a los valores singulares más grandes de la matriz $\mathcal{H}_L(\Sigma_N)$ con el Teorema 3.5 y Corolario 3.7. Se usan los primeros vectores singulares izquierdos para obtener la información más importante de la serie de tiempo con las columnas de la matriz $\mathcal{H}_L(\Sigma_N)$ que contienen los $\mathbf{x}_L(t)$ de entrenamiento. La cantidad de vectores singulares izquierdos a usar depende de la escogencia en la cota de la Ec. 3.2. Se usa un resultado del procedimiento de la prueba del Teorema 3.4 del artículo [18] para probar el Teorema 3.5

Teorema 3.4 (Vides). *Para $\delta > 0$, una matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ y los vectores singulares izquierdos \mathbf{u}_i de la matriz \mathbf{A} , \mathbf{Q} un proyector ortogonal de rango $r = rk_\delta(\mathbf{A})$ determinado por la expresión $\mathbf{Q} = \sum_{i=1}^r \mathbf{u}_i \mathbf{u}_i^T$ y $s_r = \sqrt{r (\text{mín} \{L, M\} - r)}$, se cumple*

$$\|\mathbf{A} - \mathbf{QA}\|_F \leq \frac{s_r}{\sqrt{r}} \delta.$$

Teorema 3.5. *Dado un valor $L > 0$ y un muestreo de una serie de tiempo $\Sigma_N \subset \Sigma_\infty$ que corresponde a una trayectoria tipo Hankel $\mathcal{H}_L(\Sigma_N)$. Para cualquier $\sigma_1(\mathcal{H}_L(\Sigma_N)) \geq \varepsilon \geq \sigma_L(\mathcal{H}_L(\Sigma_N))$ existe un proyector \mathbf{P} que satisface la condición*

$$\|\mathbf{P}\mathbf{x}_L(t) - \mathbf{x}_L(t)\| \leq K_L(t)\varepsilon \quad (3.2)$$

para cada $\mathbf{x}_L(t) \in \mathcal{N}_L$ y para algún número $K_L(t)$ que no depende explícitamente de ε , para $t \geq L$.

Prueba: Sea M el número de columnas de $\mathcal{H}_L(\Sigma_N)$. Para probar la existencia del proyector \mathbf{P} que satisface la Ec. 3.2, considere un vector $\mathbf{x}_L(t)$ determinado por los elementos de la serie de tiempo $\Sigma_\infty = \{x_t\}_{t \geq 1}$ que contiene a la serie de tiempo Σ_N , tal que $\mathbf{x}_L(t) \in \text{colsp}(\mathcal{H}_L(\Sigma_N))$. Por lo tanto, existe una solución $\mathbf{c}(t) = \begin{bmatrix} c_1(t) & c_2(t) & \cdots & c_M(t) \end{bmatrix}^T$ para el sistema

$$\mathcal{H}_L(\Sigma_N)\mathbf{c}(t) = \mathbf{x}_L(t). \quad (3.3)$$

Ahora, se realiza la descomposición SVD para $\mathcal{H}_L(\Sigma_N)$ con

$$\begin{aligned}\mathcal{H}_L(\Sigma_N) &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \\ &= \begin{bmatrix} | & | & \cdots & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_L \\ | & | & & | \end{bmatrix} \mathbf{\Sigma}\mathbf{V}^T.\end{aligned}$$

Sea $r = \text{rk}_\varepsilon(\mathcal{H}_L(\Sigma_N))$ y considere los primeros r vectores singulares izquierdos $\mathbf{u}_1, \dots, \mathbf{u}_r$ de $\mathcal{H}_L(\Sigma_N)$. Con todos los vectores singulares izquierdos, se tiene que

$$\text{span}(\{\mathbf{u}_1, \dots, \mathbf{u}_L\}) = \text{colsp}(\mathcal{H}_L(\Sigma_N)),$$

por hipótesis, se puede escoger un ε tal que

$$\sigma_r(\mathcal{H}_L(\Sigma_N)) \geq \varepsilon \geq \sigma_{r+1}(\mathcal{H}_L(\Sigma_N))$$

Usando el Teorema 3.4 [18] y definiendo las variables

$$\begin{aligned}s_r &= \sqrt{r(\text{mín}\{L, M\} - r)} \\ \mathbf{U}_r &= \begin{bmatrix} | & | & \cdots & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \\ | & | & & | \end{bmatrix}\end{aligned}$$

y el proyector ortogonal

$$\mathbf{P}_r = \mathbf{U}_r \mathbf{U}_r^T \tag{3.4}$$

se obtiene que

$$\|\mathcal{H}_L(\Sigma_N) - \mathbf{P}_r \mathcal{H}_L(\Sigma_N)\|_F \leq \frac{s_r}{\sqrt{r}} \varepsilon. \tag{3.5}$$

Si se combinan la Ec. 3.3 y la Ec. 3.5, se obtiene la cota

$$\begin{aligned}\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\| &\leq \|\mathcal{H}_L(\Sigma_N) - \mathbf{P}_r \mathcal{H}_L(\Sigma_N)\|_F \|\mathbf{c}(t)\| \\ &\leq \frac{s_r}{\sqrt{r}} \|\mathbf{c}(t)\| \varepsilon.\end{aligned} \tag{3.6}$$

Ahora, tomando

$$K_L(t) = \frac{s_r}{\sqrt{r}} \|\mathbf{c}(t)\|$$

y substituyendo en la desigualdad 3.6, se obtiene

$$\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\| \leq K_L(t) \varepsilon. \quad \square$$

Definición 3.6 (Cota L -relativa). Dado un entero positivo L y un muestreo de una serie de tiempo $\Sigma \subset \Sigma_\infty$, para cada $t \geq L$, los números $K_L(t)$ determinados por

$$K_L(t) = \frac{s_r}{\sqrt{r}} \|\mathbf{c}(t)\|$$

se llaman las cotas L -relativas para la serie de tiempo Σ_∞ con respecto al muestreo Σ . \diamond

Para simplificar el análisis, se puede obtener una cota que apliquen para todos los t y usarla en la Ec. 3.2.

Corolario 3.7. *Dados N, L enteros positivos, con $N > L$ y una serie de tiempo $\Sigma \subset \Sigma_\infty$ con cotas L -relativas $K_L(t)$, para $t \geq L$, si definimos*

$$\hat{K} = \sup_{L \leq t \leq N} K_L(t) \quad (3.7)$$

entonces para un $\varepsilon \geq 0$ que satisface $\sigma_1(\mathcal{H}_L(\Sigma_N)) \geq \varepsilon \geq \sigma_L(\mathcal{H}_L(\Sigma_N))$, existe un $(L, \hat{K}\varepsilon)$ -PAD \mathbf{P}_r basado en Σ para Σ_∞ , para $L \leq t \leq N$.

Prueba: Usando el Teorema 3.5 y los valores de ε , se puede construir un proyector \mathbf{P}_r usando la Ec. 3.4 tal que

$$\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\| \leq K_L(t)\varepsilon.$$

Acotando con el supremo \hat{K} (Ec. 3.7), se obtiene

$$\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\| \leq K_L(t)\varepsilon \leq \hat{K}\varepsilon$$

para cada $\mathbf{x}_L(t) \in \mathcal{N}(\Sigma)$, para cada $L \leq t \leq N$. \square

3.2. Preliminares de la heurística del detector de anomalías

En las secciones pasadas, se expone la teoría para clasificar un segmento $\mathbf{x}_L(t)$, que tiene una anomalía o no en base al PAD. En esta sección se presenta una idea más conceptual del funcionamiento del algoritmo. Para explicar la noción de manera más visual e intuitiva, se presentan primero dos heurísticas: la primera del cómo se distribuye la información en una descomposición SVD en los vectores izquierdos y la segunda de proyectores ortogonales y como se proyecta la información en los subespacios respectivos. Se combinan estas dos ideas para dar intuición de lo que realiza el PAD para detectar las anomalías.

3.2.1. Primera preliminar de la heurística

Una técnica usual para extraer la información más importante de una matriz, es usar la descomposición SVD y truncar la descomposición. Sea $\mathbf{X} \in \mathbb{R}^{m \times n}$ de rango r . La descomposición SVD económica es

$$\mathbf{X} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \quad (3.8)$$

con el orden usual de los valores singulares $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$. Cada uno de los términos $\sigma_i \mathbf{u}_i \mathbf{v}_i^T$ en la Ec. 3.8 es una matriz de rango 1 y cada término $\sigma_i \mathbf{u}_i \mathbf{v}_i^T$ captura menos información de la matriz \mathbf{X} conforme el valor singular σ_i se hace más pequeño [5, p.8]. Los términos que tiene la mayor información de \mathbf{X} son los primeros términos de la sumatoria de la Ec. 3.8.

Así, se puede extraer la información más importante de \mathbf{X} conservando los primeros k términos de la sumatoria de la Ec. 3.8, donde $k < r$. Sea \mathbf{X}_k esta suma, dígase

$$\mathbf{X}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^T. \quad (3.9)$$

Esta matriz \mathbf{X}_k es la matriz de rango k que mejor aproxima a \mathbf{X} , en el aspecto de minimizar la norma $\|\mathbf{X} - \mathbf{B}\|_F$ para cualquier matriz \mathbf{B} de rango k . A este resultado se le conoce como el Teorema de Eckart-Young.

Teorema 3.8 (Eckart-Young). [5, p. 9] *La aproximación óptima de $\mathbf{X} \in \mathbb{R}^{m \times n}$ de rango k , en el sentido de mínimos cuadrados, está dado por la descomposición SVD truncada \mathbf{X}_k (Ec. 3.9):*

$$\underbrace{\operatorname{argmin}}_{\mathbf{B}, \text{ t.q. } \operatorname{rank}(\mathbf{B})=k} \|\mathbf{X} - \mathbf{B}\|_F = \mathbf{X}_k$$

donde $\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |X_{ij}|^2}$ es la norma de Frobenius.

Para obtener la información más importante de \mathbf{X} en una matriz de rango k , se usa \mathbf{X}_k definido en la Ec. 3.9. Con esta aproximación para \mathbf{X} , veamos la información que se rescata del espacio de columnas de \mathbf{X}_k . De la Ec. 3.9, si se multiplica a \mathbf{X}_k por un vector $\mathbf{c} \in \mathbb{R}^n$,

obtenemos

$$\begin{aligned}\mathbf{X}_k \mathbf{c} &= \left(\sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^T \right) \mathbf{c} \\ &= \sum_{i=1}^k \sigma_i \mathbf{u}_i \left(\mathbf{v}_i^T \mathbf{c} \right).\end{aligned}$$

Se cambia el término $d_i = \sigma_i (\mathbf{v}_i^T \mathbf{c})$ para mayor claridad y se obtiene

$$\mathbf{X}_k \mathbf{c} = \sum_{i=1}^k d_i \mathbf{u}_i. \quad (3.10)$$

De la Ec. 3.10 se muestra que el espacio de columnas de \mathbf{X}_k está generado por la combinación lineal de los primeros k vectores singulares izquierdos \mathbf{u}_i de la matriz original \mathbf{X} , dígase $\text{gen} \{\mathbf{u}_1, \dots, \mathbf{u}_k\}$.

En resumen, si se quiere obtener la información más importante del espacio de columnas de \mathbf{X} , se usan los primeros $\mathbf{u}_1, \dots, \mathbf{u}_k$ vectores singulares izquierdos, donde el espacio de información importante de columnas es $\text{gen} \{\mathbf{u}_1, \dots, \mathbf{u}_k\}$.

3.2.2. Segunda preliminar de la heurística

La segunda idea para entender la heurística, es proyectar vectores sobre diferentes subespacios y obtener las magnitudes en base a las proyecciones.

Sean $\mathbf{u}_1, \dots, \mathbf{u}_L \in \mathbb{R}^L$, L vectores unitarios ortogonales. Estos vectores forman una base para \mathbb{R}^L . Sea el subespacio vectorial $\mathcal{V} \subset \mathbb{R}^L$, de dimension $r < L$, donde \mathcal{V} es generado por los primeros r vectores \mathbf{u}_i , dígase, $\mathcal{V} = \text{gen} \{\mathbf{u}_1, \dots, \mathbf{u}_r\}$, y el complemento ortogonal \mathcal{V}^\perp obtenido por los otros vectores $\mathbf{u}_{r+1}, \dots, \mathbf{u}_L$, con $\mathcal{V}^\perp = \text{gen} \{\mathbf{u}_{r+1}, \dots, \mathbf{u}_L\}$. El proyector ortogonal para el subespacio \mathcal{V} lo detonamos por \mathbf{P}_r , está dado por

$$\mathbf{P}_r = \mathbf{U}_r \mathbf{U}_r^T \quad (3.11)$$

donde

$$\mathbf{U}_r = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \\ | & | & & | \end{bmatrix}$$

y el proyector ortogonal para \mathcal{V}^\perp se denota con \mathbf{Q}_r , que está dado por

$$\mathbf{Q}_r = \mathbf{U}'_r (\mathbf{U}'_r)^T \quad (3.12)$$

donde

$$\mathbf{U}'_r = \begin{bmatrix} | & | & & | \\ \mathbf{u}_{r+1} & \mathbf{u}_{r+2} & \cdots & \mathbf{u}_L \\ | & | & & | \end{bmatrix}.$$

Se pueden escribir a \mathbf{P}_r y \mathbf{Q}_r como una suma de los vectores \mathbf{u}_i usando la multiplicación de matrices por bloques [12, p. 105], donde cada bloque son los vectores \mathbf{u}_i . Para \mathbf{P}_r se obtiene

$$\begin{aligned} \mathbf{P}_r &= \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \\ | & | & & | \end{bmatrix} \cdot \begin{bmatrix} - & \mathbf{u}_1^T & - \\ - & \mathbf{u}_2^T & - \\ \vdots & & \\ - & \mathbf{u}_r^T & - \end{bmatrix} \\ &= \sum_{i=1}^r \mathbf{u}_i \mathbf{u}_i^T. \end{aligned} \quad (3.13)$$

Se realiza un procedimiento similar para \mathbf{Q}_r , se obtiene

$$\begin{aligned} \mathbf{Q}_r &= \begin{bmatrix} | & | & & | \\ \mathbf{u}_{r+1} & \mathbf{u}_{r+2} & \cdots & \mathbf{u}_L \\ | & | & & | \end{bmatrix} \cdot \begin{bmatrix} - & \mathbf{u}_{r+1}^T & - \\ - & \mathbf{u}_{r+2}^T & - \\ \vdots & & \\ - & \mathbf{u}_L^T & - \end{bmatrix} \\ &= \sum_{i=r+1}^L \mathbf{u}_i \mathbf{u}_i^T. \end{aligned} \quad (3.14)$$

Como los vectores $\mathbf{u}_1, \dots, \mathbf{u}_L$ forman una base para \mathbb{R}^L , se puede escribir cualquier vector de \mathbb{R}^L como una combinación lineal de los vectores $\mathbf{u}_1, \dots, \mathbf{u}_L$. Un vector arbitrario $\mathbf{a} \in \mathbb{R}^L$ escrito en esta base, aprovechando que los vectores \mathbf{u}_i son ortogonales, es de la forma

$$\mathbf{a} = \sum_{i=1}^L a_i \mathbf{u}_i = \sum_{i=1}^L (\mathbf{u}_i^T \mathbf{a}) \mathbf{u}_i.$$

Ahora, se puede separar la sumatoria para expresar al vector \mathbf{a} con los proyectores \mathbf{P}_r y \mathbf{Q}_r .

usando las Ec.3.13 y 3.14 y se obtiene

$$\begin{aligned}
 \mathbf{a} &= \sum_{i=1}^L (\mathbf{u}_i^T \mathbf{a}) \mathbf{u}_i \\
 &= \sum_{i=1}^r (\mathbf{u}_i^T \mathbf{a}) \mathbf{u}_i + \sum_{i=r+1}^L (\mathbf{u}_i^T \mathbf{a}) \mathbf{u}_i \\
 &= \sum_{i=1}^r \mathbf{u}_i (\mathbf{u}_i^T \mathbf{a}) + \sum_{i=r+1}^L \mathbf{u}_i (\mathbf{u}_i^T \mathbf{a}) \\
 &= \left(\sum_{i=1}^r \mathbf{u}_i \mathbf{u}_i^T \right) \mathbf{a} + \left(\sum_{i=r+1}^L \mathbf{u}_i \mathbf{u}_i^T \right) \mathbf{a} \\
 &= \mathbf{P}_r \mathbf{a} + \mathbf{Q}_r \mathbf{a}.
 \end{aligned}$$

Por lo tanto, el vector \mathbf{a} se puede escribir de la forma

$$\mathbf{a} = \mathbf{P}_r \mathbf{a} + \mathbf{Q}_r \mathbf{a}, \quad (3.15)$$

donde $\mathbf{P}_r \mathbf{a}$ es la proyección sobre \mathcal{V} y $\mathbf{Q}_r \mathbf{a}$ es la proyección sobre \mathcal{V}^\perp . Esta descomposición se muestra de manera gráfica en la Fig. 3.1. Escribiendo $\mathbf{P}_r \mathbf{a}$ y $\mathbf{Q}_r \mathbf{a}$ en componentes, son

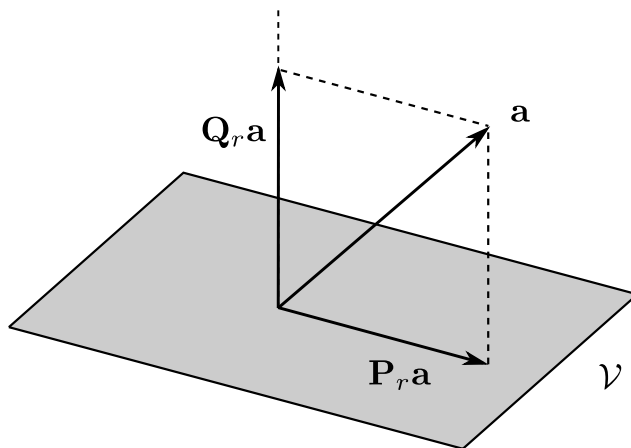


Figura 3.1. Proyección de \mathbf{a} sobre los subespacios \mathcal{V} y \mathcal{V}^\perp usando los proyectores \mathbf{P}_r y \mathbf{Q}_r .

de la forma

$$\begin{aligned}
 \mathbf{P}_r \mathbf{a} &= \sum_{i=1}^r a_i \mathbf{u}_i, \\
 \mathbf{Q}_r \mathbf{a} &= \sum_{i=r+1}^L a_i \mathbf{u}_i,
 \end{aligned}$$

junto con las magnitudes cuadradas

$$\begin{aligned}\|\mathbf{P}_r \mathbf{a}\|^2 &= \sum_{i=1}^r a_i^2, \\ \|\mathbf{Q}_r \mathbf{a}\|^2 &= \sum_{i=r+1}^L a_i^2.\end{aligned}$$

Reescribiendo la Ec. 3.15, se obtiene

$$\mathbf{Q}_r \mathbf{a} = \mathbf{a} - \mathbf{P}_r \mathbf{a}. \quad (3.16)$$

En la detección de anomalías, se va a tomar en cuenta el cálculo de $\|\mathbf{a} - \mathbf{P}_r \mathbf{a}\|$, que por la Ec. 3.16, también se puede enfocar en calcular $\|\mathbf{Q}_r \mathbf{a}\|$.

3.3. Heurística del PAD

Con ambas ideas presentadas para extraer información importante de una matriz y proyectar la información de un vector en subespacios, se presenta la heurística para el PAD.

La idea es la siguiente: suponga que se tiene un segmento temporal $\mathbf{x}_L(t)$ para cierto t y quiere saber si este segmento es anómalo.

Una manera de identificar si el segmento $\mathbf{x}_L(t)$ es anómalo, es compararlo con otros varios segmentos $\mathbf{x}_L(t')$ para varios t' sin compartamiento anómalo. Si el segmento $\mathbf{x}_L(t)$ se parece a alguno de los elementos no anómalos $\mathbf{x}_L(t')$, se puede decir que $\mathbf{x}_L(t)$ es no anómalo. Caso contrario, si parece a ninguno de los $\mathbf{x}_L(t')$, se concluye que es anómalo.

Suponga que se tienen N puntos no anómalos de una serie de tiempo Σ_N , y se usa L para el tamaño de cada segmento $\mathbf{x}_L(t')$. Se colocan cada uno de estos $\mathbf{x}_L(t')$ como la columna de una matriz que se muestra en la Ec. 3.17 y se obtiene

$$\mathcal{H}_L(\Sigma_N) = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{x}_L(L) & \mathbf{x}_L(L+1) & \cdots & \mathbf{x}_L(N) \\ | & | & & | \end{bmatrix}. \quad (3.17)$$

Esta va a ser la matriz de Hankel $\mathcal{H}_L(\Sigma_N)$, usando la construcción de esta matriz en la Sección 2.3. Debido a la forma en que construimos la matriz, se tiene $M = N - L + 1$ (Ec. 2.7).

Para identificar a $\mathbf{x}_L(t)$ como anómalo o no, se compara el segmento $\mathbf{x}_L(t)$ con todas las M columnas de la matriz de la Ec. 3.17. Sin embargo, a pesar de ser un método efectivo, si se

tienen varias columnas M con las cuales comparar, podría ser lento si M es bastante grande y sería más eficiente tener un método en donde se tengan que realizar menos comparaciones.

Para evitar comparar con todas las columnas de $\mathcal{H}_L(\Sigma_N)$ y poder usar un método más rápido, se puede usar la descomposición SVD de $\mathcal{H}_L(\Sigma_N)$, para obtener la información más importante de las columnas. La descomposición SVD de $\mathcal{H}_L(\Sigma_N)$ es

$$\begin{bmatrix} | & | & \cdots & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_L \\ | & | & \cdots & | \end{bmatrix} \boldsymbol{\Sigma} \mathbf{V}^T = \mathcal{H}_L(\Sigma_N).$$

Basado en la Subsección 3.2.1, se puede aproximar a $\mathcal{H}_L(\Sigma_N)$ con una matriz de rango $r < L$ truncando la descomposición SVD tomando las primeras r matrices $\sigma_i \mathbf{u}_i \mathbf{v}_i^T$ de rango 1, que son las matrices que contienen la mayor cantidad de información de $\mathcal{H}_L(\Sigma_N)$. Ya que nos interesa comparar $\mathbf{x}_L(t)$ con las columnas de 3.17, basándonos en la Ec. 3.10, la información más importante del espacio de columnas está contenida en los primeros r vectores singular izquierdos \mathbf{u}_i de $\mathcal{H}_L(\Sigma_N)$.

Sean los subespacios $\mathcal{V} = \text{gen}\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ y $\mathcal{V}^\perp = \text{gen}\{\mathbf{u}_{r+1}, \dots, \mathbf{u}_L\}$, junto con los proyectores \mathbf{P}_r y \mathbf{Q}_r dados por las Ec. 3.11 y 3.12 para proyectar sobre el subespacio respectivo. El subespacio \mathcal{V} es generado por los primeros vectores singulares izquierdos, que es la información más importante del espacio de columnas de $\mathcal{H}_L(\Sigma_N)$. En lugar de hacer una comparación con los vectores columna de $\mathcal{H}_L(\Sigma_N)$, se resume la información importante de las columnas que generan el subespacio \mathcal{V} . A su vez, ya que \mathcal{V}^\perp está formado por los últimos vectores singulares izquierdos $\mathbf{u}_{r+1}, \dots, \mathbf{u}_L$, este subespacio contiene información no característica de la señales no anómalas del espacio de columnas de $\mathcal{H}_L(\Sigma_N)$.

El comportamiento anómalo va a estar en el subespacio \mathcal{V}^\perp , así que se proyecta el vector $\mathbf{x}_L(t)$ sobre \mathcal{V}^\perp usando $\mathbf{Q}_r \mathbf{x}_L(t)$. Se mide la proyección obteniendo la magnitud de $\|\mathbf{Q}_r \mathbf{x}_L(t)\|$. Por la Ec. 3.16, se tiene que

$$\|\mathbf{Q}_r \mathbf{x}_L(t)\| = \|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\|. \quad (3.18)$$

La magnitud $\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\|$ en la Ec. 3.18 es la cantidad de la Def. 3.2. Cuando $\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\| \leq \varepsilon$ para algún ε , se dice que pertenece a la región de normalidad. Caso contrario, si $\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\| > \varepsilon$, se puede interpretar que hay bastante información en el subespacio \mathcal{V}^\perp de información no característica de la señal para clasificarlo como anomalía. Con el Teorema 3.5 y Colorario 3.7, el proyector se construye con los primeros vectores singulares izquierdos de $\mathcal{H}_L(\Sigma_N)$ y se obtiene la cantidad $\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\| \leq \hat{K} \varepsilon$ para

clasificar si $\mathbf{x}_L(t)$ pertenece o no a la región $\mathcal{N}_L(\Sigma)$.

Para efectos más prácticos y reducir la cantidad de cálculos, podemos enfocarnos en proyectar sobre \mathcal{V}^\perp y, específicamente, en uno de los vectores que generan a \mathcal{V}^\perp , con el último vector singular izquierdo \mathbf{u}_L , que es el vector que contiene la menor cantidad de información del espacio de columnas de $\mathcal{H}_L(\Sigma_N)$.

El cálculo de proyectar en los subespacio se reduce a proyectar sobre el vector \mathbf{u}_L , obtener esta componente y comparar si es mayor a cierta tolerancia con la cual se va a clasificar anómalo o no. Se obtiene la proyección sobre \mathbf{u}_L con $\mathbf{u}_L^T \mathbf{x}_L(t)$, que es un escalar. Si la magnitud de $\mathbf{u}_L^T \mathbf{x}_L(t)$ es mayor a cierta tolerancia (Tol), se clasifica a $\mathbf{x}_L(t)$ como comportamiento anómalo. Caso contrario, se clasifica a $\mathbf{x}_L(t)$ como no anómalo, dígame,

$$\begin{cases} |\mathbf{u}_L^T \mathbf{x}_L(t)| > Tol \Rightarrow \text{Anomalía} \\ |\mathbf{u}_L^T \mathbf{x}_L(t)| \leq Tol \Rightarrow \text{Sin anomalía.} \end{cases}$$

Este concepto se ilustra de manera gráfica en la Fig. 3.2.

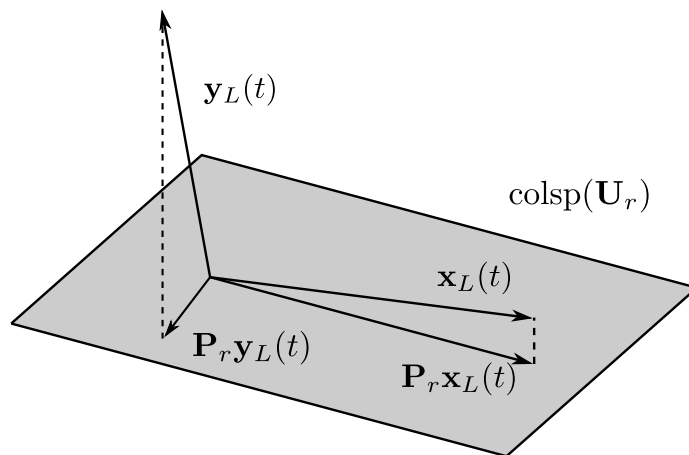


Figura 3.2. Representación de un región de la señal $\mathbf{x}_L(t)$ sin anomalías y una región $\mathbf{y}_L(t)$ con anomalías, y las proyecciones en el subespacio $\text{colsp}(\mathbf{U}_r)$.

En la Fig. 3.2, el vector $\mathbf{x}_L(t)$ no tiene una anomalía y el vector $\mathbf{y}_L(t)$ sí tiene. Se indica cada una de las proyecciones con $\mathbf{P}_r \mathbf{x}_L(t)$ y $\mathbf{P}_r \mathbf{y}_L(t)$ en el plano que representa a $\mathcal{V} = \text{colsp}(\mathbf{U}_r)$. Las líneas punteadas representan $\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\|$ y $\|\mathbf{P}_r \mathbf{y}_L(t) - \mathbf{y}_L(t)\|$. En el caso que es no anómalo para $\mathbf{x}_L(t)$, la magnitud del residuo $\|\mathbf{P}_r \mathbf{x}_L(t) - \mathbf{x}_L(t)\|$ es

pequeña, no va a ser exactamente cero, ya que quitamos información de la señal, ya que pueden haber componentes pequeñas en \mathcal{V}^\perp y, en caso de señales reales, las mediciones van a tener un ruido aleatorio asociado a las mediciones. Este ruido va a estar distribuido de manera uniforme en todas las componentes [13, p. 418], por lo que es inevitable que no haya información de la señal en \mathcal{V}^\perp . El valor de la tolerancia para clasificar como anomalía se puede calibrar para tomar estas consideraciones en cuenta. En caso que un segmento de la señal tenga un comportamiento verdaderamente anómalo y no solo ruido aleatorio, va a tener una mayor proyección sobre \mathcal{V}^\perp , que es lo que puede medir, como se ilustra en la Fig. 3.2 con $\mathbf{y}_L(t)$.

Capítulo 4

Identificación de los puntos anómalos

En los Capítulos pasados se desarrolló la teoría para la identificación de puntos anómalos. En este Capítulo se explica el algoritmo para identificar dichos puntos anómalos y además distintos métodos para calcular el vector que se va a utilizar para la detección de los puntos anómalos.

Para identificar las anomalías se requiere obtener un vector del $\ker_\varepsilon(\mathcal{H}_L(\Sigma_N)^T)$. Por lo discutido en la Sección 3.3, se va a usar el último vector singular izquierdo. Existen varias maneras de calcular este vector, por lo que hay libertad en cómo se calcula y no tiene que ser mediante el método directo de obtener la descomposición SVD de $\mathcal{H}_L(\Sigma_N)$. En las siguientes Secciones se desarrollan maneras distintas de encontrar el último vector singular izquierdo para la detección de anomalías.

4.1. Descomposición SVD de $\mathcal{H}_L(\Sigma_N)$

La primera manera, se basa directamente en la teoría y mediante el cálculo explícito de la descomposición total SVD de

$$\mathcal{H}_L(\Sigma_N) = \mathbf{U}\Sigma\mathbf{V}^T,$$

donde se utiliza una rutina numérica de Numpy `numpy.linalg.svd` [14] para obtener la descomposición SVD. Dependiendo de las dimensiones de $\mathcal{H}_L(\Sigma_N)$, podría ser más rápido obtener la descomposición sobre $\mathcal{H}_L(\Sigma_N)^T$. De la Ec. 2.3, la descomposición SVD es

$$\mathcal{H}_L(\Sigma_N)^T = \mathbf{V}\Sigma\mathbf{U}^T,$$

comparando con la salida de la función de NumPy sobre $\mathcal{H}_L(\Sigma_N)$, se aplica la función sobre $\mathcal{H}_L(\Sigma_N)^T$ y la salida de \mathbf{V} contiene el vector \mathbf{u}_L que se está buscando.

4.2. Descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$

La segunda manera es aprovechar una propiedad que se muestra en la Ec. 4.1. Debido a como se construye la matriz $\mathcal{H}_L(\Sigma_N)$, la matriz va a tener más columnas que filas. Sean L el número de filas y M el número de columnas de $\mathcal{H}_L(\Sigma_N)$. Se puede reducir la cantidad de cálculos para sacar el último vector singular izquierdo calculando la descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$. Esto se puede notar viendo que

$$\begin{aligned}\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T &= \mathbf{U}\Sigma\mathbf{V}^T \left(\mathbf{U}\Sigma\mathbf{V}^T \right)^T \\ &= \mathbf{U}\Sigma\mathbf{V}^T \left(\mathbf{V}^T \right)^T \Sigma^T \mathbf{U}^T \\ &= \mathbf{U}\Sigma\mathbf{V}^T \mathbf{V}\Sigma\mathbf{U}^T \\ &= \mathbf{U}\Sigma\Sigma\mathbf{U}^T \\ &= \mathbf{U}\Sigma^2\mathbf{U}^T.\end{aligned}$$

Por lo tanto

$$\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T = \mathbf{U}\Sigma^2\mathbf{U}^T. \quad (4.1)$$

La Ec. 4.1 es la descomposición SVD para $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$. Al comparar con la notación de la descomposición SVD, con $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ de la Ec. 4.1, los términos respectivos son

$$\begin{aligned}\mathbf{A} &\leftarrow \mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T \\ \mathbf{U} &\leftarrow \mathbf{U} \\ \Sigma &\leftarrow \Sigma^2 \\ \mathbf{V}^T &\leftarrow \mathbf{U}^T.\end{aligned}$$

De la Ec. 4.1, se observa que ambas matrices $\mathcal{H}_L(\Sigma_N)$ y $\mathcal{H}_L(\Sigma)\mathcal{H}_L(\Sigma_N)^T$ tienen el mismo último vector singular izquierdo no nulo. Así que al realizar la descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma)$ y $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ se consiguen los mismos vectores singulares izquierdos.

Las matrices son de dimensiones $\mathcal{H}_L(\Sigma_N) \in \mathbb{R}^{L \times M}$, $\mathcal{H}_L(\Sigma_N)^T \in \mathbb{R}^{M \times L}$, y así, se tiene que $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T \in \mathbb{R}^{L \times L}$. Con la construcción de la matriz $\mathcal{H}_L(\Sigma_N)$ tomando $L < M$, la matriz $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ tiene menos entradas que la matriz $\mathcal{H}_L(\Sigma_N)$, por lo que el cálculo de la descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ va a ser computacionalmente más barato que obtener la descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)$, pero va a tener el paso extra de multiplicar las matrices. Para comparación, se toma el tiempo de realizar la multiplicación

de las matrices y obtener la descomposición SVD para $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$. Se utiliza de nuevo la rutina numérica `numpy.linalg.svd` [14] para obtener la descomposición SVD.

Para aprovechar la propiedad de que $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ es simétrica, la función de NumPy tiene un argumento para indicar que la matriz es simétrica y poder usar rutinas numéricas tomando esto en cuenta y reducir el tiempo de cálculo.

4.3. Descomposición de valores propios sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$

La tercera manera es similar a la de la Sección 4.2. Se aprovecha la forma de la matriz $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$, debido a que es una matriz simétrica [7, p. 165] y se puede usar la descomposición diagonalizada para encontrar el último vector singular izquierdo. De misma Ec. 4.1, la matriz también se muestra en su forma diagonalizada $\mathbf{A} = \mathbf{V}\Delta\mathbf{V}^{-1}$, [1, p. 104] con

$$\begin{aligned}\mathbf{A} &\leftarrow \mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T \\ \mathbf{V} &\leftarrow \mathbf{U} \\ \Delta &\leftarrow \Sigma^2\end{aligned}$$

El último vector singular izquierdo va a ser el vector propio asociado al valor propio más pequeño de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$.

La matriz $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ es una matriz hermítica y existen optimizaciones numéricas que aprovechan este tipo de estructura para reducir la cantidad de FLOPS que se necesitan en comparación a una matriz no simétrica [22, p. 355]. Estas rutinas numéricas ya se encuentran implementadas en NumPy [14] con la función `numpy.linalg.eigh`.

Así que solo se va a comparar el cálculo del tiempo tomado usando la rutinas numéricas para matrices hermíticas, para encontrar el vector propio asociado al valor propio más pequeño de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$.

4.4. Iteración IPMRQ sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$

La cuarta manera que se va a usar para encontrar el vector \mathbf{p} , es usar el Método de la Potencia Inversa con el cociente de Rayleigh (IPMRQ - Inverse Power Method with Rayleigh Quotient por sus siglas en inglés). Los métodos anteriores obtienen todos los vectores para usar en la detección de anomalía cuando podemos hacer la detección usando un solo vector. Aplicando la iteración del IPMRQ, se puede hacer más eficiente el cálculo de ese vector para la detección sin tener que calcular los demás vectores singulares/proprios.

Se procede a explicar los métodos de la potencia y sus variaciones de las cuales depende el IPMRQ.

4.4.1. Método de la potencia, potencia inversa y Cociente de Rayleigh

En toda la subsección 4.4.1, se va a aplicar los métodos en una matriz de la forma $\mathbf{A} = \mathbf{B}^T \mathbf{B}$, con $\mathbf{B} = \mathcal{H}_L(\Sigma_N)^T$. Para las demostraciones, los valores propios de \mathbf{A} son no negativos, ya que para matrices que se les puede realizar una factorización de la forma $\mathbf{A} = \mathbf{B}^T \mathbf{B}$ cumplen con esta propiedad [13, p. 558]. También se cumple que \mathbf{A} es diagonalizable, ya que \mathbf{A} es simétrica, y toda matriz simétrica es diagonalizable [7, p. 165].

Método de la potencia

Sea $\mathbf{A} \in \mathbb{R}^{n \times n}$ una matriz con n vectores y valores propios. Suponga que \mathbf{A} tiene un valor propio mayor en magnitud a los demás valores propios. Ordenemos los valores propios de manera que $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$.

El método de la potencia [22, p. 314] es un técnica iterativa para encontrar el par propio $(\lambda_1, \mathbf{v}_1)$ de \mathbf{A} . Para empezar la iteración, se usa un vector arbitrario \mathbf{x}_0 . El vector se puede escribir como una combinación lineal de los vectores propios \mathbf{v}_i de \mathbf{A} . Se escribe \mathbf{x}_0 en esta base y se obtiene

$$\mathbf{x}_0 = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \dots + c_n \mathbf{v}_n. \quad (4.2)$$

Como hipótesis sobre el vector \mathbf{x}_0 , se necesita que la componente c_1 en la Ec. 4.2 sea distinta de 0 [22, p. 314].

Se aplica la iteración de multiplicar k veces por \mathbf{A} al vector resultante de $\mathbf{A}\mathbf{x}_0$. La primera iteración es $\mathbf{A}\mathbf{x}_0$. La segunda iteración es $\mathbf{A}(\mathbf{A}\mathbf{x}_0) = \mathbf{A}^2\mathbf{x}_0$. La tercera iteración es $\mathbf{A}(\mathbf{A}^2\mathbf{x}_0) = \mathbf{A}^3\mathbf{x}_0$, y así sucesivamente.

Aplicar esta iteración va a producir un vector que converge a la dirección de \mathbf{v}_1 . Se puede ver el resultado de la convergencia con la iteración al vector \mathbf{v}_1 usando un vector arbitrario \mathbf{x}_0 , si se escribe como la combinación lineal de la Ec. 4.2 para probar este resultado. Si se realiza la iteración k veces sobre \mathbf{x}_0 , se obtiene

$$\begin{aligned} \mathbf{A}^k \mathbf{x}_0 &= \mathbf{A}^k (c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \dots + c_n \mathbf{v}_n) \\ &= c_1 \mathbf{A}^k \mathbf{v}_1 + c_2 \mathbf{A}^k \mathbf{v}_2 + \dots + c_n \mathbf{A}^k \mathbf{v}_n \\ &= c_1 \lambda_1^k \mathbf{v}_1 + c_2 \lambda_2^k \mathbf{v}_2 + \dots + c_n \lambda_n^k \mathbf{v}_n. \end{aligned} \quad (4.3)$$

Normalizando el resultado obtenido en la Ec. 4.3, se obtiene

$$\begin{aligned} \frac{\mathbf{A}^k \mathbf{x}_0}{\|\mathbf{A}^k \mathbf{x}_0\|} &= \frac{c_1 \lambda_1^k \mathbf{v}_1 + c_2 \lambda_2^k \mathbf{v}_2 + \cdots + c_n \lambda_n^k \mathbf{v}_n}{\|c_1 \lambda_1^k \mathbf{v}_1 + c_2 \lambda_2^k \mathbf{v}_2 + \cdots + c_n \lambda_n^k \mathbf{v}_n\|} \\ &= \frac{c_1 \lambda_1^k \mathbf{v}_1 + c_2 \lambda_2^k \mathbf{v}_2 + \cdots + c_n \lambda_n^k \mathbf{v}_n}{\sqrt{c_1^2 (\lambda_1^k)^2 + c_2^2 (\lambda_2^k)^2 + \cdots + c_n^2 (\lambda_n^k)^2}}. \end{aligned} \quad (4.4)$$

Factorizando $c_1^2 (\lambda_1^k)^2$ de la raíz del denominador de la Ec. 4.4 se obtiene

$$\frac{\mathbf{A}^k \mathbf{x}_0}{\|\mathbf{A}^k \mathbf{x}_0\|} = \frac{\frac{c_1}{|c_1|} \mathbf{v}_1 + \frac{c_2}{|c_1|} \left(\frac{\lambda_2}{\lambda_1}\right)^k \mathbf{v}_2 + \cdots + \frac{c_n}{|c_1|} \left(\frac{\lambda_n}{\lambda_1}\right)^k \mathbf{v}_n}{\sqrt{1 + \left(\frac{c_2}{c_1} \left(\frac{\lambda_2}{\lambda_1}\right)^k\right)^2 + \cdots + \left(\frac{c_n}{c_1} \left(\frac{\lambda_n}{\lambda_1}\right)^k\right)^2}}. \quad (4.5)$$

Por hipótesis, como $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$, se cumple que $|\lambda_i/\lambda_1| < 1$ para $2 \leq i \leq n$. Por lo tanto $|\lambda_i/\lambda_1|^k \rightarrow 0$ cuando $k \rightarrow \infty$. Se aplica la iteración para un k muy grande en la Ec. 4.5, para remover varios términos, se obtiene

$$\begin{aligned} \frac{\mathbf{A}^k \mathbf{x}_0}{\|\mathbf{A}^k \mathbf{x}_0\|} &= \frac{\frac{c_1}{|c_1|} \mathbf{v}_1 + \frac{c_2}{|c_1|} \left(\frac{\lambda_2}{\lambda_1}\right)^k \mathbf{v}_2 + \cdots + \frac{c_n}{|c_1|} \left(\frac{\lambda_n}{\lambda_1}\right)^k \mathbf{v}_n}{\sqrt{1 + \left(\frac{c_2}{c_1} \left(\frac{\lambda_2}{\lambda_1}\right)^k\right)^2 + \cdots + \left(\frac{c_n}{c_1} \left(\frac{\lambda_n}{\lambda_1}\right)^k\right)^2}} \\ &\approx \frac{c_1}{|c_1|} \mathbf{v}_1, \end{aligned}$$

que es lo que se quería demostrar, que la iteración converge en la dirección de \mathbf{v}_1 con un vector arbitrario \mathbf{x}_0 .

El método se puede resumir en aplicar las iteraciones

$$\mathbf{y}_{k+1} = \mathbf{A} \mathbf{x}_k \quad (4.6)$$

$$\mathbf{x}_{k+1} = \mathbf{y}_{k+1} / \|\mathbf{y}_{k+1}\| \quad (4.7)$$

$$\lambda^{(k+1)} = \mathbf{x}_{k+1}^T \mathbf{A} \mathbf{x}_{k+1} \quad (4.8)$$

donde $\lambda^{(k)}$ representa la aproximación al valor propio de mayor magnitud y \mathbf{x}_k la aproximación al respectivo vector propio normalizado. Se sigue la iteración hasta que se cumpla una condición de convergencia. El orden de convergencia del vector propio, depende de la razón del segundo y primer valor propio dominante. La magnitud del error con el vector propio

disminuye en un factor de $|\lambda_2/\lambda_1|$ en cada iteración y es una convergencia lineal [22, p. 314-315].

Para nuestro interés, se quiere encontrar el vector propio asociado al valor propio más pequeño de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$. No se puede usar directamente el método de potencias, ya que el método converge al vector propio asociado al valor propio dominante. Se utiliza una modificación del método de potencias para encontrar otros vectores y valores propios. Se usan los siguientes resultados para hacer la variación [13, p. 501] del método de potencias.

Método de potencia inversa

Se prueban los Lemas 4.1 y 4.2 [13, p. 501] para explicar la iteración del método de la potencia inversa.

Lema 4.1. *Sea $\mathbf{A} \in \mathbb{R}^{n \times n}$, con par propio (λ, \mathbf{x}) . Si $\alpha \in \mathbb{R}$, entonces $\mathbf{A} - \alpha \mathbf{I}_n$ tiene par propio $(\lambda - \alpha, \mathbf{x})$*

Prueba: Como (λ, \mathbf{x}) es un par propio de \mathbf{A} , se cumple que

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}.$$

Sumando a ambos lados $-\alpha\mathbf{x}$, se obtiene

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \lambda\mathbf{x} \\ \Rightarrow \mathbf{A}\mathbf{x} - \alpha\mathbf{x} &= \lambda\mathbf{x} - \alpha\mathbf{x} \\ \Rightarrow \mathbf{A}\mathbf{x} - \alpha\mathbf{I}_n\mathbf{x} &= \lambda\mathbf{x} - \alpha\mathbf{x} \\ \Rightarrow (\mathbf{A} - \alpha\mathbf{I}_n)\mathbf{x} &= (\lambda - \alpha)\mathbf{x}. \end{aligned} \tag{4.9}$$

De la Ec. 4.9, se obtiene que $(\lambda - \alpha, \mathbf{x})$ es un par propio de $\mathbf{A} - \alpha\mathbf{I}_n$. \square

Lema 4.2. *Sea $\mathbf{A} \in \mathbb{R}^{n \times n}$ una matriz invertible. Si (λ, \mathbf{x}) , es un par propio de \mathbf{A} , entonces $(\lambda^{-1}, \mathbf{x})$ es un par propio de \mathbf{A}^{-1} .*

Prueba: Como (λ, \mathbf{x}) es un par propio, se cumple que

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x},$$

con $\lambda \neq 0$. Multiplicando a ambos lados por $\lambda^{-1}\mathbf{A}^{-1}$ se obtiene

$$\begin{aligned} \lambda^{-1}\mathbf{A}^{-1}\mathbf{A}\mathbf{x} &= \lambda^{-1}\mathbf{A}^{-1}\lambda\mathbf{x} \\ \Rightarrow \lambda^{-1}\mathbf{x} &= \mathbf{A}^{-1}\mathbf{x}, \end{aligned}$$

que cumple con la definición de valor propio para \mathbf{A}^{-1} . \square

Combinando los Lemas 4.1 y 4.2, se puede construir un método basado en el método de potencias, que tiene la limitación que solo encuentra el valor propio de mayor magnitud. En particular, si se escoge un α que no sea un valor propio de \mathbf{A} , la resta $\mathbf{A} - \alpha\mathbf{I}_n$ tiene valores propios $\lambda - \alpha$ (Lema 4.1). Por hipótesis, α no es valor propio, entonces $\lambda - \alpha$ va a ser distinto de 0, y por lo tanto $\mathbf{A} - \alpha\mathbf{I}_n$ es invertible. Los valores propios de $(\mathbf{A} - \alpha\mathbf{I}_n)^{-1}$ son $(\lambda - \alpha)^{-1}$ (Lema 4.2). Se puede escoger un α de tal manera que se aproxime a un valor propio en específico, para que $(\lambda - \alpha)^{-1}$ sea el valor propio de mayor magnitud de $(\mathbf{A} - \alpha\mathbf{I}_n)^{-1}$ o bien usar $\alpha = 0$ y se puede aproximar el valor propio λ de menor magnitud con su vector propio.

Se empieza una iteración similar a las de las Ec. 4.6, 4.7 y 4.8, y se escoge un α para aproximar el valor y vector propio más cercano a α , con un cambio pequeño. Para seguir la iteración 4.6, se necesita calcular $\mathbf{y}_{k+1} = (\mathbf{A} - \alpha\mathbf{I}_n)^{-1} \mathbf{x}_k$. Como regla general, es aconsejable no calcular la inversa de una matriz si se puede evitar [3, p. 65]. Se puede obtener \mathbf{y}_{k+1} resolviéndolo para el sistema $(\mathbf{A} - \alpha\mathbf{I}_n) \mathbf{y}_{k+1} = \mathbf{x}_k$ y se aplican los mismos pasos del método de la potencia. En resumen, el método de la potencia inversa consiste en aplicar los pasos: dado \mathbf{x}_0 ,

$$(\mathbf{A} - \alpha\mathbf{I}_n) \mathbf{y}_{k+1} = \mathbf{x}_k \quad (4.10)$$

$$\mathbf{x}_{k+1} = \mathbf{y}_{k+1} / \|\mathbf{y}_{k+1}\| \quad (4.11)$$

$$\lambda^{(k+1)} = \mathbf{x}_{k+1}^T \mathbf{A} \mathbf{x}_{k+1} \quad (4.12)$$

hasta que se cumpla una condición de convergencia. En la Ec. 4.10 se resuelve para \mathbf{y}_{k+1} , y $(\lambda_{k+1}, \mathbf{x}_{k+1})$ es la aproximación al par propio (λ, \mathbf{x}) para el valor propio más cercano a α .

La convergencia para el vector propio con método de potencia inversa es una convergencia lineal, donde el error decrece como $|\lambda - \alpha|/|\lambda' - \alpha|$ con cada iteración, donde λ es el valor propio más cercano a α y λ' es el segundo valor propio más cercano a α [17, p. 206-207].

Se puede acelerar aún más el método si puede variar el valor de α con cada iteración, lo que lleva a la última variación del método de potencias, que es la que se aplica en el algoritmo para encontrar el vector para detectar anomalías.

Método de la potencia inversa con Cociente de Rayleigh

Como última variación del método, se puede hacer la iteración de la potencia inversa de las Ec. 4.10, cambiando el valor de α con cada iteración. El error de convergencia del vector propio a aproximar con el método de la potencia inversa decrece en $|\lambda - \alpha|/|\lambda' - \alpha|$ con cada iteración. Si se actualiza el valor de α a que se aproxime cada vez más a λ , el error con cada iteración se va a hacer más pequeño con este pequeño cambio.

Se usa el valor de α variable con la aproximación a λ de la iteración respectiva, con $\lambda^{(k)} = \mathbf{x}_k^T \mathbf{A} \mathbf{x}_k$. A esta cantidad se le llama el cociente de Rayleigh.

Usando las iteraciones en las Ec. 4.10, 4.11 y 4.12, junto con el valor variable de $\alpha \rightarrow \lambda^{(k)}$, la iteración a realizar es

$$\begin{aligned} (\mathbf{A} - \lambda^{(k)} \mathbf{I}_n) \mathbf{y}_{k+1} &= \mathbf{x}_k \\ \mathbf{x}_{k+1} &= \mathbf{y}_{k+1} / \|\mathbf{y}_{k+1}\| \\ \lambda^{(k+1)} &= \mathbf{x}_{k+1}^T \mathbf{A} \mathbf{x}_{k+1}. \end{aligned}$$

Con este cambio, el orden de convergencia del vector propio a aproximar tiene una **convergencia cúbica** [16, p. 78-79], que mejora que el método de potencia y de potencia inversa con convergencia lineal.

4.4.2. Método usando el IPMRQ

De la Ec. 4.1, se puede ver que se necesita el vector propio asociado al valor propio más pequeño de de la matriz $\mathcal{H}_L(\Sigma_N) \mathcal{H}_L(\Sigma_N)^T$.

Para la inicialización del IPMRQ, los valores propios de la matriz $\mathcal{H}_L(\Sigma_N) \mathcal{H}_L(\Sigma_N)^T$ son no negativos. Como semilla inicial se puede usar un número pequeño, o bien, iniciar con $\lambda_0 = 0$ y empezar la iteración. Se escoge un vector inicial aleatorio \mathbf{x}_0 también de semilla inicial para iniciar la iteración.

El algoritmo IPMRQ con los detalles de la ejecución se encuentra en el Algoritmo 1 y la implementación en Python en `InversePowerMethodRayleighQuotient.py`. En el código propio para el IMPRQ, no se almacena cada uno de los \mathbf{x}_k ni λ_k , si no, solo se almacena uno a la vez, ya que no necesitan todos los \mathbf{x}_k ni λ_k , solo los últimos obtenidos. En el Algoritmo 1 en las líneas 3 y 8 se utiliza la rutina numérica `numpy.linalg.solve` para resolver para \mathbf{x}_{k+1} en las iteraciones.

Algorithm 1 Inverse Power Method with Rayleigh Quotient

Input: $\mathbf{A}, tol, k_{max}, \lambda_0$

- 1: $n \leftarrow \text{size}(\mathbf{A})[0]$ ▷ Obtener el tamaño de $\mathbf{A} \in \mathbb{R}^{n \times n}$
- 2: $\mathbf{x}_0 \leftarrow \text{rand}(n)$ ▷ Distribución uniforme entre -1 y 1 , con n entradas
- 3: Resolver \mathbf{x}_1 para $(\mathbf{A} - \lambda_0 \mathbf{I}_n) \mathbf{x}_1 = \mathbf{x}_0$ ▷ Iteración inicial
- 4: $\mathbf{x}_1 \leftarrow \mathbf{x}_1 / \|\mathbf{x}_1\|$ ▷ Normalización
- 5: $\lambda_1 \leftarrow \mathbf{x}_1^T \mathbf{A} \mathbf{x}_1$
- 6: $k \leftarrow 1$
- 7: **while** $k \leq k_{max}$ **do**
- 8: Resolver \mathbf{x}_{k+1} para $(\mathbf{A} - \lambda_k \mathbf{I}_n) \mathbf{x}_{k+1} = \mathbf{x}_k$
- 9: $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_{k+1} / \|\mathbf{x}_{k+1}\|$ ▷ Normalización
- 10: $\lambda_{k+1} \leftarrow \mathbf{x}_{k+1}^T \mathbf{A} \mathbf{x}_{k+1}$
- 11: $error \leftarrow |\lambda_{k+1} - \lambda_k|$ ▷ Verificar la convergencia
- 12: **if** $error < tol$ **then**
- 13: **return** $(\mathbf{x}_k, \lambda_k, k)$
- 14: **end if**
- 15: $k \leftarrow k + 1$
- 16: **end while**
- 17: **return** $(\mathbf{x}_k, \lambda_k, k)$

La iteración IPMRQ casi siempre converge y cuando converge tiene una convergencia cúbica. [16, p. 72]. En los experimentos realizados, el IPMRQ sí converge sin problemas.

4.5. Sobre la escogencia de ε

En teoría, se necesita escoger ε para encontrar $\mathbf{p} \in \ker_\varepsilon(\mathcal{H}_L(\Sigma_N)^T)$ para en el algoritmo usado para escoger \mathbf{p} . Para un ε suficientemente pequeño, podría ocurrir que el único elemento del $\ker_\varepsilon(\mathcal{H}_L(\Sigma_N)^T)$ sea $\mathbf{0}$ en el caso que todos los valores singulares de $\mathcal{H}_L(\Sigma_N)$ sean mayores a ε . Si este fuese el caso, no se encontraría un vector útil para obtener las anomalías. No se ha discutido como seleccionar un valor para ε para evitar este problema y poder tener un vector que usar para la detección de anomalías.

En la práctica, no hay necesidad de especificar el valor de ε de antemano para encontrar el vector $\mathbf{p} \in \ker_\varepsilon(\mathcal{H}_L(\Sigma_N)^T)$ para la detección de las anomalías. En el primer método de obtener la descomposición SVD de $\mathcal{H}_L(\Sigma_N)$, se usa el vector singular izquierdo \mathbf{u} asociado al menor valor singular. Con la descomposición obtenida, se usan $\varepsilon = \sigma_{\min}$ y $\mathbf{p} = \mathbf{u}_{\min}$, en caso que $\sigma_{\min} > 0$. En el caso que $\ker(\mathcal{H}_L(\Sigma_N)^T)$ sea no vacío, se usa un vector que pertenece al $\ker(\mathcal{H}_L(\Sigma_N)^T)$ para que sea el \mathbf{p} y se puede usar cualquier ε .

En caso de seleccionar el método de la descomposición SVD de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$, la descomposición igual se va a encargar de calcular $\mathbf{p} = \mathbf{u}_L$ para ser usado para detectar las anomalías. Para escoger el ε con esta descomposición, usando la Ec. 4.1, de la descomposición SVD de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ se usa σ_{\min}^2 como el menor valor singular positivo. Se obtiene la raíz y usamos el mismo resultado para $\varepsilon = \sigma_{\min}$. En el caso que $\ker(\mathcal{H}_L(\Sigma_N)^T)$ sea no vacío, se calcula un vector que pertenece al $\ker(\mathcal{H}_L(\Sigma_N)^T)$ para que sea el vector \mathbf{p} y se usa cualquier $\varepsilon > 0$.

En caso de usarse la descomposición de valores propios de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$, se sigue una técnica similar de escoger ε a como se selecciona con la descomposición de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$. Se utiliza el vector propio asociado a λ_{\min} para que sea el \mathbf{p} y ya que el valor propio se ve que son valores singulares cuadrados, obtiene σ_{\min} de $\lambda_{\min} = \sigma_{\min}^2$ y usa $\varepsilon = \sigma_{\min}$.

Si se usa el método de IPMRQ, tomando en cuenta que la iteración va a ser realizada sobre la matriz $\mathcal{H}(\Sigma_N)\mathcal{H}(\Sigma_N)^T$, entonces sus valores propios van a ser $\lambda_i = \sigma_i^2$, y se toma ε como la raíz del valor propio al cual el algoritmo IPMRQ converja. Como argumento inicial, se escoge un valor cercano ε' que considere cercano a ε para que el conjunto $\ker_{\varepsilon}(\mathcal{H}(\Sigma_N)^T)$ no sea nulo, o bien se puede iniciar la iteración con $\varepsilon' = 0$ y dejar que converja al valor singular cuadrado mayor a 0 en caso que la matriz sea de rango completo.

4.6. Complejidad de los métodos

Como punto de comparación entre los métodos, se analiza la cantidad de FLOPs que requiere cada uno de los métodos para obtener el último vector singular izquierdo \mathbf{u}_L .

Para matrices hermíticas, la complejidad del algoritmo para los vectores propios va en el orden $O(n^3)$ [3, p. 380]. Los algoritmos para obtener la descomposición SVD también tienen una complejidad de $O(n^3)$ [3, p. 385]. La complejidad del IPMRQ va a depender del algoritmo que se usa para resolver el sistema $(\mathbf{A} - \lambda_k \mathbf{I}_n)\mathbf{y}_{k+1} = \mathbf{x}_k$ en cada iteración. El algoritmo que usa NumPy con la función `numpy.linalg.solve` se basa en la función `_gesv` de LAPACK [2], que utiliza la descomposición LU que tiene una complejidad de $O(n^3)$ FLOPs [17, p. 152].

Todos los métodos tienen una complejidad del mismo orden $O(n^3)$ de FLOPs realizados. Una ventaja que ofrece el IPMRQ es que sólo calcula el último vector singular izquierdo y no tiene la necesidad de calcular los demás vectores singulares izquierdos a diferencia de los métodos de la descomposición SVD y la descomposición de valores propios hermítica, que ambos se basan en algoritmos QR para realizar las descomposiciones, donde se genera toda la base.

4.7. Identificación de las anomalías

Una vez obtenido el vector $\mathbf{p} \in \ker_\varepsilon (\mathcal{H}_L(\Sigma_N)^T)$ para la identificación de las anomalías, se procede a aplicarlo sobre la señal de entrada para identificar los puntos anómalos. Se define la función para este propósito.

Definición 4.3. Dado una tolerancia $\delta > 0$, un entero $L \geq 1$, un muestreo de una serie de tiempo $\Sigma = \{x_t\}_{t=1}^T \subset \Sigma_\infty$, un (L, ε) -PAD \mathbf{P} para Σ_∞ basado en Σ_N , un $\mathbf{p} \in \ker(\mathbf{P}^T)$, se escribe $\phi_\delta(\mathbf{p}, \mathbf{x}_L(t))$ para denotar la función de detección de anomalías en la región $\mathbf{x}_L(t)$ dada por la expresión

$$\phi_\delta(\mathbf{p}, \mathbf{x}_L(t)) = H_\delta \left(|\mathbf{p}^T \mathbf{x}_L(t)| \right)$$

donde $H_\delta(x)$ es la función de la Def. 2.5. ◇

Observación 4.4. En el algoritmo de detección de anomalías, se hace un traslado de la señal de manera que el promedio de todos los puntos de la señal de entramiento Σ_N tengan promedio cero. Los puntos de la señal a analizar también se les realiza el desplazamiento previo de manera que todo el análisis de anomalías se realiza sobre la señal desplazada. No se pierde generalidad aplicando este desplazamiento ya que siempre se puede realizar.

De la Def. 4.3, con la función $H_\delta(x)$, se realiza el producto punto de \mathbf{p} y un segmento $\mathbf{x}_L(t)$ para determinar si la región es anómala. En la implementación del algoritmo de este documento, se le asigna la anomalía a un punto en específico y no a una región. La manera de asignar la anomalía a un punto usando la función ϕ_δ , es midiendo la proyección en el subespacio con dos segmentos temporales que incluyen al punto a medir. Los segmentos temporales a usar se muestran de manera gráfica en la Fig. 4.1.

El primer segmento temporal para medir la anomalía en el punto x_k es usar el segmento temporal que contiene al punto x_k como última entrada del vector $\mathbf{x}_L(t)$, que corresponde al segmento temporal $\mathbf{x}_L(k)$. El segundo segmento temporal corresponde a usar un

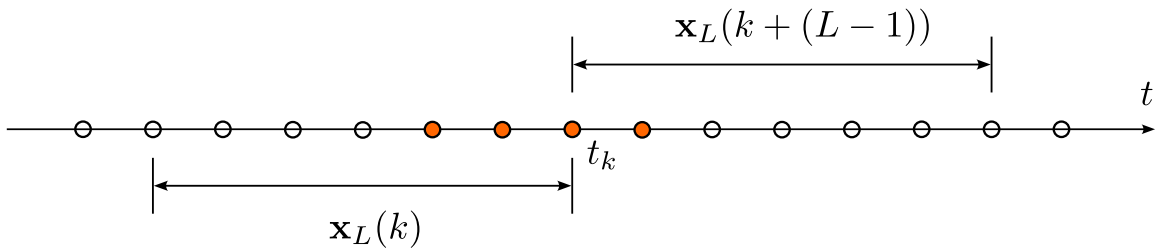


Figura 4.1. Medición de la anomalía en el punto t_k con los segmentos antes y después incluyendo al punto. Los puntos naranjas representan anomalías.

vector $\mathbf{x}_L(t)$ que contenga al punto x_k como primera entrada. Este corresponde al segmento temporal $\mathbf{x}_L(k + L - 1)$. Se mide la proyección de ambos segmentos temporales, se le aplica la función ϕ_δ y se toma una decisión de clasificación en base a esas dos proyecciones.

La heurística de usar ambas regiones, es que bastantes anomalías no se van a presentar como puntos aislados y muchos de los puntos vecinos también van a ser anómalos. Se miden con segmentos temporales disjuntos, excepto por el punto en común t_k y con eso se toma una decisión para clasificar el punto como anómalo o no. El segmento $\mathbf{x}_L(k)$ contiene el punto t_k junto con los puntos vecinos antes, el segmento $\mathbf{x}_L(k + L - 1)$ también contiene el punto t_k y los puntos vecinos después.

Con ambas proyecciones $\phi_\delta(\mathbf{p}, \mathbf{x}_L(k))$ y $\phi_\delta(\mathbf{p}, \mathbf{x}_L(k + L - 1))$, se le aplica una función lógica para la decisión final. Sea \mathcal{L} la función lógica a aplicar. Se usa a $\mathcal{L}(\cdot, \cdot)$ como las funciones OR o AND, donde los argumentos son $\phi_\delta(\mathbf{p}, \mathbf{x}_L(k))$ y $\phi_\delta(\mathbf{p}, \mathbf{x}_L(k + L - 1))$. La clasificación del punto t_k se completa con la función

$$\mathcal{L}(\phi_\delta(\mathbf{p}, \mathbf{x}_L(k)), \phi_\delta(\mathbf{p}, \mathbf{x}_L(k + L - 1))) \quad (4.13)$$

y se aplica esta función a todos los puntos de la serie de tiempo Σ que se quiere analizar.

Observación 4.5. No es necesario tener los dos detectores en paralelo. Con un solo detector se pueden hacer las mediciones. El si escoger un solo detector o cual función lógica aplicar, se decide con una calibración manual.

Usando la Ec. 4.13 para la detección de un solo punto, se puede escalar la identificación para todos los puntos de la serie Σ . Se puede aplicar la Ec. 4.13 sobre todos los puntos, sin embargo, nótese que si se aplica la detección en el punto t_k , se requiere calcular

$\mathbf{p}^T \mathbf{x}_L(k + L - 1)$ y cuando se necesite calcular la anomalía en el punto t_{k+L-1} , de nuevo se requiere el valor de $\mathbf{p}^T \mathbf{x}_L(k + L - 1)$. Para evitar calcular los productos punto dos veces, se calcula el producto punto $\mathbf{p}^T \mathbf{x}_L(t)$ para los distintos t de toda la señal y se almacena el resultado.

Sean \mathbf{d}_ℓ , \mathbf{d}_r y \mathbf{d} los vectores donde almacenan los resultados de $\phi_\delta(\mathbf{p}^T \mathbf{x}_L(t))$. Todos estos vectores son del mismo tamaño de la señal Σ a analizar. El vector \mathbf{d}_ℓ es el vector de etiquetado izquierdo, \mathbf{d}_r es el vector de etiquetado derecho y \mathbf{d} el vector de etiquetado final. Estos tres vectores se inicializan con todas las entradas igual a 0. Los nombres y uso de \mathbf{d}_ℓ y \mathbf{d}_r vienen de usar el punto t_k a analizar siendo el punto izquierdo o derecho del segmento temporal respectivo, y se almacena en la entrada $\mathbf{d}_\ell[k]$ y $\mathbf{d}_r[k]$ respectivamente para la detección de anomalía del punto t_k . Esto se ejemplifica en la Fig. 4.2. Para no repetir el

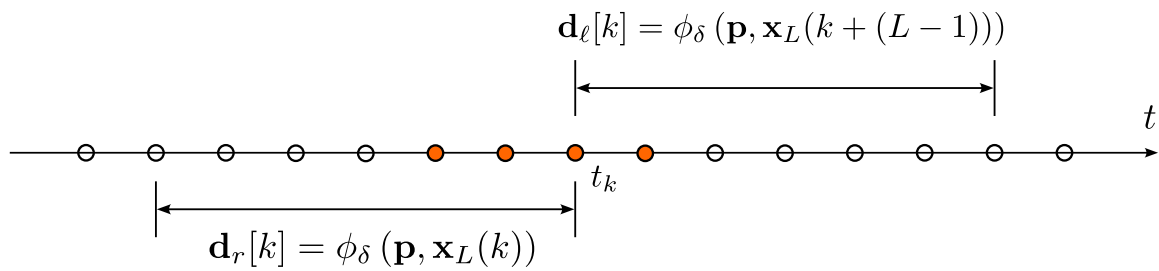


Figura 4.2. Asignación valores de $\mathbf{d}_\ell[k]$ y $\mathbf{d}_r[k]$ para el punto k .

cálculo de $\phi_\delta(\mathbf{p}^T \mathbf{x}_L(k))$ se almacena en la entrada $\mathbf{d}_\ell[k - (L - 1)]$, o dígase, $\mathbf{d}_\ell[k - L + 1]$. El cómo se almacenan los vectores \mathbf{d}_ℓ y \mathbf{d}_r para un t_k particular, se muestra en la Fig. 4.3. Se realiza este procedimiento sin repetir el cálculo de $\mathbf{p}^T \mathbf{x}_L(t)$, para los puntos de la serie

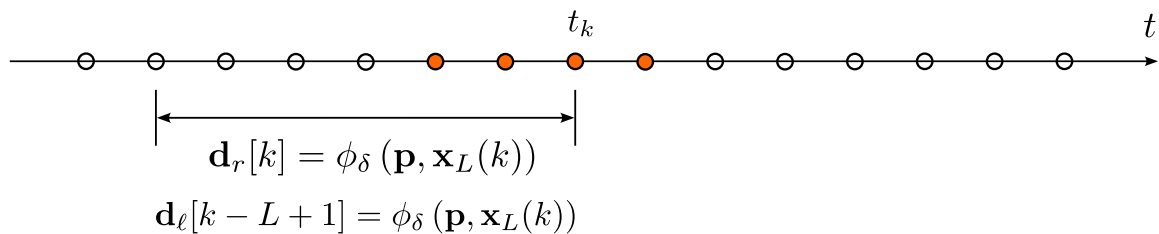


Figura 4.3. Asignación valores de $\mathbf{d}_\ell[k - L + 1]$ y $\mathbf{d}_r[k]$ para el punto k sin repetir cálculos.

de tiempo Σ con T puntos, desde $k = L$ hasta $k = T$, como se muestra en la Fig. 4.4.

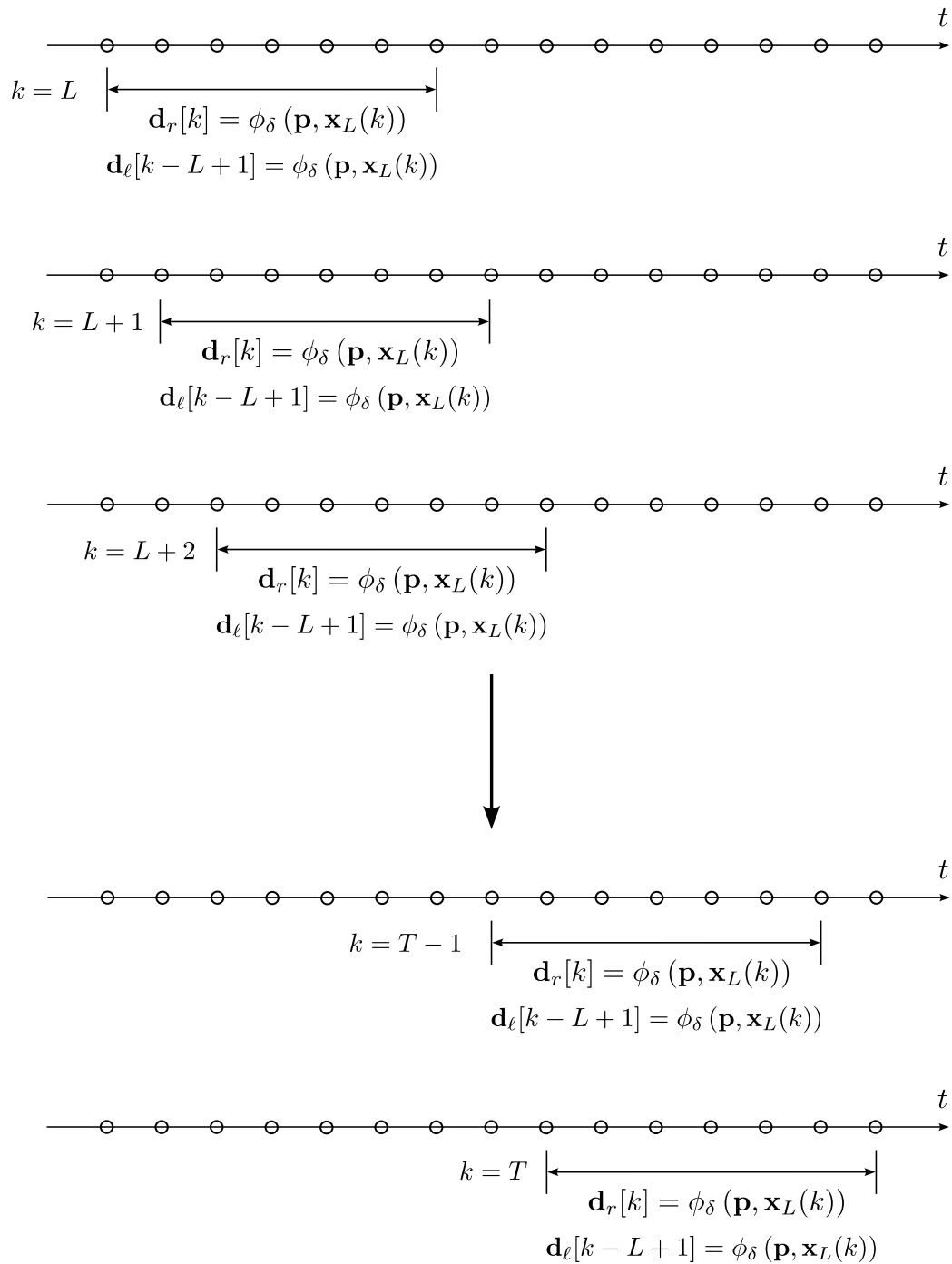


Figura 4.4. Asignación valores de $\mathbf{d}_\ell[k - L + 1]$ y $\mathbf{d}_r[k]$ para todos los puntos.

Habiendo descrito todos los pasos para la identificación de los puntos anómalos, el algoritmo final de los pasos, se muestran en el Algoritmo 2.

Algorithm 2 Determinación puntual de las anomalías

Input: $L, \mathbf{p}, \delta, \Sigma_N, \Sigma, \mathcal{L}$

- 1: Obtener $\mathbf{p} \in \ker_{\varepsilon}(\mathcal{H}_L(\Sigma_N)^T)$
- 2: $T \leftarrow \text{size}(\Sigma)$ ▷ Tamaño de la serie de tiempo.
- 3: $\mathbf{a}, \mathbf{d}, \mathbf{d}_\ell, \mathbf{d}_r \leftarrow \text{zeros}(T)$ ▷ Inicializar vectores de tamaño T .
- 4: **for** $k = L, \dots, T$ **do**
- 5: $\mathbf{x}_L(i) = \Sigma[k - L + 1 : i]$
- 6: $val \leftarrow \phi_\delta(\mathbf{p}, \mathbf{x}_L(k))$
- 7: $\mathbf{d}_\ell[k - L + 1] \leftarrow val$
- 8: $\mathbf{d}_r[k] \leftarrow val$
- 9: **end for**
- 10: **for** $i = 1, \dots, T$ **do**
- 11: $\mathbf{d}[i] \leftarrow \mathcal{L}(\mathbf{d}_\ell[i], \mathbf{d}_r[i])$
- 12: $\mathbf{a}[i] \leftarrow \Sigma[i] \cdot \mathbf{d}[i]$
- 13: **end for**
- 14: **return** (\mathbf{d}, \mathbf{a})

El Algoritmo 2 contiene todos los pasos desde encontrar el vector \mathbf{p} con una señal de entrenamiento con el comportamiento regular de la señal hasta la identificación de los puntos anómalos. En el Capítulo 5 se muestran los resultados del algoritmo total con distintas señales.

Observación 4.6. El Algoritmo 2 funciona conociendo toda la serie de tiempo Σ a analizar. El algoritmo puede ser modificado para recibir la señal en tiempo real. Para la identificación en tiempo real del punto k , se tienen que medir primero $L - 1$ puntos después de k para la identificación de la anomalía en k , como se muestra en la Fig. 4.2. Ya que se ocupan medir $L - 1$ después, se puede seguir usando el almacenamiento sobre los vectores de etiquetado \mathbf{d}_ℓ y \mathbf{d}_r de la forma como se muestra en la Fig. 4.3 para no tener que repetir el cálculo de los productos puntos $\mathbf{p}^T \mathbf{x}_L(t)$.

Observación 4.7. No se ha discutido el como escoger el valor de L . La manera realizada en el documento es encontrar un L de manera empírica calibrando la detección para que no tenga muchos falsos positivos ni negativos.

Capítulo 5

Experimentos y Resultados

5.1. Equipo y experimentos

Con la teoría y algoritmos desarrollados en el documento, se pone a prueba en mediciones sobre señales reales para la detección de anomalías. Se prueban los métodos usando tres señales distintas. La primera señal es una señal sintética y las otras dos son señales reales. Dichas señales reales corresponden a datos de aceleración recolectados usando el software Phyphox [15], que es una aplicación que se puede instalar en teléfonos celulares y que tiene la capacidad de medir cantidades físicas usando los propios aparatos de medición que contenga el celular. Para los experimentos realizados, la serie de tiempo es la aceleración del celular, que es medida usando el acelerómetro de un smartphone Android A21. El software, para ambas señales, realiza una medición cada $\overline{\Delta t} = 7.93$ ms en promedio, con desviación estandar de 5.02×10^{-2} ms, que es una recolección uniforme de datos, para que cumpla con la hipótesis que los datos estén igualmente espaciados en el tiempo.

Los experimentos numéricos para la detección de anomalías fueron realizados usando Python 3.9.12, NumPy 1.22.3 y SciPy 1.8.0 en una PC con Linux Mint 20.2 Uma, un Intel Core i7 8700 de 6 núcleos y G.Skill RipJaws Serie V 8 GB DDR4 2400. Para estos experimentos, los parámetros N y L que son usados fueron encontrados calibrando de manera manual usando el Jupyter notebook `anomaly_detector_calibration_widget.ipynb`. Los parámetros se varían y se seleccionan cuando las anomalías se indentificaban de manera apropiada, sin clasificar regiones de comportamiento normal como una anomalía.

Para encontrar un vector \mathbf{p} que pertenezca al kernel aproximado, se usan cuatro maneras. La primera es obtener la descomposición SVD de $\mathcal{H}_L(\Sigma_N)$, la segunda es calcular la descomposición SVD de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$, la tercera es calculando la descomposición de

la forma diagonalizada de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$, aprovechando su forma hermítica, y la cuarta es aplicar la iteración del IPMRQ sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$.

Para todas las señales se escoge el N de tal manera para usar información desde el punto inicial hasta algún punto que se vea que no contenga una anomalía.

5.1.1. Señal sintética

El primer ejemplo es una señal sintética formada por una suma de cosenos y senos con distinta frecuencia. Se le agrega un ruido pequeño a ciertas partes de la señal para simular un comportamiento anómalo como se muestra en la Fig. 5.1. Estos segmentos con ruido son los que se desean detectar con el algoritmo. Los datos para la señal sintética se encuentran en el archivo `synthetic_signal.csv`.

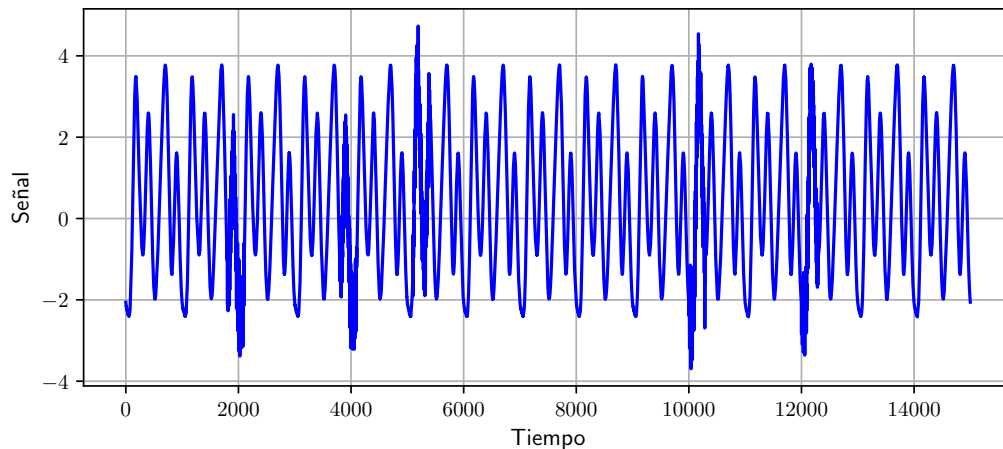


Figura 5.1. Señal sintética con ruido añadido.

5.1.2. Máquina vibradora

Una de las dos señales reales que se usaron es una máquina vibradora para masajes de espalda. La máquina tiene niveles para aumentar/disminuir la intensidad del masaje. La señal es la aceleración de la máquina. Para medir la aceleración de la máquina, se adhiere el celular con cinta adhesiva a la máquina y se utiliza la aplicación Phypbox. Para generar un estado normal para la señal, se coloca la intensidad en el mínimo que permite la máquina. Para generar una anomalía, aumentamos de manera súbita la intensidad de la vibración, y

se baja la intensidad al mínimo para regresar al estado normal. Ya que la intensidad varía de manera continua, setear el estado de operación en el mínimo que permite la máquina permite volver al mismo estado de manera controlada.

En la Fig. 5.2 se muestra una foto del arreglo experimental, con la máquina vibradora y el celular con la aplicación Phyphox abierta con las mediciones de la aceleración.



Figura 5.2. Máquina vibradora de masajes con el celular para medir la aceleración.

La señal generada se muestra en la Fig. 5.3. Se pueden ver tres incrementos en la intensidad de la máquina. Esos son los puntos que queremos identificar. Los datos de la medición de la vibración de esta máquina se encuentran en el archivo `signal_machine_with_time.csv`.

5.1.3. Caminata

Para medir el desempeño del algoritmo en otra señal real, específicamente enfocado a posible uso de dispositivos de salud de uso cotidiano, se mide el ritmo de caminata de una persona. Se usa de nuevo que la señal sea la aceleración del celular mientras lo usa una persona que está caminando. Para generar la señal, el autor procede a caminar a cierto ritmo, procurando que sea lo más constante que se pueda. Se sostiene el celular en la mano y el brazo también se mueve a un ritmo constante, con el fin de obtener una aceleración similar a la que se obtendría con un reloj inteligente para usos de fitness/salud general.

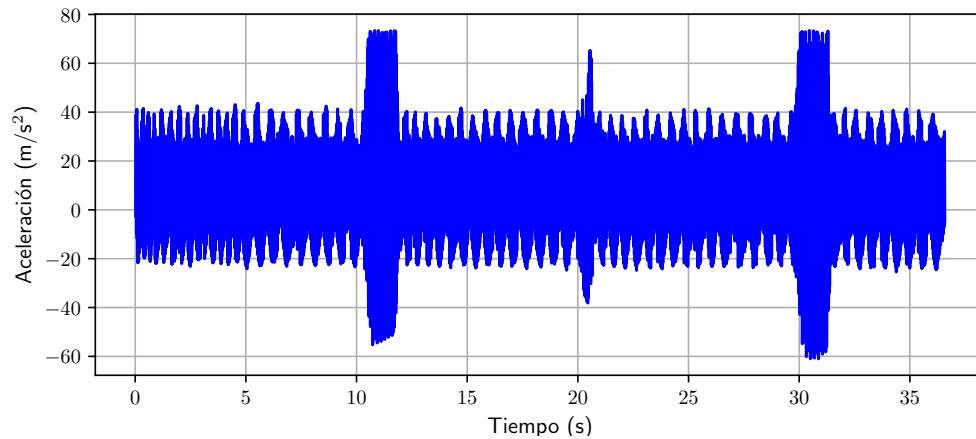


Figura 5.3. Señal de la máquina vibradora.

Para generar una anomalía, se hace un cambio abrupto en el caminado, como un salto o detenerse de manera súbita y se empieza a caminar de nuevo. La aceleración obtenida caminando con los cambios abruptos se muestra en la Fig. 5.4. Se pueden ver los cambios abruptos en la caminata en el aumento de la magnitud de la aceleración. Los datos se encuentran en el archivo `signal_walking_with_time.csv`.

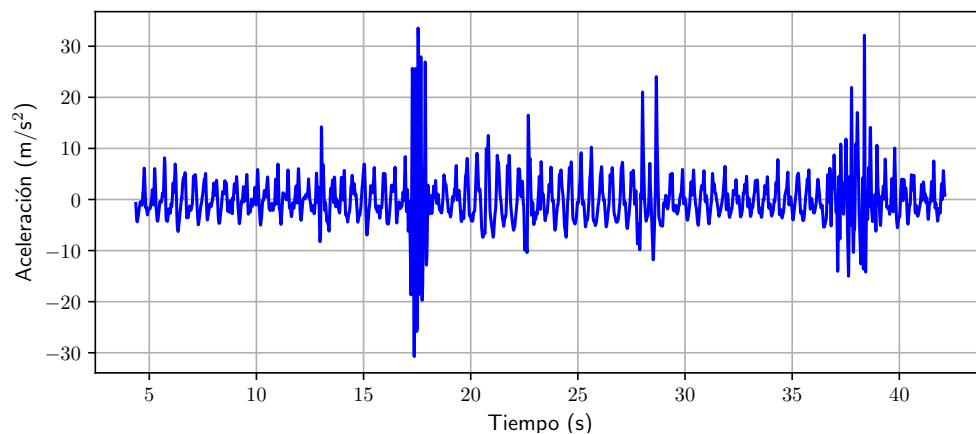


Figura 5.4. Señal obtenida caminando.

5.2. Resultados de la detección de anomalías

5.2.1. Señal sintética

Para obtener los parámetros para la identificación de anomalías, calibrando de manera manual con el widget, se obtiene que los valores útiles para L y N para construir la matriz de Hankel $\mathcal{H}_L(\Sigma_N)$, son $L = 150$, $N = 1200$ tolerancia = 0.13 y función lógica AND. En la Fig. 5.5 se observa que se identifican las regiones que se les agregó ruido como las anomalías en la serie de tiempo, que es lo que se quiere identificar para este ejemplo. Estos resultados se pueden generar con el código `example_synthetic_signal.py`.

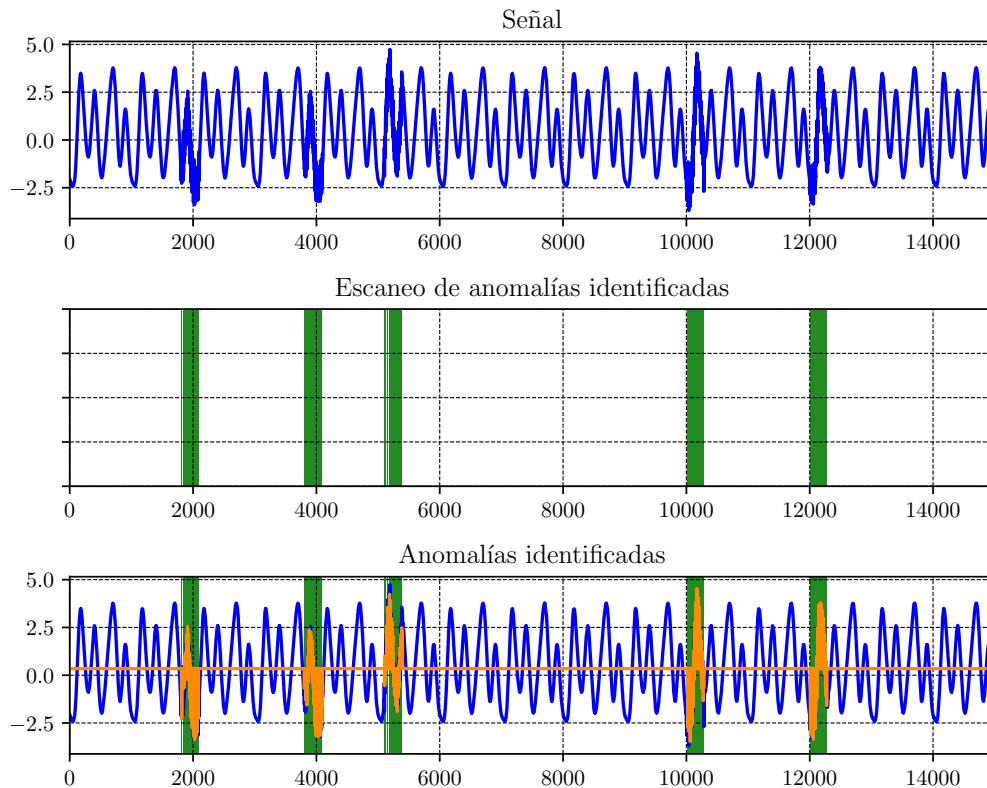


Figura 5.5. Señal sintética con las anomalías identificadas usando los valores de $L = 150$, $N = 1200$, tolerancia = 0.13 y función lógica AND.

Observación 5.1. Debido a que este es un ejemplo básico sin ruido, excepto en las partes que son anomalías, se pueden obtener varios valores para L , N y la tolerancia para todos los métodos y que sirva igual de bien que el ejemplo mostrado de la Fig. 5.5.

5.2.2. Máquina Vibradora

Experimentando con los valores de L y N para construir la matriz de Hankel $\mathcal{H}_L(\Sigma_N)$ y cantidad de puntos de entrenamiento a tomar, los valores que funcionaron mejor fueron $L = 75$ y $N = 1300$, tolerancia = 6.85 y función lógica OR. donde se calibró de manera manual usando el widget.

En la Fig. 5.6 se muestran los resultados obtenidos de usar el método de descomposición de valores propios hermítica, con una tolerancia de 1.06. Se detectan las tres veces que se aumentó de manera súbita la intensidad de la máquina con una mínima cantidad de falsos positivos. Este resultado puede ser generado con el código `example_signal_machine.py`.

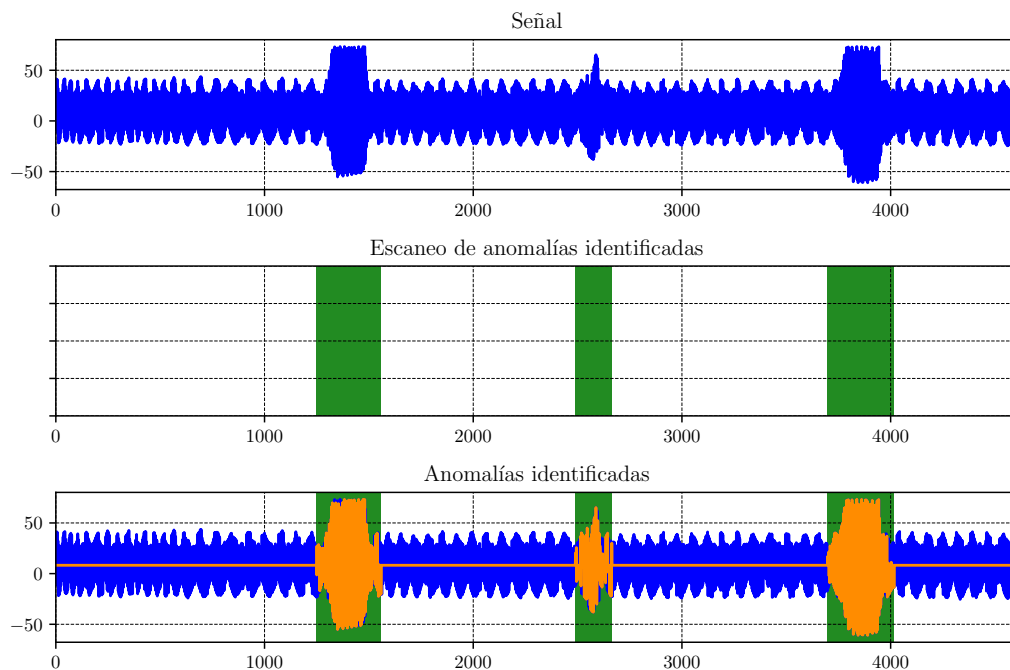


Figura 5.6. Señal de la máquina con las anomalías identificadas usando los valores de $L = 75$, $N = 1300$, tolerancia = 6.85 y función lógica OR.

5.2.3. Caminata

Se calibra de manera manual y usan los valores $L = 90$ y $N = 900$ para construir la matriz de Hankel $\mathcal{H}_L(\Sigma_N)$ y se usa una tolerancia = 0.19 y función lógica OR para encontrar los puntos anómalos. De la Fig. 5.7, se puede ver cómo el algoritmo identifica

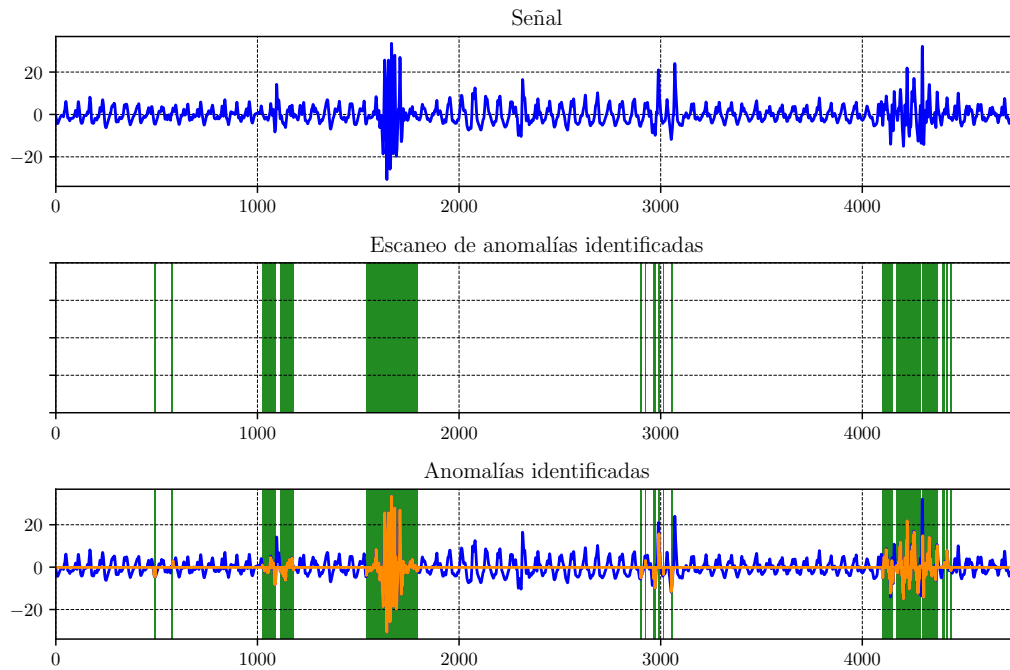


Figura 5.7. Señal de la caminata con las anomalías identificadas usando los valores de $L = 90$, $N = 900$, tolerancia = 0.19 y función lógica OR.

varios puntos donde no se mantiene una caminata regular y logra identificar varias de estas anomalías.

También se observa que el algoritmo no es únicamente dependiente del aumento de amplitud, ya que en los puntos que se muestran en la Fig. 5.7, se muestran puntos identificados como anomalías, y una anomalía de amplitud más grande no es detectada. Este resultado puede ser generado con el código `example_signal_walking.py`.

5.3. Selección de distintas tolerancias

Dependiendo del valor de la tolerancia escogida, el detector de anomalías va a ser más sensible. Un valor muy bajo de la tolerancia va a producir que varios puntos de la señal que no son anómalos sean mal clasificados como anomalías (falsos positivos). Por el contrario, si se pone una tolerancia muy alta, va a producir que puntos de la señal que son anómalos sean clasificados como no anómalos (falsos negativos). Se presentan los resultados de varias la tolerancia para las señales de la máquina y la caminata, con una tolerancia alta, adecuada y baja para cada señal.

5.3.1. Variando la tolerancia con la máquina vibradora

Los efectos de variar la tolerancia en la señal de la maquina vibradora se muestran en la Fig. 5.8. Se utilizan los mismos valores de $L = 75$, $N = 1300$ y función lógica OR de la Subsección 5.2.2. En la primera subfigura con una tolerancia de 12.40, se observa que solo detecta bien el tercer aumento en la vibración de la máquina. El primer aumento en la vibración detecta comportamiento anómalo, pero no por completo. El segundo aumento en la vibración no lo detecta del todo. En la segunda subfigura se usa una tolerancia de 2.45, la cual es adecuada para medir las anomalías. En la tercera subfigura, con una tolerancia de 0.85, se ve que detecta puntos de funcionamiento normal como anómalo y la tolerancia está demasiado baja para este sistema en particular. Este resultado puede ser generado con el código `different_tolerance_machine.py`

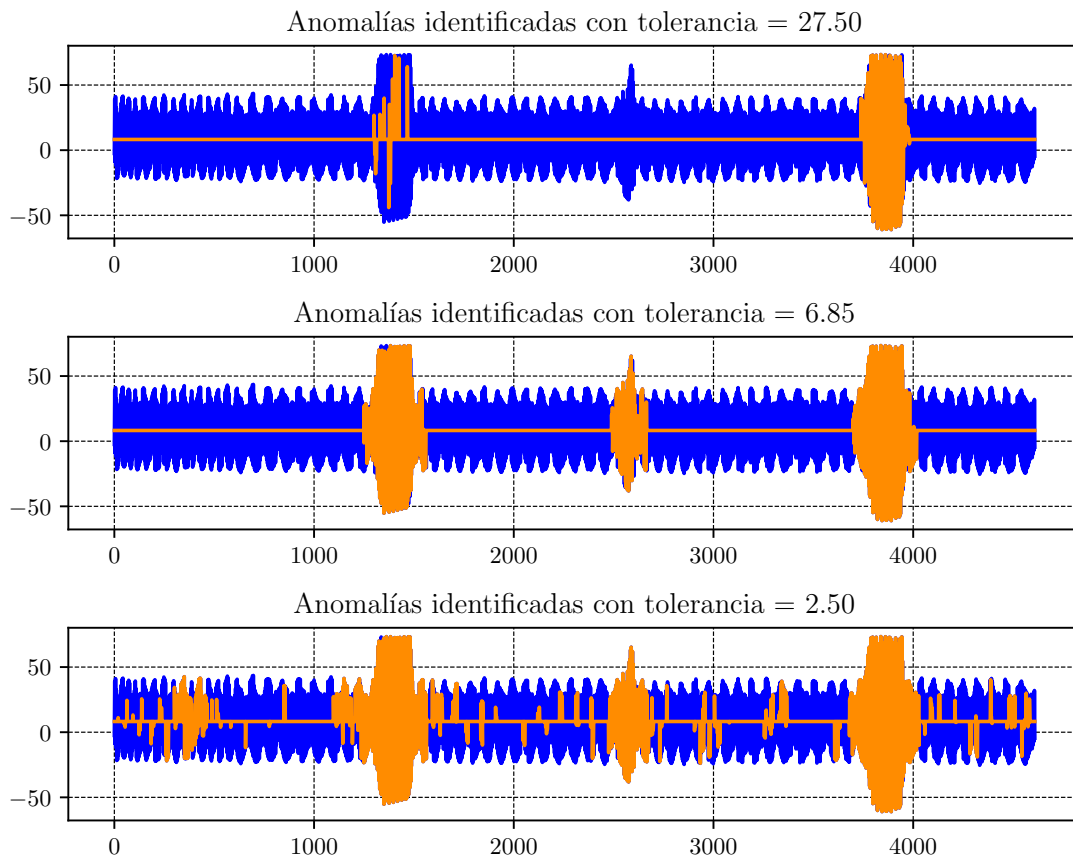


Figura 5.8. Anomalías detectadas variando la tolerancia sobre la máquina vibradora, con $L = 75$, $N = 1300$ y función lógica OR.

5.3.2. Variando la tolerancia con la caminata

Para ver el efecto de cambiar la tolerancia sobre la señal generada con la caminata, se usan los valores $L = 90$, $N = 900$ y función lógica OR que se encontraron en la Subsección. Los efectos de variar la tolerancia sobre la señal de la caminata se observan en la Fig. 5.9. Sucede el mismo resultado que con la señal de la maquina vibradora. Para una tolerancia alta de 1.35, se observa que el algoritmo solo detecta una de las anomalías en la señal de la caminata. Bajando la tolerancia a 0.19, se llega a a un valor más adecuado, que es el usado en la Fig. 5.7. Para una tolerancia baja de 0.06, gran parte de la señal está siendo detectada como anomalía, con una gran cantidad de falsos positivos. Este resultado puede ser generado con el código `different_tolerance_walking.py`.

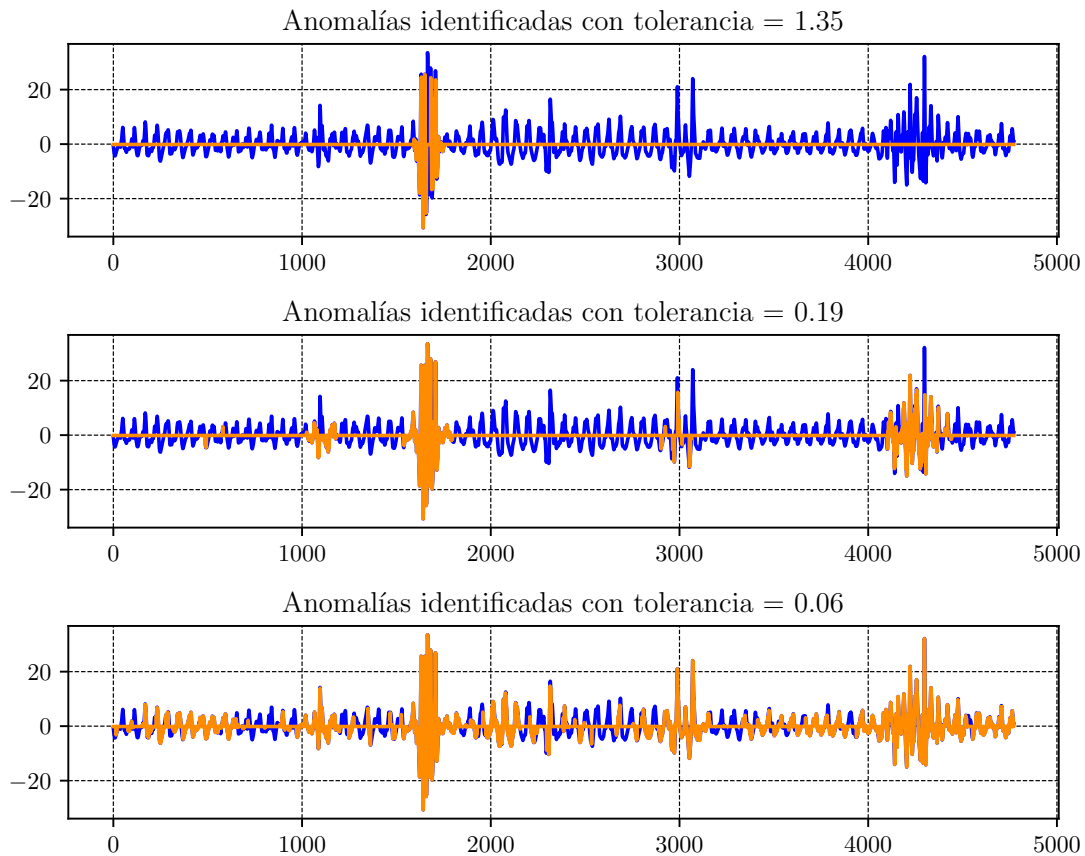


Figura 5.9. Anomalías detectadas variando la tolerancia sobre la caminata con valores $L = 900$, $N = 900$ y función lógica OR.

5.3.3. Comparación de tiempo cada método para obtener \mathbf{p}

En la Sección 4.6, se explica cómo todos los métodos tienen una complejidad similar. En esta Sección se va a medir de manera empírica el tiempo para comparar la rapidez de cada método para obtener el vector \mathbf{p} . Para los experimentos, se utilizan las mismas especificaciones de hardware y software de la Sección 5.1. Se comparan los métodos:

- (a) La descomposición SVD directamente sobre $\mathcal{H}_L(\Sigma_N)$.
- (b) La descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)^T$.
- (c) La descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$, de manera directa y aprovechando que la matriz es simétrica.
- (d) La descomposición en valores propios de $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ con una rutina numérica para matrices hermíticas.
- (e) IPMRQ sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$.

Sobre todos los métodos, se calcula únicamente el vector \mathbf{p} , sin incluir de la detección de la anomalías, ya que es un procedimiento aparte y en común para todos. Con esta medición de la velocidad, se verifica si vale la pena hacer la multiplicación de matrices sobre el cálculo directo sobre $\mathcal{H}_L(\Sigma_N)$. No se va a comparar la descomposición de valores propios normal, ya que la rutina numérica para matrices hermíticas va a ser mas rápida y, sin pérdida de generalidad, la matriz $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ siempre va a ser hermítica.

Para cada método y cada señal se corre el experimento 1000 veces y se mide el tiempo que toma cada corrida. Se grafican los diagramas de cajas y las densidades para comparar los tiempos.

Los resultados sobre cada método sobre la señal producida por la máquina y la señal de caminar se muestran en las Fig. 5.10, 5.11, 5.12 y 5.13. Las figuras se ordenan en base al valor de la mediana de las mediciones del tiempo que tarda cada método.

En las Fig. 5.11 y 5.13 se observa que el método más lento es obtener directamente la descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)$, seguido por obtener la descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)^T$ en ambas señales.

En ambas señales, ignorando el IPMRQ, se obtiene que el orden de rapidez, de menor a mayor tiempo tomado en mediana para ordenarlos, es usando la descomposición de valores propios Hermítica, la descomposición $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$, SVD con la rutina numérica para

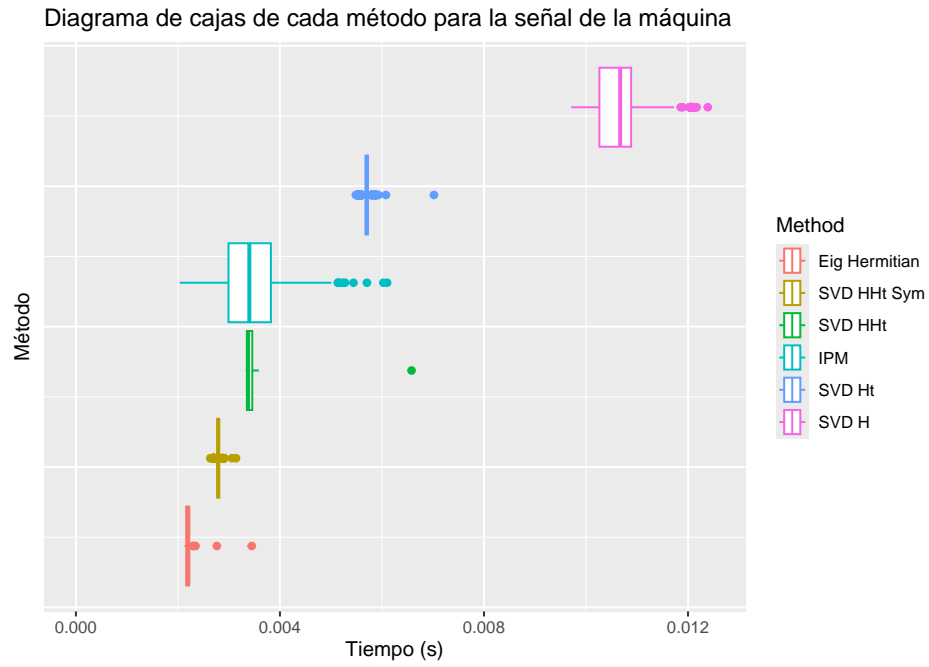


Figura 5.10. Diagramas de caja de cada método para la señal de la máquina.

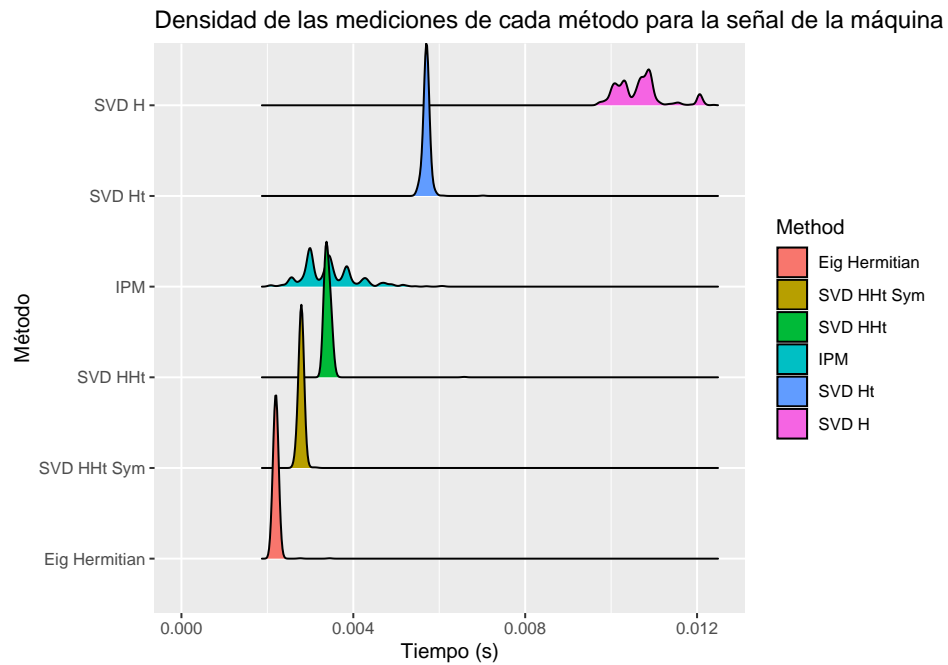


Figura 5.11. Densidades de cada método para la señal de la máquina.

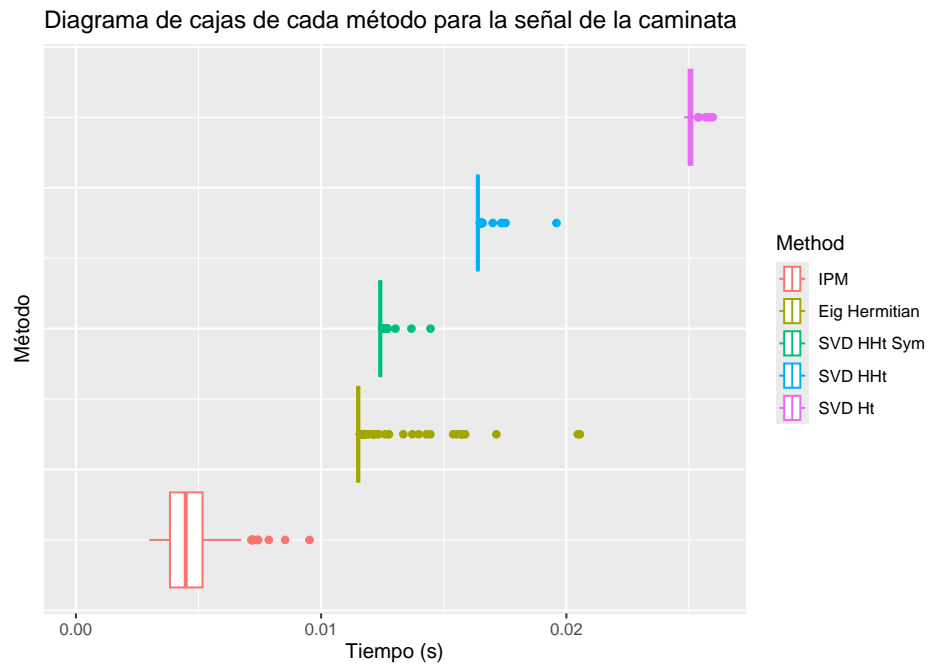


Figura 5.12. Diagramas de cada método para la señal de la caminata.

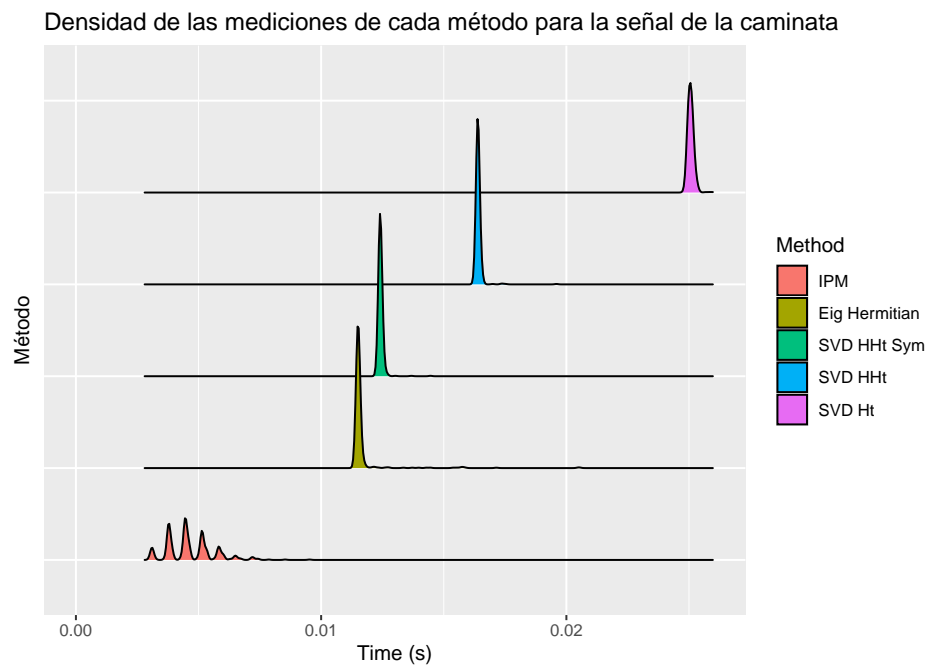


Figura 5.13. Densidades de cada método para la señal de la caminata.

aprovechar la estructura simétrica $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$, SVD con rutina numérica normal sobre $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$, SVD sobre $\mathcal{H}_L(\Sigma_N)^T$ y por último SVD sobre $\mathcal{H}_L(\Sigma_N)$.

En estos casos, los métodos que aprovechan la simetría de la matriz $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ son los que obtienen el vector \mathbf{p} de manera más rápida que los métodos que no lo aprovechan. De las mediciones del tiempo tomado, también se evidencia hacer el paso extra la multiplicación $\mathcal{H}_L(\Sigma_N)\mathcal{H}_L(\Sigma_N)^T$ sirve para ahorrar tiempo al usar las rutinas numéricas optimizadas para matrices simétricas.

El método de usar el IPMRQ para obtener el vector \mathbf{p} es mejor que usar la descomposición SVD sobre $\mathcal{H}_L(\Sigma_N)$ o $\mathcal{H}_L(\Sigma_N)^T$. Dependiendo de los datos, puede ser más rápido que los métodos optimizados para matrices simétricas como se muestra en las Fig. 5.13, o puede ser más lento como se muestra en la Fig. 5.11. Por lo que el IPMRQ es un método competitivo en comparación a los métodos optimizados de hacer descomposiciones.

En ambas Fig. 5.11 y 5.13 se observa como la densidad para el IPMRQ tiene varios máximos, debido al vector aleatorio que se usa para inicializar el algoritmo, en comparación a los demás métodos que tienen menos varianza debido que no tienen un factor aleatorio.

El código `measuring_times_methods.py` se utiliza para resultados de las Fig. 5.10, y 5.11, 5.12 y 5.13. El código `plot_times_statistics_ggplot.R` se usa para generar los gráficos.

5.3.4. Comparación de los vectores obtenidos para cada método.

En la Subsección 5.3.3 se compara tiempo que se toma para obtener el vector con cada uno de los métodos. En esta subsección se compara la diferencia entre los vectores obtenidos usando los distintos métodos.

Para comparar la diferencia entre los vectores obtenidos. Sea el vector $\mathbf{p} \in \mathbb{R}^N$ con componentes $\mathbf{p} = (p_1, p_2, \dots, p_N)$ y $p_1 \geq 0$, obtenido por uno de los métodos y el vector $\mathbf{q} \in \mathbb{R}^N$ con componentes $\mathbf{q} = (q_1, q_2, \dots, q_N)$ y $q_1 \geq 0$ obtenido por otro método. Se toman las direcciones de los vectores de manera que las primeras entradas tengan el mismo signo. Se mide la diferencia de los vectores con la función $\Delta(\mathbf{p}, \mathbf{q})$ dada por

$$\Delta(\mathbf{p}, \mathbf{q}) = \frac{1}{N} \sum_{i=1}^N |p_i - q_i|.$$

La diferencia entre los vectores obtenidos para distintos métodos excepto el IPMRQ dan una diferencia menor a $\Delta(\mathbf{p}, \mathbf{q}) < 10^{-10}$ para la señal de la máquina vibradora y la señal

de la caminata. La diferencia de los vectores si uno es obtenido usando el IPMRQ es del orden de magnitud $\Delta(\mathbf{p}, \mathbf{q}) \sim 10^{-1}$. Estos resultados se pueden replicar con los códigos `vector_difference_machine.py` y `vector_difference_walking.py`.

Capítulo 6

Conclusiones y futuro trabajo

6.1. Conclusiones

Basado en el trabajo teórico y aspectos computacionales, el método de proyección de subespacios presentado provee una herramienta interpretable y efectiva para la detección de anomalías de bajo costo computacional, memoria ocupada y tiempo de detección.

Estas características hacen que el método de proyección en subespacios sea fácilmente implementable en dispositivos de salud cotidiano como los relojes inteligentes con una limitada cantidad de recursos computacionales. Además, con la relativa simplicidad del método numérico del algoritmo de proyección sobre el subespacio, la técnica también resulta adecuada para esquemas colaborativos de inteligencia artificial, que puedan ser usados para aumentar la precisión de la detección, cuando más recursos computacionales se encuentren disponibles, como por ejemplo, conectar el reloj inteligente con acceso a computación en la nube.

Como futuros trabajos con el algoritmo, también se podrían implementar para esquemas de inteligencia artificial donde se combine la componente lineal y no lineal de un modelo [19], donde la parte lineal de los modelos puede ser computada usando los métodos discutidos en este documento.

6.2. Disponibilidad de los datos

Los programas de Python y datos para poder reproducir los resultados en este documento se encuentran disponibles en el repositorio de Github de la tesis. [21].

Bibliografía

- [1] Aggarwal, Charu C. *Linear Algebra and Optimization for Machine Learning*. Springer International Publishing, 2020.
- [2] Anderson, Edward, et al. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, 1999.
- [3] Björck, A., and Dahlquist, G. *Numerical Methods in Scientific Computing. Volume II*, Society for Industrial and Applied Mathematics, 2008.
- [4] Blázquez-García, Ane, et al. *A Review on Outlier/Anomaly Detection in Time Series Data*. ACM Computing Surveys, vol. 54, no. 3, 2021.
- [5] Brunton, Steven L., and Jose Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2022.
- [6] Burden, Richard L., et al. *Numerical Analysis*. Cengage Learning, 2011.
- [7] Byron, Frederick W., and Robert W. Fuller. *Mathematics of Classical and Quantum Physics*. Dover Publications, 1992.
- [8] Chandola, Varun, et al. *Anomaly detection: A survey*. ACM computing surveys, vol. 41 no. 3, 2009.
- [9] Cormen, Thomas H., et al. *Introduction to Algorithms*. MIT press, 2022.
- [10] Golub, Gene H, and Charles F Van. *Matrix Computations*. JHU Press, 2013.
- [11] Guillén, Helen., et al. *Álgebra Lineal: Un enfoque práctico y consiso*. euna, 2022
- [12] Johnston, Nathaniel. *Advanced Linear and Matrix Algebra*. Springer, 2021.
Meyer, C. D. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, 2023.

- [13] Meyer, Carl D. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, 2000.
- [14] Harris, Charles R., et al. *Array programming with NumPy*. Nature vol 585, no. 7825, 2020.
- [15] Staacks, S, et al. *Advanced tools for smartphone-based experiments: Phyphox*. Physics education, vol. 53, no. 4, 2018.
- [16] Trangenstein, John A. *Scientific Computing: Vol. II-Eigenvalues and Optimization*. Springer, 2018.
- [17] Trefethen, Lloyd N., and David Bau. *Numerical linear algebra*. Society for Industrial and Applied Mathematics, 1997.
- [18] Vides, Fredy. *Sparse system identification by low-rank approximation*. ArXiv (Cornell University), arXiv:2105.07522, 2021.
- [19] Vides, Fredy. *Computing Semilinear Sparse Models for Approximately Eventually Periodic Signals*. IFAC-PapersOnLine, vol. 55, no. 20, 2022.
- [20] Vides, Fredy, et al. *A Subspace Method for Time Series Anomaly Detection in Cyber-Physical Systems*. IFAC-PapersOnLine, vol. 55, no. 41, 2022.
- [21] Vides, Fredy, et al. *PAD: Projective anomaly detection toolset*. 2022. URL https://github.com/carlosjva/PAD_thesis.
- [22] Watkins, David S. *Fundamentals of Matrix Computations*. Wiley, 2004.