

Instituto Tecnológico de Costa Rica



TEC

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Programa de Maestría en Computación

Análisis de señales y sus correlaciones.

**Propuesta de Tesis sometida a consideración del
Departamento de Computación, para optar por el grado
de Magister Scientiae en Computación, con énfasis en
Ciencias de la Computación**

Juan Francisco Quesada Brizuela

PhD. Arnoldo Müller Molina

Junio, 2015

Resumen

La mejora continua, el mantenimiento preventivo y la predicción de fallas en máquinas son los principales focos de concentración de los ingenieros que las desarrollan hoy en día. Por medio del análisis de señales se pueden encontrar sus correlaciones y anomalías lo que facilitaría el análisis de las mismas. El crear una herramienta que facilite a los ingenieros resolver estos problemas es deseable.

Los análisis de similitud involucran mucha complejidad computacional, son pesados y, generalmente, toman muchas horas de trabajo. Es necesario crear un método que sea eficiente y se pueda utilizar efectivamente para realizar los análisis.

En este trabajo el concepto de '*agrupamiento*' consiste en crear grupos de objetos similares. Corresponde al término inglés '*cluster*' utilizado en '*cluster analysis*' o '*data analysis*'. El agrupamiento de objetos toma un papel central brindándonos la posibilidad de clasificar segmentos de datos en palabras para simplificar el procesamiento.

El proceso consiste en definir una función de distancia para poder comparar señales con efectividad. Las señales deben segmentarse y, utilizando un algoritmo de agrupamiento, se van a crear agrupamientos de las señales, una a la vez. Esto se efectúa para abstraer el comportamiento de una señal como si fuera una palabra. Organizando las palabras correspondientes de diferentes señales que ocurren en un mismo punto en el tiempo, podemos crear documentos de palabras de señales. Estos conjuntos de palabras de señales constituyen la entrada para un algoritmo de modelos de tópicos que permiten encontrar anomalías co-ocurrencias de las señales.

Por ejemplo si se realiza el análisis con los datos de un motor de un automóvil, podríamos encontrar temas tales como aceleración, ralentí o frenado. Estos serían determinados automáticamente por el sistema. Probablemente tendríamos varios ejemplos para cada uno de estos, que podrían ser cuando el motor está frío, cuando este llega a su temperatura de operación y demás variables. Cambiando las configuraciones del algoritmo se pueden crear más temas sobre el comportamiento de aceleración. Por otro lado, se puede determinar el momento en que algún componente empieza a trabajar inapropiadamente sin provocar una falla inmediata en el sistema, pero que a largo plazo podría crear fallas catastróficas.

Este proyecto de tesis implementa un sistema de descubrimiento patrones comunes y anomalías de señales. El proceso implica configuración de distintos parámetros tales y como el largo de la señal, la normalización de la señal, la selección y optimización de funciones de distancia y los respectivos rangos de distancia. Para los grupos de señales, parámetros de configuración de los algoritmos de descubrimiento de tópicos son también muy relevantes.

Palabras claves: análisis de señales, agrupamientos, modelos de temas, k vecinos más cercanos, normalización de señales.

Abstract

Continuous improvement, preventive maintenance and machine fault prediction are the main focus of engineers that develop such machines today. Through signal analysis, correlations and anomalies can be found which would facilitate their analysis. Creating a tool that enables engineers to solve these problems is desirable.

These analyzes involve a lot of computational complexity, are heavy and generally take many hours of work. It is necessary to create a method that is efficient and can be effectively used to perform the analysis.

The process consists of defining a distance function to compare signals effectively. The signals must be segmented and, by using a clustering algorithm, clusters of signals are going to be created, one at a time. This is intended to extract a behavior of a signal as a word. By organizing the corresponding words of different signals at the same time; we can create the documents which are the input to an algorithm of topic models that let us find anomalies and co-occurrences in the signals.

For example if you are going to perform the analysis with data from a car engine, we could find topics such as acceleration, braking or idling. These would be automatically determined by the system. We would probably have several examples for each of these, such as when the engine is cold, when it reaches its operating temperature and other variables. Changing the settings of the algorithm can create more issues about acceleration behavior. On the other hand, you can determine when a component begins working improperly without causing an immediate system failure, but that in the long term could create catastrophic failure.

This thesis project implements a system to discover common patterns and signal anomalies. The process involves setting various parameters such as the signal length, the signal normalization, the selection and optimization of distance functions and the respective distance ranges. For the signal groups, configuration parameters for the discovery algorithms of topics are also very relevant.

Key words: signals analysis, clusters, topic models, k nearest neighbors, signal normalization, distance function, metric spaces, pattern recognition, similarity search.

APROBACIÓN DE LA TESIS

“Análisis de datos de un motor de plasma”

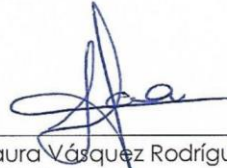
TRIBUNAL EXAMINADOR



PhD. Arnólido J. Müller-Molina
Profesor Asesor



PhD. José Enrique Araya Monge
Profesor Lector



MSc Laura Vásquez Rodríguez
Profesor Externo



Dr. Roberto Cortés Morales
Coordinador del Programa
de Maestría en Computación

Junio, 2015

Dedicatoria

A mis padres, mis hermanos y Laura

Gracias por todo el apoyo incondicional.

Agradecimientos

PhD. Arnoldo Müller es la persona más influyente que he conocido en los últimos años. Me ha transmitido conocimientos invaluable. Le agradezco mucho todas las oportunidades que me ha dado en estos últimos años. Gracias a Arnoldo he podido cumplir muchas metas personales y profesionales.

A la empresa simMachines por brindar algoritmos de última tecnología en agrupamientos y k vecinos más cercanos

Al Consejo Nacional para Investigaciones Científicas y Tecnológicas (CONICIT), que brindó apoyo financiero durante la Maestría.

A la empresa Ad Astra Rocket, ya que, gracias a un trabajo en conjunto con ellos, surgió la necesidad de crear el algoritmo que se desarrolló en esta tesis.

Índice

Resumen	1
Abstract.....	3
Dedicatoria.....	6
Agradecimientos	7
Índice.....	8
Índice de Figuras	11
Índice de Tablas	12
Índice de Fórmulas.....	12
Capítulo 1. Planteamiento del problema.....	12
1.1 Motivación	13
1.2 Definición del problema.....	15
1.3 Objetivos	16
1.3.1 Objetivo General	16
1.3.2 Objetivos Específicos	16
1.4 Hipótesis.....	17
1.5 Justificación del Proyecto	18
Capítulo 2 - Marco Teórico	20
2.1 Definiciones.....	20

2.1.1 Espacio Métrico	21
2.1.2 Consultas por similitud	21
2.1.3 Distancia L1	22
2.1.4 Distancia L2	22
2.1.5 Distancia LP	23
2.1.6 Dynamic Time Warping (DTW)	23
2.1.7 Distancia Levenshtein	24
2.1.8 Pendiente	24
2.1.9 Vector de pendientes.....	24
2.1.10 Libsim	24
2.2 Las funciones de distancia y similitud hoy en día	25
2.3 k vecinos más cercanos.....	26
2.4 Sistemas de agrupamientos.....	27
2.5 Topic Models.....	29
2.6 Componentes utilizados	30
2.6.1 Graphviz	30
Capítulo 3 Metodología	32
3.1 Primera Etapa: Preparación de los datos de prueba	33
3.2 Segunda Etapa: Crear estructura del prototipo y visualización.....	34

3.3 Tercera Etapa: Definir funciones de distancia	36
3.4 Cuarta Etapa – Agrupaciones de señales.....	36
3.5 Quinta etapa - Anomalías de señales	37
Capítulo 4 – Solución implementada	38
4.1 Modelo Computacional	38
4.2 Etapas del modelo.....	39
4.2.1 Seleccionar un conjunto de datos (<i>dataset</i>) de pruebas.....	39
4.2.2 Leer los datos y cargar las estructuras de los datos en memoria.....	40
4.2.3 Separar los datos por un intervalo deseado y con un traslape deseado.	41
4.2.4 Cargar los datos de la misma señal en el algoritmo de agrupamientos y ejecutarlo..	43
4.2.5 Imprimir los gráficos lineales que van a ser usados para la visualización	44
4.2.6 Crear la visualización de los mapas utilizando Graphviz	45
4.2.7 Normalizar las señales	46
4.2.8 Crear los documentos a partir de las señales normalizadas.	48
4.2.9 Ejecutar el algoritmo de los modelos de temas con los documentos.....	48
4.2.10 Crear la visualización de los resultados	49
4.2.11 Tecnologías usadas	50
Capítulo 5 – Experimentos.....	51
5.1 Funciones de distancia utilizadas.....	51

5.2 Rangos para el algoritmo de agrupamientos.....	51
5.3 Pendientes y valores.....	51
Capítulo 6 - Análisis de resultados.....	53
Capítulo 7 - Conclusiones.....	62
Bibliografía.....	65

Índice de Figuras

Figura 1 Modelo Computacional.....	38
Figura 2 Etapas del modelo.....	39
Figura 3 Gráfico de ejemplo de señales.....	40
Figura 4 Ejemplo de cortes transversales de las señales.....	41
Figura 5 Ejemplo de traslape en los cortes transversales.....	42
Figura 6 Ejemplo de árbol de carpetas.....	45
Figura 7 Ejemplo de estructura de Graphviz.....	46
Figura 8 Ejemplo Graphviz.....	46
Figura 9 Ejemplos de centros de clústeres.....	47
Figura 10 Ejemplo de señal normalizada.....	48
Figura 11 Ejemplo de resultado de un tema.....	49
Figura 12 Grafico del total de agrupamientos por señal.....	55
Figura 13 Percentiles del promedio de elementos.....	55
Figura 14 Vista del agrupamiento correspondiente a la señal 1.....	56

Figura 15 detalle de un agrupamiento de la región central	57
Figura 16 Agrupamiento de tamaño mediano con sus vecinos	58
Figura 17 Agrupamientos de elementos pequeños.....	59
Figura 18 Correlaciones frecuentes entre la señales.....	59
Figura 19 Correlaciones poco frecuentes entre las señales	60

Índice de Tablas

Tabla 1 Resultados resumidos (menos es mejor)	54
---	----

Índice de Fórmulas

Fórmula 1 L1.....	22
Fórmula 2 L2.....	22
Fórmula 3 LP	23
Fórmula 4 DTW - 1	23
Fórmula 5 DTW - 2	23
Fórmula 6 DTW - 3	23
Fórmula 7 Levenshtein.....	24

Capítulo 1. Planteamiento del problema

1.1 Motivación

Existe una gran cantidad de máquinas a nuestro alrededor. Muchas de estas máquinas son complejas y pasan completamente desapercibidas en nuestro día a día. Probablemente, si vive en Costa Rica, la energía eléctrica del país fue generada por una represa hidroeléctrica, la cual es hecha a la medida (no son producidas en serie, ya que todas tienen sus características específicas), y se necesita de expertos para comprender su funcionamiento. Existe gente entrenada con muchos años de experiencia para poder manejar estas piezas de ingeniería de manera exitosa.

Los motores de automóvil, las turbinas de un avión, son otros ejemplos de máquinas complejas que son producidas en serie. Estas normalmente son calibradas por el fabricante y, según las condiciones externas, tales como altura, temperatura del aire y demanda de potencia, entre otras, se calibran automáticamente para trabajar lo más eficientemente que sea posible. La búsqueda de la optimización en la configuración de estas máquinas es un objetivo primordial en la industria. Este objetivo se lleva a cabo mediante experimentación y medición de salidas de sensores. Múltiples experimentos son ejecutados y los resultados son calibrados y optimizados. Los fabricantes elaboran sus propias recetas para determinar cuándo es que un motor debe calibrarse de forma específica para trabajar lo más cercano a la configuración ideal. Los motores de automóvil cuentan con unos 100 años de investigación, por lo que es muy sencillo entender su comportamiento y saber cómo se van a desempeñar. Por su parte, las turbinas de avión llevan más de 50 años de desarrollo. En contraste, un motor de plasma es muy novedoso, y entonces, cualquier técnica que ayude en el desarrollo de los sistemas va a ser de mucha

ayuda para los científicos. Hasta el momento no existe un motor de plasma a nivel productivo, ya que todos se han creado con fines de investigación.

Muchos procesos en la industria se han visto beneficiados por la computación. Gracias a nuevas técnicas en computación se pueden llevar a cabo procesos eficientes al punto de ser rentables para la industria. Existen muchas oportunidades en la actualidad, las que, con técnicas tradicionales de computación, no podrían llegar a una optimización que las torne rentables para la industria. Es necesario el uso de técnicas innovadoras y combinaciones de estas técnicas para poder trabajar eficientemente y crear estos procesos rentables para la industria.

La herramienta propuesta en esta tesis busca analizar múltiples señales tomadas de sensores de manera simultánea para encontrar patrones y anomalías tanto a nivel de señal como en la manera que las señales se presentan en un mismo punto en el tiempo. Se busca poder generar un análisis más completo sobre los datos que son generados. El sistema automáticamente puede determinar cuándo se están presentando anomalías en el sistema. También puede encontrar los comportamientos más comunes que ocurren en el sistema. Estos dos tipos de análisis pueden ayudar a encontrar posibles problemas o determinar si algunos comportamientos son esperados.

1.2 Definición del problema

Gracias a un trabajo en conjunto con Ad-Astra Rocket Company (AARC) surge la idea de crear un algoritmo que pueda interpretar los datos que son generados. *Ad Astra Rocket Company es una compañía de ingeniería aeroespacial dedicada al desarrollo de tecnologías de propulsión avanzadas basadas en plasma. La compañía está desarrollando el Motor de Magnetoplasma de Impulso Específico Variable (VASIMR, por sus siglas en inglés) y sus tecnologías asociadas.*¹ Este motor dispone de muchos sensores y dispositivos que generan una gran cantidad de información. Toda esta información está siendo almacenada, pero no existe una técnica apropiada para estudiar dicha información. Inicialmente, se planea que con el uso de algoritmos de *K nearest neighbor, clustering* (agrupamiento), *topic models*, es posible hacer un análisis de los datos que son generados por el motor de plasma. La idea es generar una especie de tabla de analogías, una piedra roseta, para poder trabajar más sencillamente con la información que se genera del motor.

El análisis del motor de plasma va acompañado de un tipo de visualización Web donde se pueden ver los gráficos de las señales y toda la información generada. Se busca crear una herramienta con la que los expertos en el motor de plasma puedan visualizar los experimentos y puedan encontrar anomalías y comportamientos típicos de manera más sencilla.

¹ "Ad Astra Rocket." 2005. 24 May. 2015 <<http://www.adastrarocket.com/>>

1.3 Objetivos

1.3.1 Objetivo General

Crear un algoritmo que consuma y analice datos de señales, utilizando técnicas de *k nearest neighbor* y *topic models*, y presentar los resultados utilizando técnicas de agrupamientos y visualización.

1.3.2 Objetivos Específicos

- Definir un formato de archivo para el consumo de los datos.
- Buscar un conjunto de datos (*dataset*) de pruebas con un formato e información similar a la que se almacena en Ad-Astra Rocket.
- Buscar una función de distancia ligera y consistente para comparar señales que satisfaga criterios de densidad y similitud.
- Agrupar a través de un sistema de agrupamientos señales similares.
- Evaluar la calidad de los agrupamientos promediando las diferencias de su centro a los objetos.
- Utilizar una técnica de "*Topic Models*" para encontrar grupos de señales de diferentes sensores que co-ocurren al mismo tiempo. Esto para identificar patrones y anomalías.
- Implementar una visualización que permita a los expertos analizar la información de forma más sencilla.

1.4 Hipótesis

Utilizando las técnicas existentes para procesar grandes cantidades de datos, combinadas con técnicas para procesamiento de datos de señales, es posible crear un algoritmo que trabaja con diferentes señales que son originadas por distintos instrumentos. Esta herramienta puede ayudar a los científicos que desarrollan máquinas complejas. Se pueden encontrar correlaciones entre los datos, así como anomalías. La herramienta puede analizar distintas entradas con magnitudes y tipos de datos completamente diferentes para mostrar resultados complejos con toda la información que es suministrada. La segunda hipótesis es que para el análisis de señales propuesto, la función Euclidiana puede obtener agrupamientos compactos y de calidad y esto permite ahorrar en tiempo obteniendo alta calidad de resultados.

1.5 Justificación del Proyecto

El desarrollo de un motor de plasma o de cualquier máquina de magnitud considerable se mide en miles de millones de dólares. El costo de una prueba puede rondar los miles de dólares y la construcción de un prototipo puede representar millones de dólares.

El análisis simultáneo de varias señales o salidas de sensores tiene un sinnúmero de aplicaciones. Una represa hidroeléctrica es una máquina compleja que necesita de ingenieros capacitados para manejarla. Probablemente ya existan técnicas y demás procesos para mantener estas máquinas dentro de un rango optimizado. La complejidad para comprender estas máquinas puede reducirse mediante una herramienta de análisis de datos de señales que son generados por la represa.

En esta tesis de maestría se propone utilizar diversas técnicas tales como *K nearest neighbor* (KNN), *topic models*, agrupamientos y técnicas avanzadas de visualización. *K nearest neighbor* es una técnica que existe desde principios de la computación, pero por ser tan costosa en términos de computación no es posible usarla a gran escala. La empresa simMachines ha suministrado, en el contexto de esta investigación, una licencia académica de un algoritmo de *k*-NN muy rápido que permite obtener resultados eficientes. El algoritmo de *topic models* nos ayuda a encontrar segmentos en las señales que frecuentemente se repiten; utilizándolo de manera inversa, podemos encontrar cosas poco comunes que también se repiten o anomalías que se presentan en los datos. Vamos a normalizar las señales usando un algoritmo de agrupamientos, cuyo proceso se detallará más adelante. Finalmente, vamos a usar técnicas de visualizaciones para crear mapas de segmentos similares de las señales. Vamos a crear una

visualización web donde podremos ver cuán frecuente es un comportamiento o cuán anómalo es este comportamiento.

Se trabajará con una base de datos de pruebas, ya que los datos generados por la empresa AARC son confidenciales.

Normalmente, cuando se utilizan técnicas de análisis para señales, solo se utiliza una señal a la vez. Luego de una revisión de la literatura, no se ha encontrado análisis múltiples de señales con agrupamientos en varios niveles del proceso.

Hasta la fecha no se ha utilizado un motor de plasma que no sea con fines de pruebas. Los motores de plasma aún están en etapa de desarrollo y pruebas. Estas pruebas son limitadas por los recursos necesarios para ejecutarlas. Con esta tesis se busca suministrar a los científicos una herramienta robusta para que ellos puedan realizar un mejor análisis de la información que están generando. El objetivo es que los científicos puedan tener una mejor comprensión de las pruebas que están realizando. El proyecto busca combinar varias técnicas que, trabajadas de forma separada, son bastante buenas. A la hora de combinar estas técnicas se pueden tener diferentes puntos de vista útiles de los datos.

Capítulo 2 - Marco Teórico

2.1 Definiciones

En el siguiente capítulo describimos funciones de distancia que pueden ser utilizadas para procesar señales de un largo temporal fijo.

El concepto de distancia es básico en la experiencia humana, significa que dos objetos físicos o ideas están cercanos. Puede ser por distancia, un intervalo de tiempo, la brecha entre ellos y demás características. La métrica es el estándar para la medición entre estos dos objetos o ideas. [Deza, Deza 2009]

Podríamos tomar un platón de frutas que contiene limones, naranjas y sandias como ejemplo. Debido a que todas las frutas están juntas en el espacio métrico de distancia geográfica todas las frutas están muy relacionadas ya que todas se encuentran juntas en el platón. Si utilizamos el espacio métrico de peso podríamos ver que las naranjas y los limones están relacionados hasta cierto punto ya que poseen un peso similar. El peso de una sandía es evidentemente de más magnitud. Nuestra función de distancia para el primer caso de distancia geográfica podría ser una distancia euclidiana en un plano cartesiano. Para el segundo caso de peso podría ser el valor absoluto de una resta entre los valores de peso en este caso utilizaríamos una función de distancia similar a la L1 que va a ser definida formalmente más adelante. Como se puede apreciar con estos simples ejemplos las funciones de distancia y el espacio métrico sobre el que estas trabajan cumplen un rol importante en cómo se van a relacionar los datos. El tipo de dato que se quiere relacionar también es importante a la hora de definir una función de distancia ya que el tipo de dato va de la mano con la efectividad de la función de distancia para ese espacio métrico en específico. Se van a definir cinco funciones de

distancia que fueron seleccionadas. También se van a utilizar diferentes tipos de datos, los vectores crudos con los datos y el vector de pendientes correspondiente.

Las funciones de distancia no son nuevas y están muy presentes en muchas áreas desde la geometría hasta los gráficos por computadoras. Estas se utilizan para definir métricas y cuantificar las distancias entre dos objetos. [Deza, Deza 2009] La mayoría de estudios formales recientes incluyen directa o indirectamente la definición de una función de distancia y espacio métrico. Esto porque son muy útiles a la hora de hacer comparaciones, clasificaciones y búsquedas. Al poder cuantificar las distancias entre dos objetos los investigadores tienen la posibilidad de controlar mejor sus experimentos y extraer valor.

2.1.1 Espacio Métrico

Sea $M = (D, d)$ el espacio métrico para un conjunto X , $d : X \times X \rightarrow \mathbb{R}$ es llamada distancia en X si para todo $x, y \in X$ tenemos que:

1. $\forall x, y \in D, d(x, y) \geq 0$
2. $\forall x, y \in D, d(x, y) = d(y, x)$
3. $\forall x, y \in D, d(x, y) = 0 \Leftrightarrow x = y$
4. $\forall x, y, p \in D, d(x, y) \leq d(x, p) + d(p, y)$

[Deza, Deza 2009]

2.1.2 Consultas por similitud

Dada una colección $X \subseteq D$, se definen tres tipos de consultas por similitud.

- Consulta de k vecinos más cercanos (k -NN) nos da el k vecino más cercano al objeto consultado q . Formalmente, el conjunto $R \subseteq X$, tal que $|R| = k$, para cualquier $x \in R$ y cualquier $y \in X - R : d(q, x) \leq d(q, y)$.
- Consulta de rango: Dado $q \in D$ y r consulta todos los objetos con distancia r , que es el conjunto $\{x \in X | d(x, q) \leq r\}$
- Consulta de rango k -NN, sería la intersección de las 2 consultas anteriores.

[Hjaltason, Samet 2003]

2.1.3 Distancia L1

A la distancia L1 también se le llama “taxicab” se define como, sean (x, y) vectores $x = (x_1, x_2, \dots, x_n)$ y $y = (y_1, y_2, \dots, y_n)$ entonces:

$$d_1(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Fórmula 1 L1

[Deza, Deza 2009]

2.1.4 Distancia L2

La distancia L2 también es conocida como la distancia euclidiana, sean (x, y) vectores $x = (x_1, x_2, \dots, x_n)$ y $y = (y_1, y_2, \dots, y_n)$ entonces:

$$d_2(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Fórmula 2 L2

[Deza, Deza 2009]

2.1.5 Distancia LP

Sean (x, y) vectores $x = (x_1, x_2, \dots, x_n)$ y $y = (y_1, y_2, \dots, y_n)$ entonces:

$$d_p(x, y) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$$

Fórmula 3 LP

[Deza, Deza 2009]

2.1.6 Dynamic Time Warping (DTW)

Suponga que tiene dos vectores x de largo n y y de largo m , entonces

$$x = (x_1, x_2, \dots, x_n)$$

$$y = (y_1, y_2, \dots, y_m)$$

Fórmula 4 DTW - 1

Para alinear estos dos vectores usando DTW se debe crear una matriz de n por m donde el elemento (i, j) corresponde a $d(x_i, y_j) = (x_i - y_j)^2$; entonces tenemos que:

$$DTW(x, y) = \min \left\{ \sqrt{\sum_{k=1}^K w_k} \right\}$$

Fórmula 5 DTW - 2

Donde w_k es el elemento $(i, j)_k$ de la matriz definida por

$$\gamma(i, j) = d(x_i, y_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$$

Fórmula 6 DTW - 3

Para el elemento $d(i, j)$ que corresponde a la distancia en la celda actual, $\gamma(i, j)$ es la distancia acumulativa de $d(i, j)$ y la distancia mínima de las celdas adyacentes. [Ratanamahatana, Keogh

2004]

2.1.7 Distancia Levenshtein

Dadas dos cadenas x, y la distancia Levenshtein está dada por $lev_{x,y}(|x|, |y|)$ donde

$$lev_{x,y}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} lev_{x,y}(i-1,j) + 1 \\ lev_{x,y}(i,j-1) + 1 \\ lev_{x,y}(i-1,j-1) + 1_{(x_i \neq y_j)} \end{cases}, & \min(i,j) \neq 0 \end{cases}$$

Fórmula 7 Levenshtein

[Deza, Deza 2009]

2.1.8 Pendiente

Sean dos puntos $a = (x_1, y_1)$, $b = (x_2, y_2)$ la pendiente de estos puntos es definida por:

$$p(a,b) = \frac{y_2 - y_1}{x_2 - x_1}$$

2.1.9 Vector de pendientes

Sea x un vector $x = (x_1, x_2, \dots, x_n)$ entonces el vector de pendientes estaría definido por

$$P_x = (p((x_1, 1), (x_2, 1)), p((x_2, 1), (x_3, 1)), \dots, p((x_{n-2}, 1), (x_{n-1}, 1)))$$

2.1.10 Libsim

Libsim es un diccionario de funciones de distancia creado por simMachines. Contiene alrededor de 100 funciones de distancia, de las cuales varias pueden ser posibles candidatas para trabajar para esta tesis.

2.2 Las funciones de distancia y similitud hoy en día

Existen muchas técnicas de “*Dynamic Time Warping*” que se utilizan para hacer comparaciones de secuencias que cambian con el tiempo o la velocidad. Estas técnicas buscan efectuar comparaciones similares a las que vamos a estar realizando. Un ejemplo puede ser el descrito en el trabajo [Berndt, Clifford 1994], en el cual los autores proponen una de las primeras técnicas para realizar comparaciones entre secuencias.

En el trabajo de [Yi, Jagadish, Faloutsos 1998] se comentan trabajos previos realizados sobre “*Time Warping*”, utilizando la distancia euclidiana y unas variaciones como la distancia L_p . Los autores también proponen otras técnicas mucho más rápidas para la similitud de señales.

En el trabajo de [Ruiz 1986] se habla sobre cómo efectuar la similitud en un tiempo constante. Por su parte, el [Youssef, Abdel-Galil, El-Saadany, Salama 2004] nos relata sobre técnicas de similitud a la hora de aplicarlas a señales. En el contexto de estas técnicas podemos encontrar algoritmos de DTW, “*vector quantization*” y “*fast match*”. En este proyecto nos vamos a enfocar en similitud y vamos a probar varias funciones de distancia para encontrar la que produzca mejores resultados.

En el paper “*Searching in metric spaces*” [Chávez, Navarro, Baeza-Yates, Marroquín 2001], los autores nos relatan el problema que existe para buscar elementos de un conjunto que se encuentra cerca de una consulta dada. Este problema tiene un gran número de aplicaciones en la rama de las ciencias de la computación. Se puede encontrar desde el

reconocimiento de patrones para crear información textual, hasta la extracción de información de datos multimedia. Se enfocan en el caso más general donde el criterio de similitud define un espacio métrico. Los autores afirman que existen muchas propuestas en diversas áreas para resolver estos problemas. Debido a esto, las mismas ideas han sido reconcebidas varias veces y en muy diferentes presentaciones, pero aun así terminan siendo los mismos enfoques. Los autores concluyen que han creado una estructura unificada que aclara más las estrategias actuales desde un mismo punto de vista.

2.3 k vecinos más cercanos

Esta es una técnica que existe desde hace ya varios años, pero que tiene un costo computacional muy elevado. Existen muchos algoritmos conocidos para efectuar este tipo de búsquedas. En los libros de [Samet 2006] y [Zezula, Amato, Dohnal, Batko 2006] se discute ampliamente sobre el tema de la búsqueda por similitud.

Existe el algoritmo de LSH (Locality Sensitive Hashing) [Gionis, Indyk, Motwani 1999], por sus siglas en inglés. Este es un algoritmo de búsqueda por similitud que tiene buenas prestaciones.

El algoritmo seleccionado para realizar la investigación fue amablemente proporcionado por simMachines. Su nombre es R01 y es el algoritmo con mejores prestaciones. [Müller-Molina 2012]

Existe un estudio sobre la forma para atacar el problema de la alta dimensionalidad. Los autores del artículo [Houle, Kriegel, Kröger, Schubert, Zimek 2010] exponen que el rendimiento en las medidas de similitud para la búsqueda de indexación y aplicación de minería de datos

tiende a degradarse rápidamente a medida que aumenta la dimensionalidad de los datos. Los efectos de la llamada maldición de dimensionalidad han sido estudiados por investigadores para conjuntos de datos genéticos.

En este trabajo se estudian los efectos de ese fenómeno en distintas medidas de similitud de datos distribuidos multiplicados. Se evalúa el desempeño en particular de las medidas compartidas de similitud vecinas, las cuales son medidas de similitud secundarias.

Basados en la clasificación de los objetos inducidos por cierta distancia primaria, se encontró que las medidas de similitud pueden dar lugar a un mayor rendimiento estable que sus medidas de distancia primaria.

2.4 Sistemas de agrupamientos

Un agrupamiento se define como un subconjunto de elementos que tienen una o más características en común.

En el trabajo sobre *“Fast algorithms for projected clustering”* [Aggarwal, Wolf, Yu, Procopiuc, Park 1999], los autores hablan sobre los problemas de agrupamientos que son muy familiares en las literaturas de bases de datos para la infinidad de aplicación en los problemas, tales como la clasificación, segmentación y análisis de tendencias de clientes.

Puede que resulte imposible encontrar un subconjunto de dimensiones para todos los grupos. Todos los algoritmos conocidos tienden a descomponerse en espacios dimensionales elevados. En estos espacios elevados no todas las dimensiones pueden ser de interés para un grupo determinado. Una manera de abordar esto consiste en acumular las dimensiones relacionadas y encontrar grupos en el sub espacio correspondiente. En nuestro proyecto vamos

a efectuar pruebas con distintas funciones de distancia a la hora de crear los agrupamientos. Se van a evaluar estos agrupamientos con el objetivo de que los grupos sean de mejor calidad.

Hay algoritmos de selección de función que intentan lograrlo, aunque la única debilidad de ese enfoque radica en la dimensión de aplicaciones de minería de datos de diferentes conjuntos y de puntos que se pueden agrupar mejor para distintos subconjuntos de dimensiones.

Los autores del artículo sobre *“Exploring Expression Data: Identification and Analysis of Coexpressed Genes”* [Heyer, Kruglyak, Yooseph 1999] indican que es necesario realizar procedimientos de análisis para extraer información útil partiendo de la gran cantidad de datos de expresión genética que se encuentra disponible. En este trabajo se describe un conjunto de herramientas de análisis y su aplicación en el ciclo celular.

Los componentes del enfoque consisten en la medida de similitud que reduce el número de falsos positivos.

Un nuevo algoritmo de agrupamiento que está diseñado específicamente para agrupar los patrones de expresión de genes, y un sistema interactivo de agrupamiento que permite comentarios de los usuarios y la validación.

Utilización de agrupaciones generadas por el algoritmo para resumir la expresión de todo genoma e iniciar la agrupación supervisada de genes en grupos biológicamente significativos.

Por otro lado los autores del artículo *“Automatic Subspace Clustering of High Dimensional Data”* [Agrawal, Gehrke, Gunopulos, Raghavan 2005] hablan sobre las aplicaciones de minería de datos, ponen requisitos especiales en los algoritmos de agrupación, conteniendo

la capacidad de encontrar agrupamientos en sub espacios de datos de altas dimensiones, escalabilidad, comprensión de los usuarios finales de los resultados e insensibilidad a la orden de entrada de registros.

Se presenta CLIQUE, el cual es un algoritmo de agrupamiento que satisface los requisitos anteriormente mencionados, el cual funciona para identificar agrupamientos en sub espacios de dimensionalidad máxima. Genera descripciones en forma DNF, que se reduce al mínimo para facilitar la comprensión.

Produce resultados idénticos, sin importar el orden en que se presenten los registros de entrada, y no suponen una forma matemática específica para la distribución de datos. CLIQUE encuentra agrupamientos eficientemente en grandes conjuntos de datos dimensionales elevados.

En el trabajo sobre agrupamientos de información de muchas dimensiones [Kriegel, Kröger, Zimek 2009], se expone como un área productiva en la búsqueda de minería de datos el sub espacio de agrupamiento y los problemas en relación con la gran cantidad de soluciones propuestas.

2.5 Topic Models

La técnica de "*Topic Models*" es un modelo estadístico para descubrir los temas abstractos en una colección de documentos. Un tema se define cuando varias palabras se repiten en varios documentos. Si utilizamos el algoritmo en periódicos y hacemos que cada artículo del periódico sea un documento, el algoritmo podría crear temas como deportes, sucesos, etc.

“*Topic Models*” se puede definir también como un algoritmo que descubre los temas principales en una colección no estructurada de objetos [Blei, 2002].

Por ejemplo, "perro" y "hueso" van a estar presentes con mayor frecuencia en un documento acerca de caninos, mientras que "gato" y "miau" aparecerán más frecuentemente en los documentos relacionados con los felinos, y "él" y "es" aparecerán de igual manera en los dos documentos. Por consiguiente, un documento se refiere a varios temas en distintos porcentajes, así que si el 10% de información es sobre los gatos y el 90% sobre perros, probablemente habría 9 veces más palabras “perro” que “gato”.

Los modelos de temas se van a utilizar para poder encontrar temas entre la información que se va a analizar. En nuestro caso, si aplicáramos el algoritmo a un motor de automóvil, un ejemplo de tema que se quisiera encontrar con este algoritmo sería aceleración o frenado. Construyendo correctamente los documentos para que sean consumidos por el sistema se podrán encontrar estos temas de forma automática.

2.6 Componentes utilizados

2.6.1 Graphviz

Este consiste en un sistema para la generación automática de gráficos, diagramas y redes abstractas [Ellson, Gansner, Koutsofios, North, Woodhull 2002]. Cuenta con una serie de aplicaciones en la creación de redes, bioinformática, bases de datos y diseño de páginas web y en interfaces visuales para dominios técnicos.

Graphviz utiliza descripciones de gráficos en un lenguaje de texto muy simple y hace diagramas en formatos útiles, tales como imágenes, SVG para páginas web, PDF, PostScript

para su inclusión en documentos o para mostrar en un navegador gráfico interactivo. Contiene muchas características como opciones de diseños, fuentes, colores, diseños de nodos de tabla, estilos de línea, hipervínculos y formas personalizadas.

Capítulo 3 Metodología

El objetivo de esta tesis de maestría es desarrollar un algoritmo que pueda hacer comparaciones simultáneas de señales de diferentes fuentes de sensores o instrumentos y compararlas y analizarlas para una mejor comprensión de cómo está trabajando el dispositivo que se monitorea. De igual manera, se intenta buscar patrones de señales que se presentan simultáneamente y también patrones que rara vez ocurren, por lo que se pueden considerar como anomalías. En esta tesis también se intenta encontrar una forma de hacer agrupamientos de esas señales para que los científicos puedan analizar más fácilmente los datos que están obteniendo de los instrumentos.

La idea es desarrollar software en lapsos cortos de iteraciones de 1 - 2 semanas. Cada iteración incluirá: planificación, análisis de requisitos, diseño, codificación, revisión y documentación (breve). Al ser un proyecto de investigación y experimentación, es más importante efectuar pruebas de concepto rápidas para poder validar teorías que se van generando. Luego, a medida que se van haciendo las pruebas necesarias, se pueden ir descartando ciertas técnicas.

Existe también una parte de experimentación en la que se llevan a cabo varias corridas del sistema con diferentes valores para intentar encontrar los valores óptimos para trabajar. Se deben efectuar diferentes series de experimentos a la hora de trabajar con las funciones de distancia y con los agrupamientos. Muchas de las técnicas que se van a utilizar deberán someterse a un proceso de experimentación donde se busca contar con la mejor combinación de parámetros para poder obtener las mejores combinaciones de agrupamientos.

Para validar el algoritmo se creará un prototipo que trabajará con una base de datos de pruebas. Este prototipo se implementará en Java. El prototipo debe leer los datos de pruebas y tener como salidas una visualización de agrupamientos de señales y una página web donde se muestren las anomalías o similitudes que presentan los datos.

Se han definido cinco etapas en las cuales se va a trabajar un tema específico. A continuación se detallan estas etapas.

3.1 Primera Etapa: Preparación de los datos de prueba

La información generada por Ad Astra Rocket sobre las pruebas del motor de plasma es confidencial. Son datos propietarios de Ad Astra Rocket y no pueden ser usados para pruebas de este algoritmo.

Es necesario encontrar una base de datos de pruebas que cumpla con los requisitos para poder efectuar pruebas con el prototipo. Estos datos de prueba tienen que ser suficientes para poder encontrar similitudes entre los datos, pero tampoco puede ser una cantidad excesiva ya que entonces las pruebas y el trabajo con los datos podrían llegar a tomar muchos recursos de hardware, lo que volvería el proceso lento y tedioso. Si se llegara a determinar que los datos son excesivos, se puede trabajar simplemente con un subconjunto de estos datos. No es necesario que estos datos de prueba contengan información similar a los datos reales, puesto que las calibraciones con los datos reales probablemente van a ser necesarias. Esto se debe a que la probabilidad de encontrar unos datos de prueba que sean extraídos de un mismo dispositivo y que contengan instrumentos y sensores que generen información similar es sumamente baja. Lo importante es buscar datos que tengan un formato similar.

Nos vamos a concentrar en buscar datos que simplemente estén asociados a la misma fuente pero que sean de diferentes sensores. Esto debería ser suficiente para poder generar un esquema de calibración de los datos que podría usarse en un futuro con datos reales de un motor de plasma. Los datos de este tipo deberían ser suficientes para poder ejecutar todo el algoritmo sin tener problemas.

Es importante tomar en cuenta el formato en el cual se van a descargar los datos, ya que esto puede influir en la toma de decisiones con respecto a cuál base de datos se elegirá para trabajar.

El siguiente paso sería crear un lector de los datos para poder cargarlos en el sistema y poder trabajarlos. Este lector puede al mismo tiempo separar los datos y alistarlos en la forma en que vayan a ser necesarios para su posterior procesamiento.

3.2 Segunda Etapa: Crear estructura del prototipo y visualización.

En la segunda etapa del proceso se va a crear la estructura del prototipo, una especie de chasis donde se van a montar las visualizaciones y todos los algoritmos que se utilizarán en el proceso. Las visualizaciones son las partes donde el prototipo establecerá una interacción con el usuario. Nos ayudará a validar si efectivamente la función de distancia que estamos utilizando va a trabajar bien.

Esta etapa nos suministrará una herramienta visual que puede ser utilizada por los expertos para brindarnos información sobre cómo está trabajando el prototipo. Representará la forma de validar si efectivamente la función de distancia está haciendo su trabajo y la forma gráfica de representar los agrupamientos. Para simplificar la construcción se va a construir un

flujo de la forma en que se crean los agrupamientos y otro flujo para mostrar las anomalías. Vamos a utilizar las funciones de distancia L1, L2, LP, DTW y Levenshtein. La configuración de los algoritmos será la básica por defecto para todos los algoritmos que vamos a utilizar. La idea es poder contar con una base sólida sobre la cual podemos trabajar y realizar experimentos. Se implementarán dos tipos diferentes de visualizaciones: en primer lugar, una visualización sencilla con un mapa de los agrupamientos sobre la forma en que se agrupan las señales, y la segunda parte consistiría en una página web que contiene las anomalías que se presentan en el sistema. Es necesario crear también la estructura de software que va a contener todos los procesos mediante los cuales se van a trabajar los datos. Esta estructura del software será como el chasis que va a sostener todos los componentes del prototipo.

La visualización de los agrupamientos se llevará a cabo mediante un mapa de agrupamientos donde cada señal se va a desplegar con una pequeña imagen. La salida del prototipo consistirá en un mapa de agrupamientos donde los países son conjuntos de elementos y su centro es el elemento más común o estándar del agrupamiento. Existirán diferentes países. Estas visualizaciones pueden llegar a generar archivos de salida muy grandes difíciles de manejar. Debido a que van a utilizarse para observar visualmente y validar de forma subjetiva, no va a ser necesario desplegar todos los elementos de los agrupamientos; puede ser que solo se tenga un conjunto. También se contará con un filtro para desplegar solamente agrupamientos de más de 5 elementos.

3.3 Tercera Etapa: Definir funciones de distancia

Gran parte del éxito de este prototipo radica en poder comparar efectivamente las señales. Buscar una función de distancia rápida y efectiva es la clave para poder realizar comparaciones de señales.

Es necesario que esta función de distancia sea rápida, ya que los algoritmos que van a utilizar esta función de distancia van a usarla en varias ocasiones, y si se cuenta con una gran cantidad de datos, el tiempo que tomará crear la visualización puede llegar a ser muy alto, incluso días y hasta semanas. Es necesario poder correr el algoritmo en un tiempo razonable. Es posible que los recursos de hardware que puede llegar a requerir el sistema sean muchos; de momento, todos los datos se están cargando en memoria por facilidad y debido a que no es un problema para los datos de prueba.

La función de distancia tiene que ser efectiva según el contexto en el que se esté trabajando. Se pueden escoger funciones que trabajan por magnitud, por la forma del gráfico de esta, combinadas o por deformaciones.

3.4 Cuarta Etapa – Agrupaciones de señales.

Se va a utilizar un algoritmo que proporcionó amablemente simMachines para esta tesis. Este algoritmo de agrupamiento nos va a proveer los agrupamientos necesarios para poder encontrar resultados en muy poco tiempo.

Para esta tesis vamos a utilizar varias funciones de distancia L1, L2, LP, DTW y Levenshtein. Vamos a comparar los agrupamientos que estas crean. Se puede comparar la cantidad de objetos en los agrupamientos, así como qué tan unidos están los elementos de

estos agrupamientos. Se van a tomar tiempos de la construcción de los agrupamientos que también van a ser usados para determinar la calidad de estos resultados. Se evaluarán todos estos aspectos para determinar cuál es la mejor función de distancia. Al mismo tiempo, se efectuarán varias pruebas con configuraciones diferentes para el sistema de agrupamientos y para determinar cuáles resultados son mejores.

3.5 Quinta etapa - Anomalías de señales

En esta etapa se trabajará para encontrar anomalías en diferentes señales. Se buscarán cosas anómalas que estén sucediendo simultáneamente en las señales.

Esta es la parte más extensa del prototipo. Una vez que se tienen creados los agrupamientos, se procederá a cargar los centros de estos agrupamientos en un sistema llamado Ramiel (k -nn) que simMachines facilitó amablemente para la creación de este prototipo. Ramiel es un motor de similitud. Lo que buscamos es cargar estos centros que fueron creados en la etapa cuatro para entrenar el algoritmo. Estos centros son la señal más representativa de cada agrupamiento, por lo cual se va a utilizar como una palabra base para normalizar las señales. Luego se irá consultando cada segmento de cada señal para lograr normalizar esta señal con los centros correspondientes.

Una vez que tenemos la señal normalizada, utilizamos el algoritmo de modelos de temas para que nos ayude a encontrar cosas que ocurren comúnmente en las señales. Simultáneamente, al final del algoritmo se desplegarán cosas que suceden también frecuentemente, pero con una baja frecuencia.

Capítulo 4 – Solución implementada

El prototipo del algoritmo fue implementado en Java. Utiliza varias librerías y componentes externos. A continuación se brinda una descripción más detallada de todas las partes del algoritmo.

4.1 Modelo Computacional

El sistema cuenta con una capa de acceso a datos, lógica y presentación.

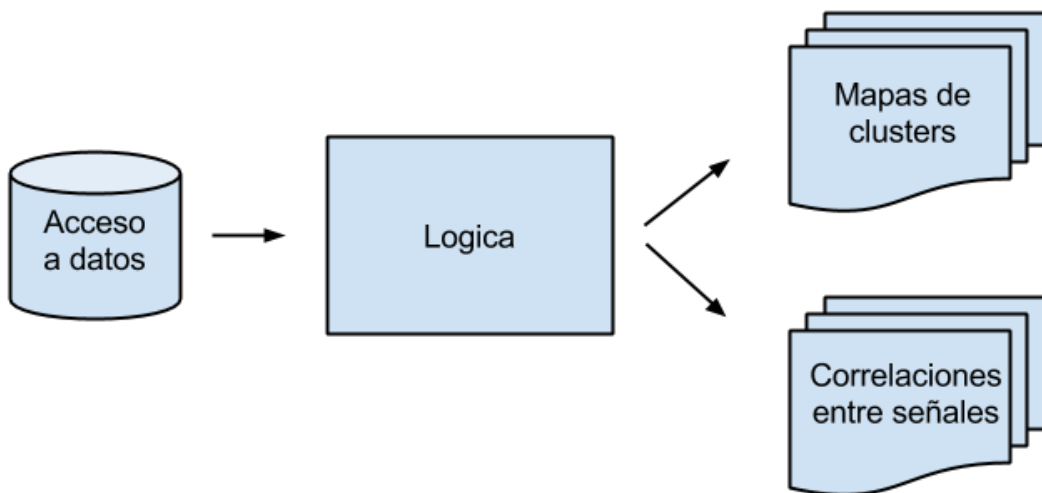


Figura 1 Modelo Computacional

La capa de acceso a datos realiza la lectura de los datos de prueba. Esta los alista para que posteriormente puedan ser consumidos por los otros sistemas.

La capa lógica es el fuerte del prototipo; cuenta con todos los algoritmos que se van a utilizar para transformar los datos y crear las presentaciones.

Por último, la capa de presentación estará comprendida por los mapas de agrupamientos y una página web que muestra las concurrencias entre la señales.

4.2 Etapas del modelo

A continuación se detallan todas las etapas del modelo:

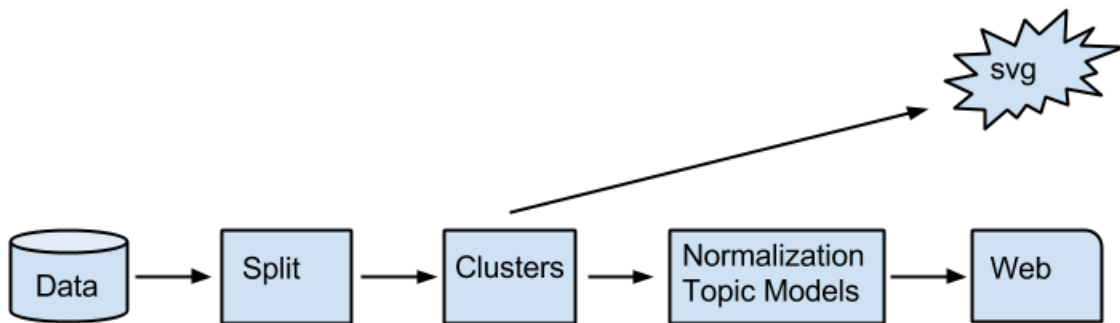


Figura 2 Etapas del modelo

4.2.1 Seleccionar un conjunto de datos (*dataset*) de pruebas

Se seleccionó una base de datos de pruebas que son “*Spoken Arabic Digits*”. Esta información fue recolectada por el Laboratorio de Automatización y Señales de la Universidad de Badji-Mokhtar Annaba, Argelia.

La base de datos cuenta con trece señales de diferentes fuentes; con esto cumple con los requisitos para la base de datos que necesitamos. También cuenta con suficientes datos: existen 8800 bloques de señales distribuidos en dos archivos. Estas señales son cortas, y por consiguiente, hacer varias iteraciones sobre los datos resultará favorable para los experimentos. El largo de las señales es de cuatro tramas hasta noventa y tres tramas.

En la Figura 3 Gráfico de ejemplo de señales, podemos observar cómo se ve un bloque de estos graficado con sus respectivas 13 señales. Podemos ver que hay diferentes magnitudes entre las señales, así como señales con valores positivos, valores negativos y combinadas.

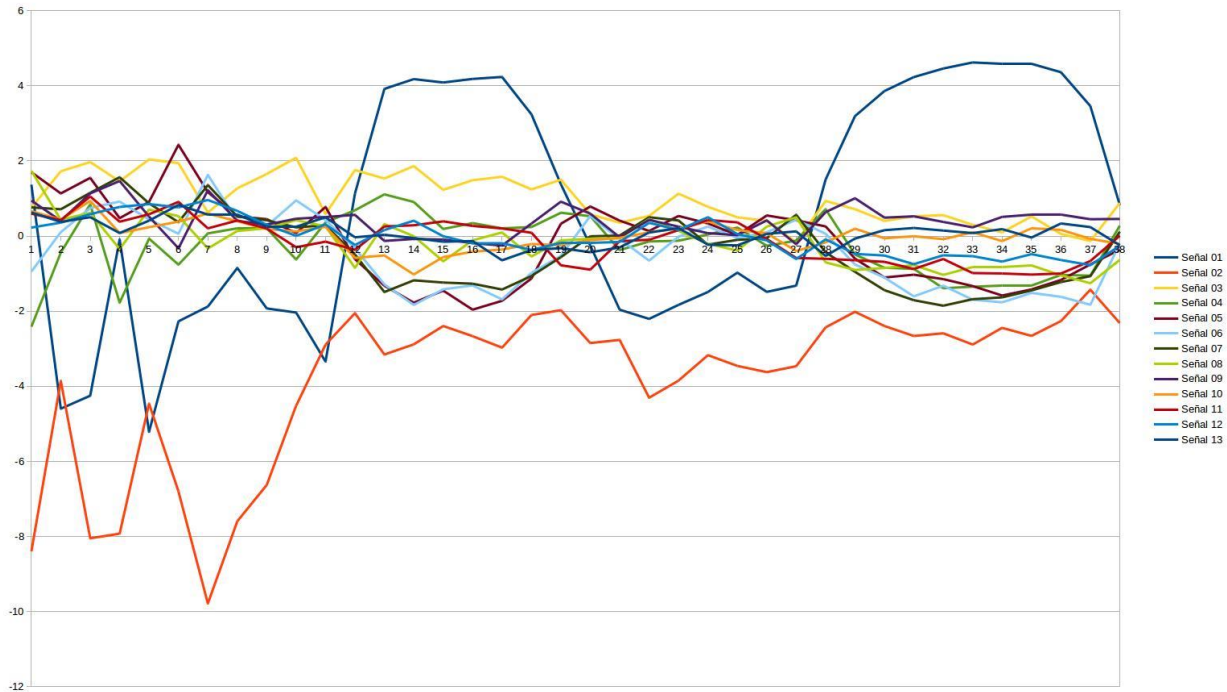


Figura 3 Gráfico de ejemplo de señales

4.2.2 Leer los datos y cargar las estructuras de los datos en memoria.

Una vez que tenemos seleccionada la base de datos de pruebas, el siguiente paso consiste en cargar estos datos y ponerlos en estructuras donde vayan a ser útiles para trabajarlos. Se creó una clase que recibe el nombre del archivo de pruebas, y lo lee para cargar la información. La información es guardada y etiquetada según su origen. Vamos a contar con índices sobre el bloque al que pertenece la información y la señal que esta representa.

4.2.3 Separar los datos por un intervalo deseado y con un traslape deseado.

Se define como intervalo α la cantidad de tramas en la cual se va a hacer un corte transversal de los datos. Vamos a definir también como traslape la cantidad de tramas que se repiten entre cada intervalo.

En la Figura 4 podemos ver gráficamente cómo trabaja el concepto de intervalos. La franja verde corresponde al primer segmento deseado. Para este caso usamos un intervalo de cuatro tramas, lo que crea segmentos de señales de cuatro datos de largo. La franja azul representa los siguientes cuatro datos de las señales, y así sucesivamente se van a ir tomando cuatro datos de cada señal para ir creando cortes transversales de las señales.

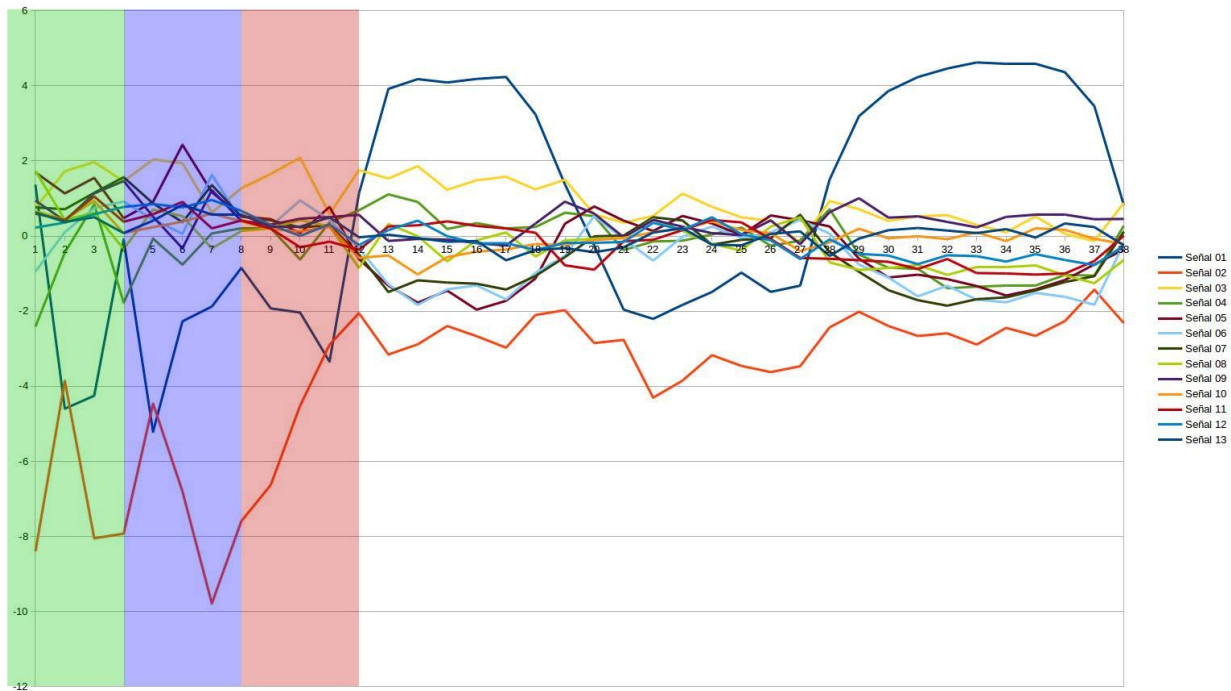


Figura 4 Ejemplo de cortes transversales de las señales

En la Figura 5 podemos ver cómo funciona el concepto de traslape. Los colores verde y azul están superpuestos en una trama para este caso. Esto quiere decir que se repite el cuarto

dato para el primer segmento y el segundo segmento. Entonces el primer segmento va a tener los primeros cuatro datos. Luego el segundo segmento va a tener los datos del tercer elemento hasta el séptimo elemento. Esto se hace ya que existe una probabilidad de perder datos a la hora de cortarlos, debido a que un segmento que debería ser analizado por sus características podría estar cortado a la mitad. Por lo tanto, con este traslape podemos asegurarnos de que los datos importantes siempre van a estar disponibles para ser analizados. Para las pruebas se definieron intervalos de 4 datos y con un traslape de 0 datos. Para cualquier futuro análisis de datos de señales, estas configuraciones tendrán que ser definidas por los expertos de los datos, ya que son ellos los que conocen el tamaño que sería apropiado para efectuar el análisis.

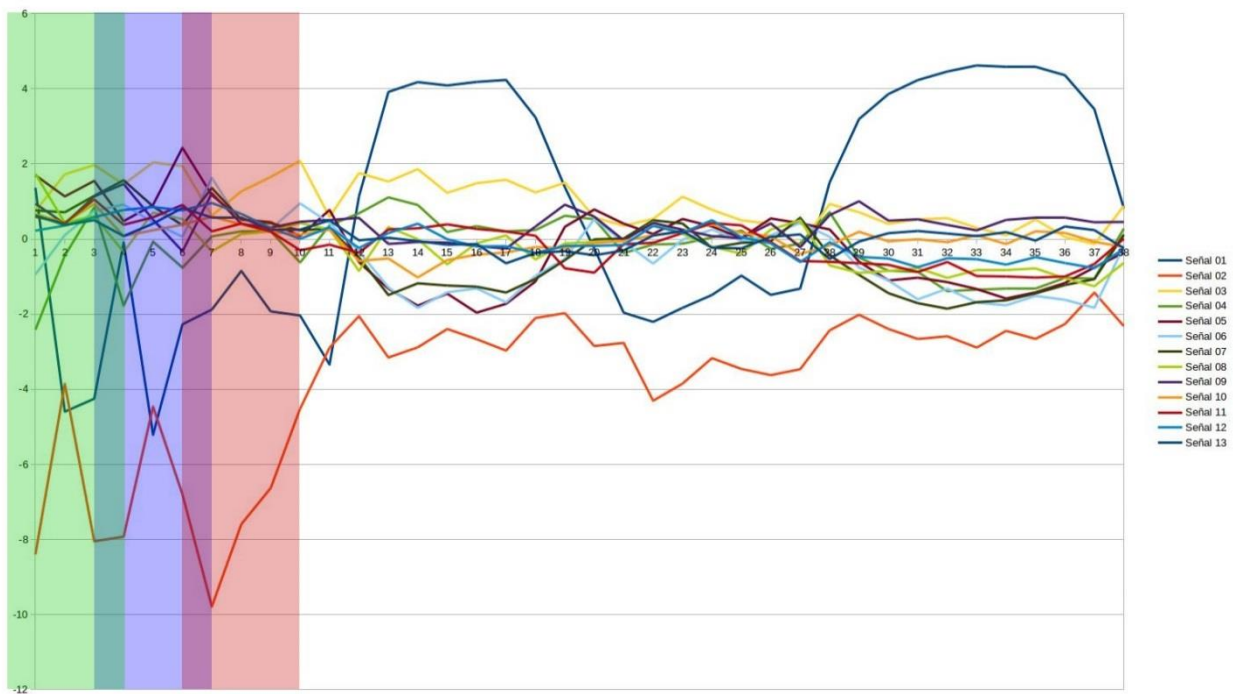


Figura 5 Ejemplo de traslape en los cortes transversales

4.2.4 Cargar los datos de la misma señal en el algoritmo de agrupamientos y ejecutarlo

Una vez que tenemos todos los datos debidamente identificados y segmentados podemos proceder a crear los agrupamientos. Para la creación de agrupamientos se utilizará el algoritmo de Sandalphone. Este es un algoritmo de agrupamiento que utiliza similitud para crear los agrupamientos. El algoritmo Sandalphone necesita de varios parámetros de configuración para trabajar un rango en especial. Este rango es utilizado para ver qué tan grande puede ser la distancia de un elemento al centro del agrupamiento. En otras palabras, este rango se puede variar para crear agrupamientos más grandes o más pequeños. Se crearon varios experimentos con diferentes rangos para determinar cuál sería un valor donde se pudieran encontrar suficientes agrupamientos para trabajar, aunque no demasiados para no crear mucho ruido en la normalización de datos.

También se crearon varios experimentos con diferentes funciones de distancia para determinar cuál función de distancia podía crear agrupamientos con más calidad. Se hicieron pruebas con una función de distancia L1, L2, LP ($P = 0.1$), *Dynamic Time Warping* y Levenshtein. Estas funciones de distancia también se trabajaron con las pendientes de cada punto respectivamente.

Las pendientes se calcularán por cada par de puntos consecutivos de los datos. El vector de pendientes va a tener $n-1$ pendientes.

La forma de ver la calidad de los agrupamientos es a través del promedio de las distancias entre los elementos del agrupamiento y su centro. Se extraerán los cuantiles sobre estas diferencias.

Se seleccionarán solo los agrupamientos de más de 5 elementos.

4.2.5 Imprimir los gráficos lineales que van a ser usados para la visualización

Una vez que tenemos los centros de los agrupamientos se procede con la impresión de los gráficos correspondientes a cada segmento, necesarios para crear las visualizaciones posteriores. Estos se identificarán con el índice creado anteriormente para identificar cada segmento.

Estos gráficos fueron creados con JFreeChart, que es una librería de Java para crear gráficos. Esta librería fue sumamente útil en el proceso, ya que es fácil de implementar, y el resultado final de los gráficos es de bastante calidad. Se debe crear un gráfico por cada segmento de señal; cada gráfico se ~~va a~~ imprimirá en 128 x 85 píxeles, que es un tamaño suficiente para poder ver una señal de cuatro tramas.

Con las configuraciones que se han definido anteriormente, existen 8800 bloques de señales, 13 señales y cerca de 1 - 23 segmentos, según el bloque de las señales, pero en promedio son 10 segmentos por señal; esto nos da un total máximo teórico de 1144000 imágenes de los segmentos de las señales para crear los agrupamientos. Esto, en términos de carpetas del sistema operativo, es un problema, ya que no se deberían trabajar tantos elementos en una sola carpeta. Para solucionar el problema de grandes cantidades de elementos en una carpeta se decidió crear un árbol de carpetas donde se van a guardar los gráficos.

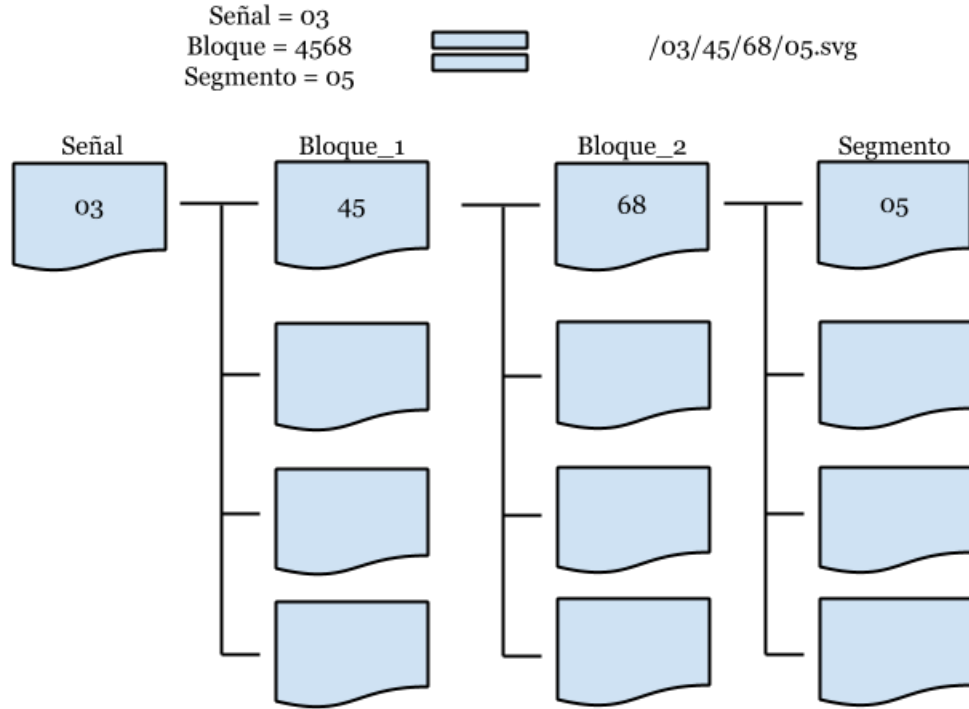


Figura 6 Ejemplo de árbol de carpetas

Para la Figura 6 tenemos el ejemplo en el que el gráfico de la señal 03 del bloque 4568 y del segmento 05 se va a guardar en la jerarquía de carpetas creada como /03/45/68/05.svg

4.2.6 Crear la visualización de los mapas utilizando Graphviz

Graphviz es un sistema para crear visualizaciones con el que se pueden crear diferentes tipos de visualizaciones con este algoritmo. La parte de Graphviz que nos interesa es la generación de mapas.

Se debe crear un archivo “.dot” que va a ser la entrada para el Graphviz. Simplemente se tiene que crear un nodo y especificar cómo se tiene que desplegar este nodo. También se deben representar todos los enlaces entre los nodos y la forma en que estos se relacionan.

Entonces se crea un nodo y se define como se va a desplegar. En este caso, la definición del nodo será una imagen “png” que fue creada en el paso anterior.

Un ejemplo del archivo que se crea de Graphviz se puede ver en la Figura 7

```
digraph g {  
    n1;  
    n1 -> n2;  
    n2;  
    n1 -> n3;  
    n3;  
    n4;  
    n4 -> n5;  
    n5;  
}
```

Figura 7 Ejemplo de estructura de Graphviz

Esta información va a crear un gráfico en la forma que se muestra en la Figura 8.

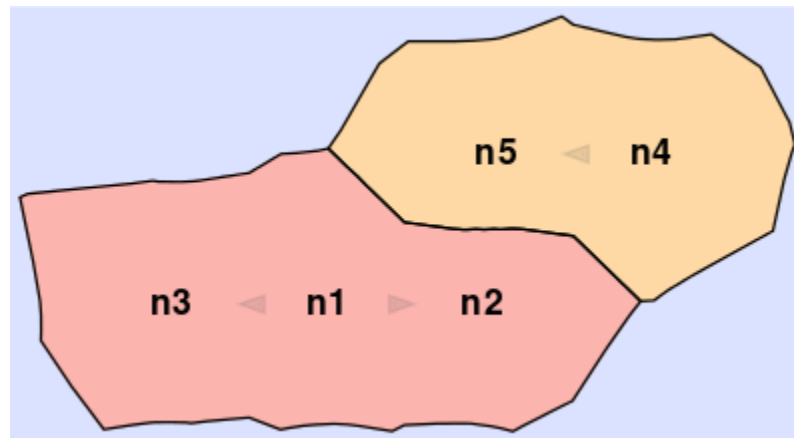


Figura 8 Ejemplo Graphviz

4.2.7 Normalizar las señales

Una vez que tenemos los centros creados, vamos a cargar los centros en Ramiel (R-01) (Algoritmo de k -NN) y a consultar cada segmento de todas las señales para determinar cuál

centro le corresponde y crear las señales normalizadas. En la imagen podemos ver los centros que tenemos creados para esa señal.

Necesitamos cargar el algoritmo de k -NN con todos los centro de los agrupamientos creados previamente. En la Figura 9 se pueden ver unos ejemplos de posibles centros de agrupamientos. Luego hay que consultar cada segmento para encontrar el centro más cercano a este segmento que necesitamos normalizar. Antes de normalizar el segmento necesitamos definir el rango en el cual tiene que estar la distancia entre el nuevo objeto que se va a normalizar, y el nuevo centro con el cual va a quedar normalizado. Podemos extraer este rango del algoritmo de Sandalphone. Con este rango tenemos que validar si la distancia de este segmento hacia el centro que se encontró con el algoritmo de k -NN (Ramiel) es menor o igual. Si la distancia es mayor, vamos a suponer que es un segmento que podemos ignorar; entonces, en su sección correspondiente, lo marcamos con un -1. En la Figura 10 podemos ver cómo se partió en segmentos la señal. Una vez partidas estas señales, se buscó el centro más parecido a la señal y se utilizó el identificador de ese centro para crear la señal normalizada.

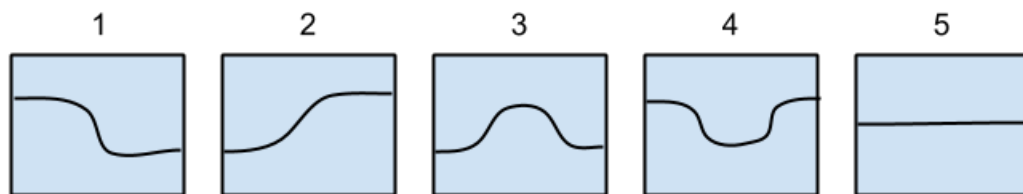


Figura 9 Ejemplos de centros de clústeres

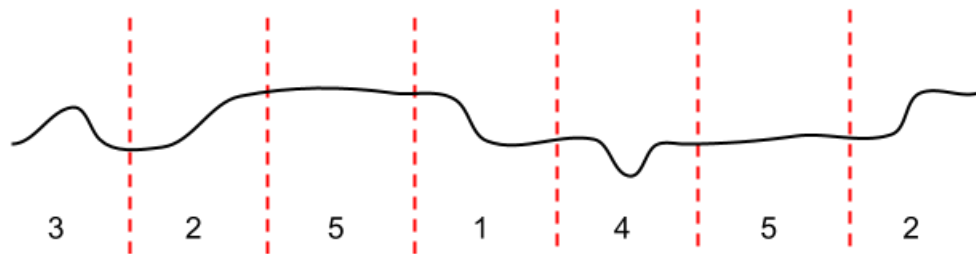


Figura 10 Ejemplo de señal normalizada

Una vez efectuado todo esto, nuestras señales estarán normalizadas y se procederá a guardar la información para la siguiente etapa.

4.2.8 Crear los documentos a partir de las señales normalizadas.

Estos documentos son la estructura de datos de entrada para los modelos de temas. Seguidamente efectuamos un corte transversal de las señales completas y vamos tomando cada segmento de cada parte de las señales. En otras palabras, creamos cada documento con el segmento normalizado de todas las señales. Los documentos van a ser de 13 objetos de largo, ya que estos están asociados a cada señal. Un bloque de 20 tramas con 13 señales nos va a crear 20 documentos de 13 elementos cada uno.

4.2.9 Ejecutar el algoritmo de los modelos de temas con los documentos

Esta etapa es sencilla y no requiere de mucho esfuerzo. Con los documentos creados, estos se cargan simplemente en el algoritmo de “*topic models*” y se ejecuta el algoritmo.

Este algoritmo encontrará anomalías y comportamientos frecuentes que ocurren en los documentos que se le proporcionan.

4.2.10 Crear la visualización de los resultados

Con los modelos de temas ya creados es fácil extraer los resultados y desplegarlos en orden. El peso del primer elemento del tópicos es el valor por medio del cual se van a ordenar los resultados.

Se creará una página web estática que contiene los resultados de los experimentos. Esta página web está hecha utilizando *twitter bootstrap*. En la página se muestra en una línea los resultados de los “*topic models*”. Cada línea contiene un resultado. En lugar de estar desplegando el identificador del centro de agrupamiento que usamos para crear la normalización, vamos a desplegar la imagen correspondiente a este centro.

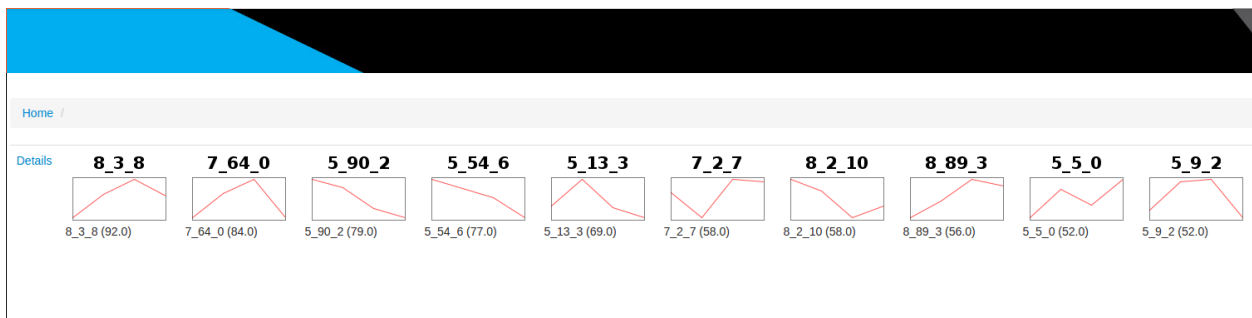


Figura 11 Ejemplo de resultado de un tema

En la Figura 11 se observa cómo se presentan los resultados para cada tema. 8_3_8 es el identificador del centro del agrupamiento y la forma en la que se puede graficar este objeto está representada en forma de gráfico que fue previamente hecho.

Cada línea de resultados muestra un tema diferente. Si fueran modelos de temas de un conjunto de periódicos, las palabras “policía”, “robo” y “asalto” estarían juntas en el mismo

tópico, solo que para este prototipo los “*topic models*” nos están creando grupos de señales que se comportan de forma similar.

4.2.11 Tecnologías usadas

- Ubuntu 14.04.1 LTS
- IntelliJ Idea 13.1.4
- Java(TM) SE Runtime Environment (build 1.7.0_45-b18)
- Twitter Bootstrap Version 2.3.2
- Graphviz
- Librería de Java jfreechart
- Librería de Java MALLET is a Java "MAchine Learning for LanguagE Toolkit".
- Ramiel = k -NN similarity search engine.
- Sandalphone = Algoritmo de agrupamientos que utiliza Ramiel como base.

Capítulo 5 – Experimentos

5.1 Funciones de distancia utilizadas.

Se van a realizar pruebas con las funciones de distancia: L1, L2, LP, *Dynamic Time Warping* y Levenshtein.

5.2 Rangos para el algoritmo de agrupamientos.

El rango en el algoritmo de agrupamientos es el valor que define qué tan diferentes pueden ser las distancias en un agrupamiento para que estos puedan ingresar a él.

Se utilizó el mismo rango para todas las pruebas, y así se puede comparar efectivamente cada señal. Esto se debe a que el rango real con el que se va a trabajar se calcula según una muestra aleatoria de las distancias que existen en los datos. El rango que se utilizó es de 0.0001

5.3 Pendientes y valores.

Para todas las funciones de distancia se usaron las pendientes para cada dos puntos consecutivos de los datos. Es decir, si se tiene un vector $v1 = [x1, x2, \dots, xn]$ se va a calcular la pendiente entre $x1$ y $x2$, luego para $x2$ y $x3$ y así consecutivamente. Con esta operación obtenemos un vector de pendientes sobre las cuales vamos a trabajar $Vp = [p1, p2, \dots, p(n - 1)]$. Se van a realizar pruebas usando los vectores de pendientes de los datos, debido a que es importante cómo se modifica un valor con respecto a su valor previo. Al hacer el análisis usando las pendientes nos enfocamos en el comportamiento de los datos más que en su

magnitud. En algunos casos es más importante saber si un valor está disminuyendo o aumentando.

Se van a generar corridas completas solo con los mejores resultados que se obtengan de los agrupamientos.

Capítulo 6 - Análisis de resultados

Se efectuaron 130 experimentos en total, con dos tipos de datos (valores y pendientes), cinco funciones de distancia (L1, L2, LP, DTW y Levenshtein) y trece señales.

La ejecución de la creación de todos los agrupamientos dura aproximadamente 4 horas. Las imágenes resultantes de los agrupamientos son realmente grandes, de 1500 a 2000 Megapíxeles.

El primer criterio para determinar si un agrupamiento es mejor que otro se establece de acuerdo con el promedio de distancias de su centro a sus hijos. Se debe obtener el promedio de distancias del centro a sus hijos y comparar estas dos distancias. Cuanto menor sea este promedio, tanto mejor será la calidad del agrupamiento.

El segundo criterio para determinar cuáles agrupamientos son mejores se establece según la cantidad de objetos contenida en el agrupamiento. Es evidente que es más fácil obtener un promedio bajo de distancias del centro a sus hijos con agrupamientos más pequeños. A modo de ejemplo, podemos imaginar unas piezas de Legos desarmados: si empezamos a crear agrupamientos con estas piezas, primero dos agrupamientos, luego tres, y así sucesivamente, veremos cómo la distancia promedio del centro a sus hijos en cada agrupamiento va a ir disminuyendo. En algún momento vamos a obtener agrupamientos con piezas iguales y su distancia promedio del centro a sus hijos va a ser prácticamente cero.

Dados estos dos criterios, se estableció una clasificación por el promedio de distancias del centro a sus hijos, y luego por el promedio de elementos contenidos en el agrupamiento. Se sumaron las dos posiciones en estas clasificaciones y se obtuvo la clasificación general para

cada experimento. Como vamos a estar trabajando con trece señales simultáneamente, necesitamos sumar todas las clasificaciones generales de estas trece señales. Los resultados se resumen en la Tabla 1.

Tipo de Dato	Función	Sumatoria
Pendientes	L2	1098
Pendientes	Levenshtein	1222
Pendientes	L1	1383
Valores	Levenshtein	1486
Valores	L2	1681
Pendientes	LP	1893
Valores	LP	1988
Valores	DTW	2012
Valores	L1	2112
Pendientes	DTW	2155

Tabla 1 Resultados resumidos (menos es mejor)

Como se puede apreciar en la Tabla 1, la combinación de la función de distancia L2 utilizando las pendientes obtuvo el menor puntaje. Esto confirma nuestra segunda hipótesis, la función L2 puede obtener agrupamientos compactos y de calidad. La función L2 es una función desde el punto de vista de carga computacional liviana que permite ahorrar tiempo de procesamiento y nos brinda resultados de calidad.

Es importante destacar que todas las distancias fueron normalizadas para poder compararlas efectivamente. Se creó una muestra aleatoria de objetos de la base de datos para poder determinar el promedio y la desviación estándar necesarios para poder normalizar las distancias. Todos los resultados de los experimentos pasaron por este proceso para ser normalizados.

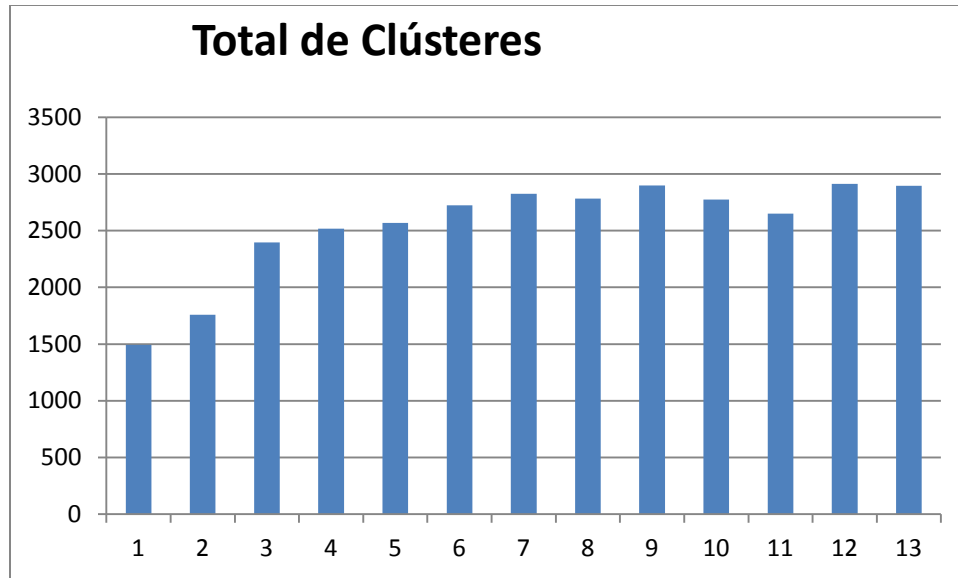


Figura 12 Grafico del total de agrupamientos por señal

En la Figura 12 tenemos el total de agrupamientos por señal para la combinación de la función de distancia L2 usando las pendientes.

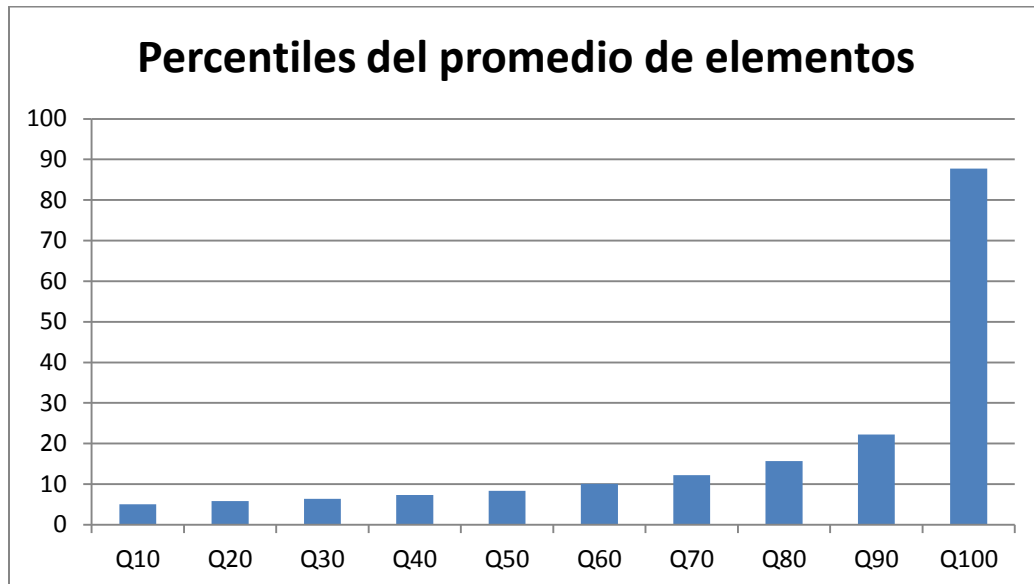


Figura 13 Percentiles del promedio de elementos

En la Figura 13 podemos ver los promedios de todas las señales para los percentiles del Q10, al Q100. Como es de esperarse, se obtiene un comportamiento exponencial.

En la Figura 14 podemos ver una imagen de cómo se ven los agrupamientos generados durante el proceso. La imagen fue reducida alrededor de trescientas ochenta veces para poderla incluir en este documento.

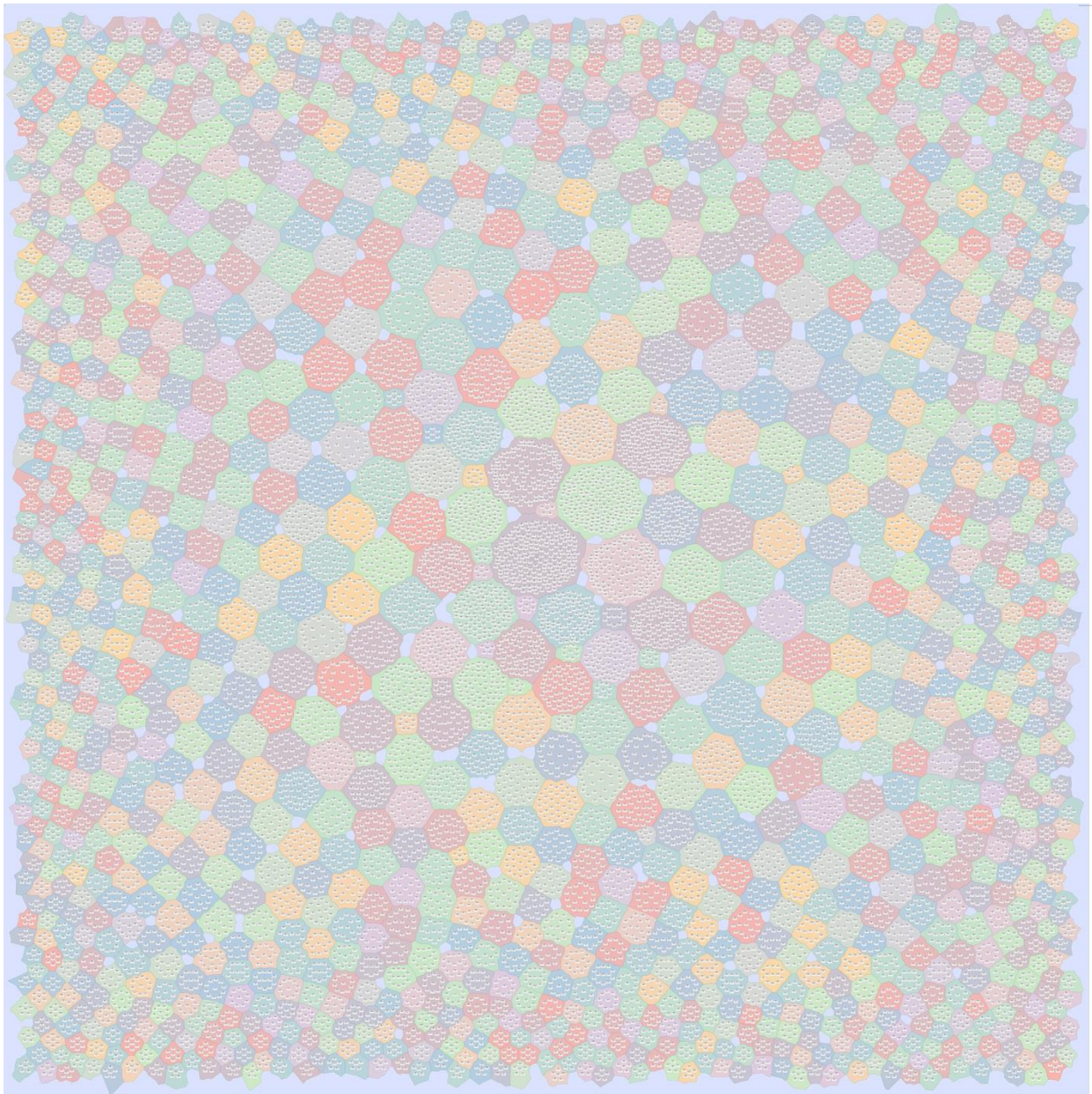


Figura 14 Vista del agrupamiento correspondiente a la señal 1

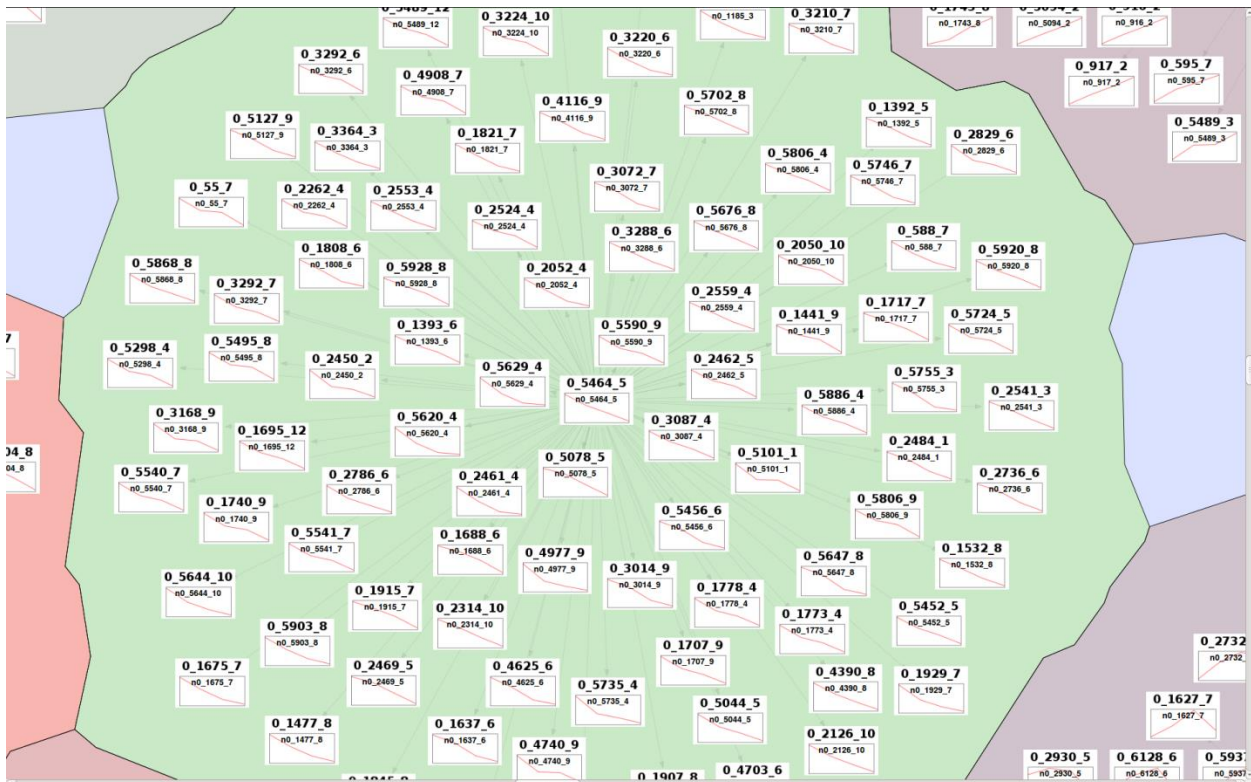


Figura 15 detalle de un agrupamiento de la región central

En la Figura 15 podemos ver con detalle un agrupamiento que se encuentra en la parte central del mapa; es uno de los agrupamientos más grandes. Se puede apreciar que las señales graficadas dentro de él tienen prácticamente la misma forma.

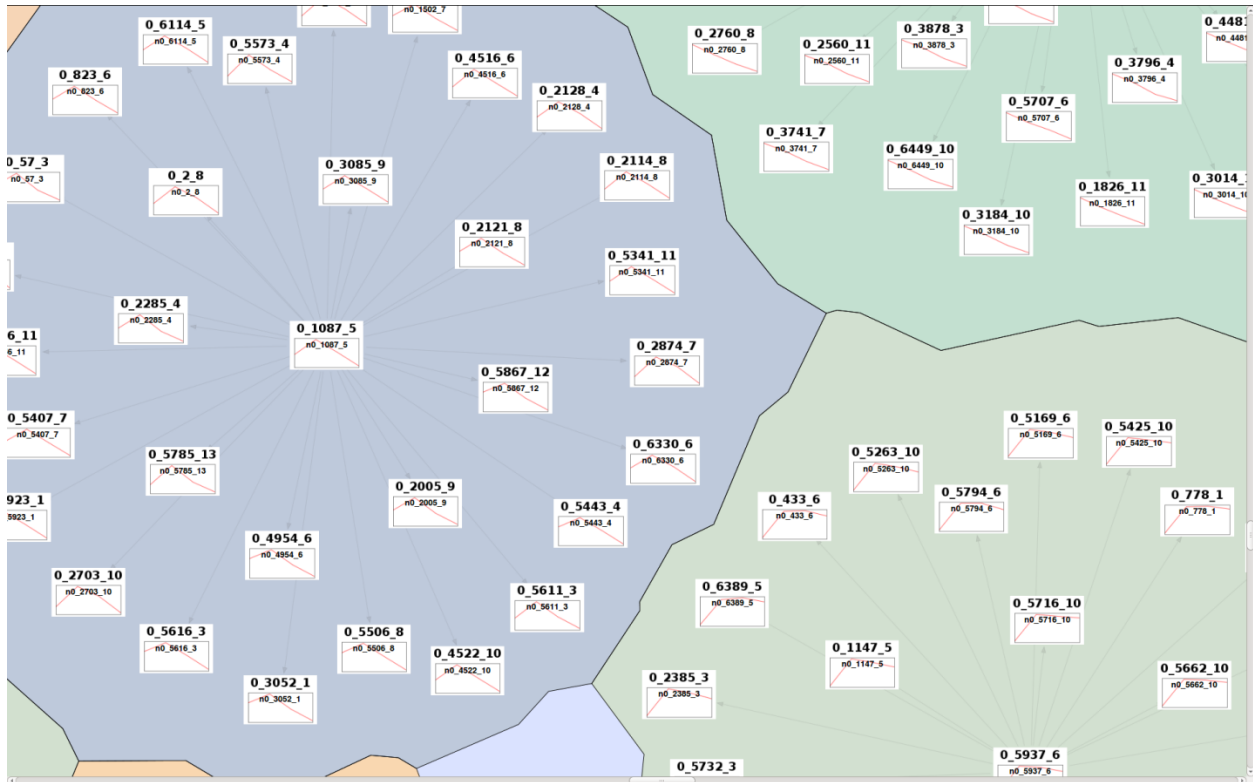


Figura 16 Agrupamiento de tamaño mediano con sus vecinos

En la Figura 16 podemos ver un agrupamiento de tamaño mediano con sus vecinos. Se puede apreciar que los elementos de un mismo país son muy parecidos.

La Figura 17 nos muestra una sección del mapa que contiene agrupamientos pequeños. Es posible que se puedan ver dibujos de señales bastante similares entre diferentes países. En su momento se realizó una revisión sobre estas situaciones y se concluyó que, debido a que no se está imprimiendo la escala en los gráficos, esta varía mucho entre diferentes mapas. Al obviar este detalle, diferentes países pueden parecer similares, ya que los gráficos que se están representando son de diferentes magnitudes. Si se imprime la escala en los gráficos, por ser tan pequeños se toma mucho espacio. Se decidió seguir trabajando sin esta escala, ya que la imagen de los mapas es meramente para visualización.

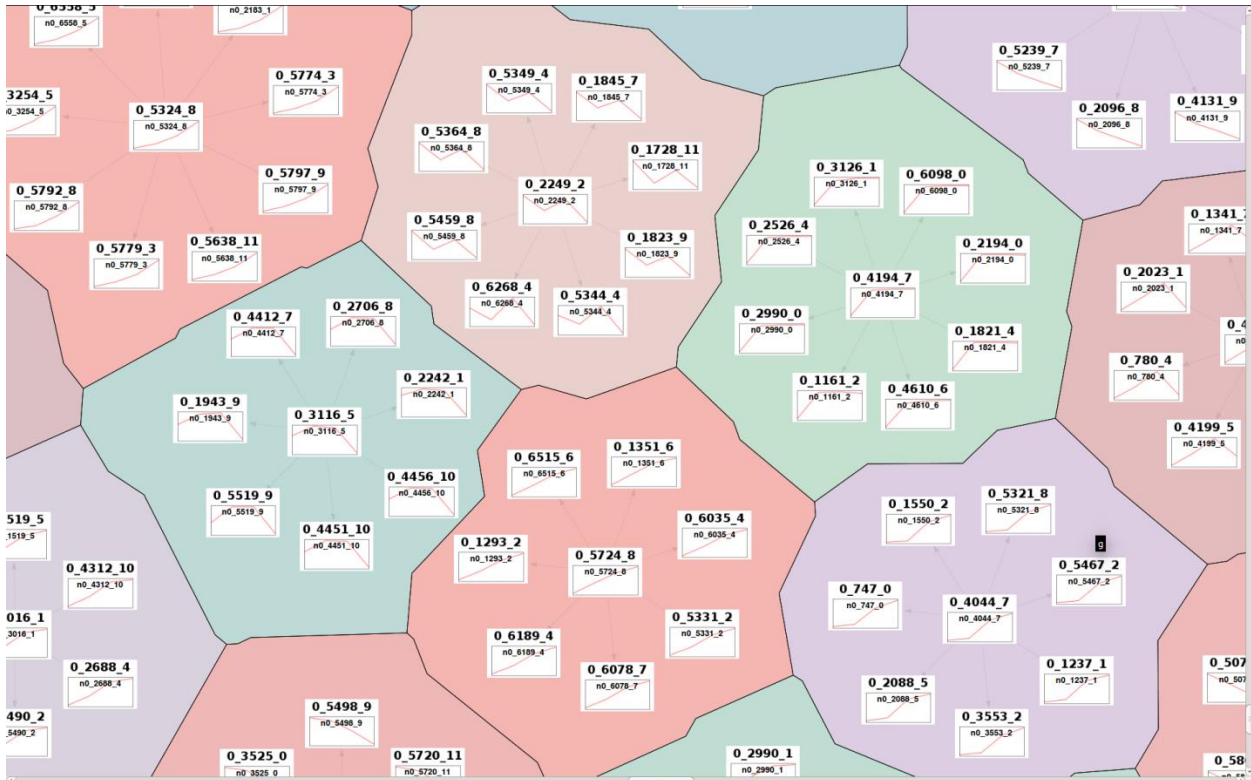


Figura 17 Agrupamientos de elementos pequeños

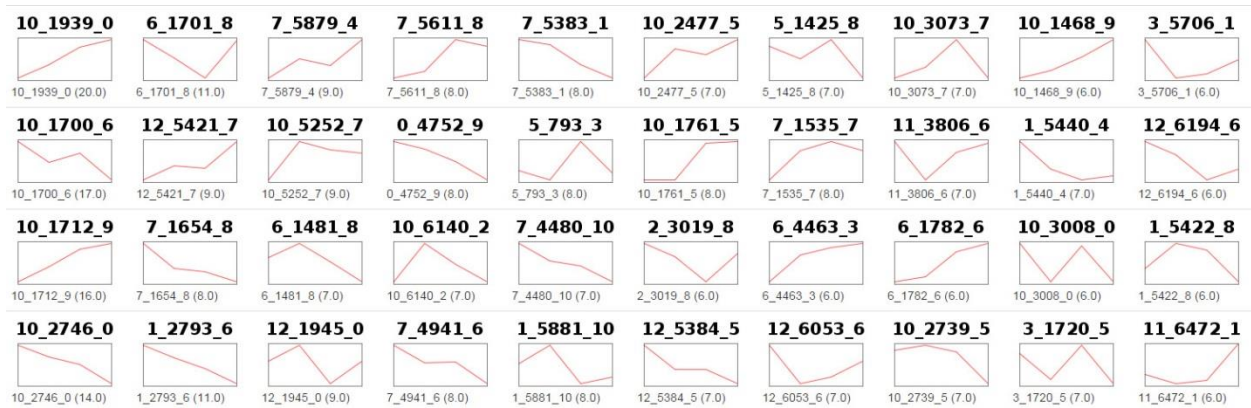


Figura 18 Correlaciones frecuentes entre la señales

En la Figura 18 podemos ver correlaciones frecuentes entre las señales. Cada fila representa un tema común. Si hubiéramos usado el algoritmo para sacar temas de documentos de periódicos, tendríamos un tema como por ejemplo ambiente. Entonces las palabras verde, amigable, reducir y reciclar serían palabras frecuentes en ese tema. El segundo tópico podría

ser sobre conciertos, entonces las palabras cantante, anfiteatro, boleto y banda estarían como las más frecuentes en este tema. En la Figura 18 la primera fila representa un tema. El primer elemento de la primera fila corresponde a la señal 10 del segmento 1939 en la posición 0. Esto nos dice que el comportamiento que se muestra se está repitiendo con frecuencia cuando las otras señales se comportan como se muestra en sus respectivos gráficos. El valor entre paréntesis es un peso sobre qué tan frecuente es el elemento en el tema.

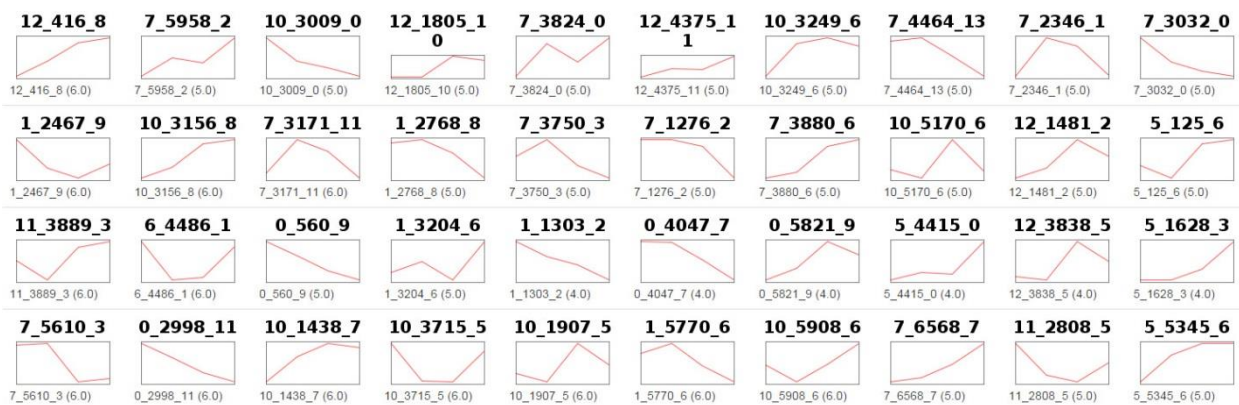


Figura 19 Correlaciones poco frecuentes entre las señales

En la Figura 19 se muestran correlaciones poco frecuentes entre las señales. Cabe destacar que no hay ruido en estas correlaciones. Elementos que son muy frecuentes en muchos temas se excluyen del tema para solo mostrar elementos que consistentemente se repiten con poca frecuencia, pero con un patrón.

Estas correlaciones de señales poco frecuentes y frecuentes van a ayudar en el monitoreo del comportamiento de los sistemas. Estos temas que fueron creados con el algoritmo de modelos de temas se determinaron automáticamente por medio del algoritmo. Cada tema es un comportamiento definido que se presenta con mucha o poca frecuencia en los datos que fueron proporcionados sobre el sistema. Utilizando la analogía de estudiar el

comportamiento de un motor de automóvil, podríamos ver que alguno de estos temas podría ser aceleración. Cuando el conductor del automóvil pisa el acelerador los sensores deberían detectar ciertas cosas que generalmente suceden, como por ejemplo que la velocidad aumente. La temperatura de salida del refrigerante del motor también aumenta y así sucesivamente con todos los sensores. En otras palabras, el tema de aceleración para un motor de automóvil resumiría todas las veces que se aceleró el vehículo. Por lo tanto, cuando el experto en motores quiere revisar la aceleración, no necesita ir a leer todos los registros para determinar cuándo es que se acelera. El sistema le va a mostrar como acelera el vehículo típicamente. Con unos cambios de configuraciones se podrían encontrar temas más específicos tales como aceleración brusca, aceleración leve, aceleración con el motor frío, etc.

Por otro lado, el sistema puede encontrar comportamientos poco frecuentes. Siguiendo con la analogía del automóvil, supongamos que por alguna razón el sistema de enfriamiento no está en perfecto estado. Las pruebas podrían mostrar que el sistema de enfriamiento está trabajando de forma perfecta, puesto que no se presentaron inconvenientes. Pero al analizar los datos con el algoritmo propuesto se podría determinar que la temperatura no es la adecuada para algunos casos. Esto lo encontramos en las correlaciones poco frecuentes, donde podríamos ver que para algunos casos aislados la temperatura no es la correcta. Cabe destacar que la temperatura se podría mantener dentro de los límites aceptables. Pero mediante el análisis se puede determinar que esta temperatura no era la correcta con los parámetros y condiciones que se tenían en ese momento.

Capítulo 7 - Conclusiones

Para esta tesis se tiene como hipótesis utilizar técnicas existentes para procesamiento de grandes cantidades de datos y el uso de técnicas de procesamiento de datos de señales. Esto para crear un algoritmo que analiza datos de señales de diferentes dispositivos simultáneamente.

Se ha implementado un algoritmo que consume datos de señales para encontrar correlaciones entre las señales en distintos intervalos de tiempo. Se pudieron encontrar también anomalías frecuentes. Todos los algoritmos y técnicas empleados crean una herramienta de trabajo para poder analizar datos de señales de manera simultánea y a un costo computacional aceptable.

Definir un formato de archivo para el consumo de los datos. Para este objetivo se pudo definir el formato del archivo de entrada de los datos se pudo implementar y validar. El formato es flexible y liviano.

Buscar un conjunto de datos (dataset) de pruebas con un formato e información similar a la que se almacena en Ad-Astra Rocket. Se encontró una base de datos de pruebas con características de forma muy similares a las de los datos de Ad-Astra Rocket. Esta base de datos es un poco más compacta que la original, pero esto es ideal para poder llevar a cabo pruebas sin un costo computacional muy elevado. Los datos fueron transformados al formato definido previamente.

Buscar una función de distancia ligera y consistente para comparar señales que satisfaga criterios de densidad y similitud: se efectuaron ciento treinta experimentos para determinar

cuál función de distancia trabajaba mejor matemáticamente para crear los agrupamientos de datos. Una vez analizados los resultados de los ciento treinta experimentos se determinó que la función de distancia L2 utilizando los datos de las pendientes era la mejor combinación desde el punto de vista de densidad. Otra función de distancia que presenta buenas características ha sido la Levenshtein también utilizando las pendientes. Este resultado viene a confirmar nuestra hipótesis.

Agrupar a través de un sistema de agrupamientos señales similares. Se utilizó un sistema de agrupamientos para crear conjuntos de segmentos de señales similares. Estos grupos de segmentos similares son segmentos de señales con características similares por lo que van a ser resumidos y utilizados como si fueran una palabra en el lenguaje de la máquina que se está analizando. Se utilizan solo señales de un tipo para crear estos agrupamientos por lo que el algoritmo se tiene que ejecutar una vez por cada señal que se va a analizar.

Evaluar la calidad de los agrupamientos promediando las diferencias de su centro a los objetos. Una vez que se crearon los agrupamientos es necesario evaluar la calidad de los mismos para determinar si la función de distancia utilizada es efectiva. Esto se hizo promediando las distancias del elemento central del agrupamiento a todos los otros miembros del agrupamiento. Se determinó que la función de distancia L2 era la que tenía los mejores resultados.

Entre los objetivos se tenía utilizar una técnica de “Topic Models” para encontrar grupos de señales de diferentes sensores que co-ocurren al mismo tiempo. Esto para identificar patrones y anomalías. Con esta técnica lo que se busca es encontrar comportamientos que co-ocurren al mismo tiempo. Es decir si la señal A aumenta y la señal B disminuye también en el

mismo momento y esto se repite varias veces a lo largo de los datos, este comportamiento va a ser un tema que el algoritmo va a encontrar. En los resultados se puede apreciar como distintos segmentos de señales co-ocurren a lo largo de los datos.

Implementar una visualización que permita a los expertos analizar la información de forma más sencilla. Se pudo crear una página web donde se pueden ver los temas ordenados de mayor a menos según su frecuencia.

El algoritmo se puede utilizar para monitoreo revisando constantemente si se presentan anomalías no controladas. Se pueden revisar también correlaciones frecuentes entre señales que suceden en el mismo instante.

Como oportunidades de mejora, el algoritmo está implementado para ejecutarse una sola vez creando los agrupamientos y toda la información sobre los modelos de temas. Se pueden realizar mejoras para guardar los agrupamientos y tener lista la información de documentos anteriores para futuras ejecuciones del sistema. En otras palabras, sería crear un pre-análisis para que en futuras ejecuciones solo se ejecute parcialmente el algoritmo.

Los pasos que deben seguirse consistirían en buscar un entorno real donde pueda probarse el algoritmo. Los campos donde existan pocos estudios o poca información acerca de cómo se comporta determinada máquina serían entornos ideales para poder demostrar el potencial del algoritmo. El algoritmo va a aprender sobre el entorno en el cual va a estar trabajando, y va a presentar los temas que probabilísticamente sean los más importantes, y además, a su vez, los temas que podrían ser considerados como anomalías.

Bibliografía

[Agrawal, Gehrke, Gunopulos, Raghavan 2005]

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (2005). Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11(1), 5-33.

[Aggarwal, Wolf, Yu, Procopiuc, Park 1999]

Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S. (1999, June). Fast algorithms for projected clustering. In *ACM SIGMOD Record* (Vol. 28, No. 2, pp. 61-72). ACM.

[Berndt, Clifford 1994]

Berndt, D. J., & Clifford, J. (1994, July). Using Dynamic Time Warping to Find Patterns in Time Series. In *KDD workshop* (Vol. 10, No. 16, pp. 359-370).

[Blei, 2002]

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4), 77-84.

[Chávez, Navarro, Baeza-Yates, Marroquín 2001]

Chávez, E., Navarro, G., Baeza-Yates, R., & Marroquín, J. L. (2001). Searching in metric spaces. *ACM computing surveys (CSUR)*, 33(3), 273-321.

[Deza, Deza 2009]

Deza, M. M., & Deza, E. (2009). *Encyclopedia of distances* (pp. 1-583). Springer Berlin Heidelberg.

[Ellson, Gansner, Koutsofios, North, Woodhull 2002]

Ellson, J., Gansner, E., Koutsofios, L., North, S. C., & Woodhull, G. (2002, January). Graphviz—open source graph drawing tools. In *Graph Drawing* (pp. 483-484). Springer Berlin Heidelberg.

[Gilbert 2002]

Gilbert, D. (2002). The jfreechart class library. *Developer Guide. Object Refinery*, 7.

[Gionis, Indyk, Motwani 1999]

Gionis, A., Indyk, P., & Motwani, R. (1999, September). Similarity search in high dimensions via hashing. In *VLDB* (Vol. 99, pp. 518-529).

[Heyer, Kruglyak, Yooseph 1999]

Heyer, L. J., Kruglyak, S., & Yooseph, S. (1999). Exploring expression data: identification and analysis of coexpressed genes. *Genome research*, 9(11), 1106-1115.

[Hjaltason, Samet 2003]

Hjaltason, G. R., & Samet, H. (2003). Index-driven similarity search in metric spaces (survey article). *ACM Transactions on Database Systems (TODS)*, 28(4), 517-580.

[Houle, Kriegel, Kröger, Schubert, Zimek 2010]

Houle, M. E., Kriegel, H. P., Kröger, P., Schubert, E., & Zimek, A. (2010, January). Can shared-neighbor distances defeat the curse of dimensionality?. In *Scientific and Statistical Database Management* (pp. 482-500). Springer Berlin Heidelberg.

[Kriegel, Kröger, Zimek 2009]

Kriegel, H. P., Kröger, P., & Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1), 1.

[Lerner 2012]

Lerner, R. M. (2012). At the forge: twitter bootstrap. *Linux Journal*, 2012(218), 6.

[Müller-Molina 2012]

Müller-Molina, A. (2012). Performance of the simMachines R-01 Similarity Search Engine. <http://simmachines.com/we-discover/> 2015

[Ratanamahatana, Keogh 2004]

Ratanamahatana, C. A., & Keogh, E. (2004, August). Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data*.

[Ruiz 1986]

Ruiz, E. V. (1986). An algorithm for finding nearest neighbours in (approximately) constant average time. *Pattern Recognition Letters*, 4(3), 145-157.

[Samet 2006]

Samet, H. (2006). *Foundations of multidimensional and metric data structures*. Morgan Kaufmann.

[Yi, Jagadish, Faloutsos 1998]

Yi, B. K., Jagadish, H. V., & Faloutsos, C. (1998, February). Efficient retrieval of similar time sequences under time warping. In *Data Engineering, 1998. Proceedings., 14th International Conference on* (pp. 201-208). IEEE.

[Youssef, Abdel-Galil, El-Saadany, Salama 2004]

Youssef, A. M., Abdel-Galil, T. K., El-Saadany, E. F., & Salama, M. M. A. (2004). Disturbance classification utilizing dynamic time warping classifier. *Power Delivery, IEEE Transactions on*, 19(1), 272-278.

[Zezula, Amato, Dohnal, Batko 2006]

Zezula, P., Amato, G., Dohnal, V., & Batko, M. (2006). Similarity search: the metric space approach (Vol. 32). Springer Science & Business Media.