



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Doctorado

Selección de Variables en descripciones de objetos
crono-valuados

presentada por

MC. Marilú Cervantes Salgado

como requisito para la obtención del grado de
Doctor en Ciencias de la Computación

Director de tesis

Dr. Raúl Pinto Elías

Codirector de tesis

Dra. Andrea Magadán Salazar

Cuernavaca, Morelos, México. Febrero de 2022.



ESC\FORDOC09

Cuernavaca, Morelos, 08/febrero/2022

ASUNTO: ACEPTACIÓN DEL TRABAJO DE TESIS DOCTORAL

DR. JUAN GABRIEL GONZÁLEZ SERNA
JEFE DEL DEPARTAMENTO DE CIENCIAS COMPUTACIONALES
PRESENTE

Los abajo firmantes, miembros del Comité Tutorial de la Tesis Doctoral de la alumna **M.C. MARILÚ CERVANTES SALGADO** manifiestan que después de haber revisado su trabajo de tesis doctoral titulado **"SELECCIÓN DE VARIABLES EN DESCRIPCIONES DE OBJETOS CRONO-VALUADOS"**, realizado bajo la dirección del **Dr. Raúl Pinto Elías** y la codirección de la **Dra. Andrea Magadán Salazar**, el trabajo se ACEPTA para proceder a su impresión.

ATENTAMENTE

"Excelencia en Educación Tecnológica®"
"Educación Tecnológica al Servicio de México"



DR. RAÚL PINTO ELÍAS
CENIDET



DRA. ANDREA MAGADÁN SALAZAR
CENIDET



DR. JUAN GABRIEL GONZÁLEZ SERNA
CENIDET



DR. GERARDO REYES SALGADO
CENIDET



DR. DANTE MÚJICA VARGAS
CENIDET



DR. JORGE ALBERTO FUENTES PACHECO
INST. DE INVESTIGACIÓN EN CIENCIAS BÁSICAS

C.c.p.: Lic. Silvia del Carmen Ortiz Fuentes/ Jefa del Depto. de Servicios Escolares
Dr. Carlos Manuel Astorga Zaragoza / Subdirector Académico Expediente

Recibe aceptación
17/02/2022




Cuernavaca, Mor., 18/febrero/2022
No. De Oficio: SAC/51/2022
Asunto: Autorización de impresión de tesis

**MARILÚ CERVANTES SALGADO
CANDIDATA AL GRADO DE DOCTORA EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E**

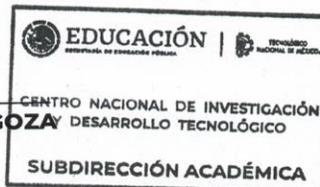
Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "SELECCIÓN DE VARIABLES EN DESCRIPCIONES DE OBJETOS CRONO-VALUADOS", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE
Excelencia en Educación Tecnológica®
"Educación Tecnológica al Servicio de México"



DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO



C. c. p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/CHG

DEDICATORIA

A Dios, porque Él hizo posible las cosas imposibles para mí.

AGRADECIMIENTOS

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada para la realización de mis estudios de doctorado. Al Tecnológico Nacional de México (TecNM) y al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por facilitar sus instalaciones para desarrollar mi investigación doctoral.

A mis director y codirectora de tesis: Doctor Raúl Pinto Elías y la Doctora Andrea Magadan Salazar, por tener la confianza en mí y por aceptarme como su tesista. Por todos sus consejos, por compartir conmigo sus ideas y por todas esas horas dedicadas para llevar a buen fin mi proyecto doctoral.

A mi comité revisor: Dr. Dante Mújica Vargas, Dr. Juan Gabriel González Serna, Dr. Gerardo Reyes Salgado y Dr. Jorge Alberto Fuentes Pacheco, por el tiempo invertido en la presente investigación y por sus comentarios.

A mi familia y amigos por su apoyo y amor incondicional.

RESUMEN

Al establecerse la pregunta de investigación: ¿Es posible incorporar el sentido evolutivo de los datos crono-valuados en una técnica de selección de variables? Se propuso la presente investigación de dos técnicas que mantuvieran o mejoraran la clasificación, conservando la separabilidad del universo de los datos. Los datos crono-valuados describen un fenómeno en diferente número de muestras, mientras dicho fenómeno va cambiando en el tiempo, las características se mantienen fijas. En esta investigación, la variación de las muestras en el tiempo es referido como sentido de evolución y su importancia para la técnica de selección es concluyente. El acumulado de estos fenómenos evolutivos forma los conjuntos de datos crono-valuados y, en esta investigación, el entorno se presenta en un ambiente supervisado.

Como propuesta de solución se presentan dos enfoques: a) en base a las redes Bayesianas para modelado de los datos y la sábana de Markov para la selección de variables, y b) en base al algoritmo de la Lógica Combinatoria para el Reconocimiento de Patrones. Las aportaciones de la investigación se pueden enlistar como: 1) formulación matemática de la sábana de Markov para redes Bayesianas Dinámicas, 2) metodología para la función de compactación de datos crono-valuados a lineales tomando en cuenta el sentido evolutivo de los datos y 3) heurística para el establecimiento de umbral de decisión de similitud.

Dados los experimentos en la investigación se encontró que la aplicación de las técnicas de selección de variables propuestas supera los resultados obtenidos de la clasificación utilizando todas las variables con un mínimo de 19% (Redes Bayesianas Dinámicas con sábana de Markov propuesta) y un máximo de 75% (redes Bayesianas con función de compactación). Teniendo en comparación la clasificación exhaustiva a tres de los *dataset*, se tienen que las propuestas se encuentran entre 10 puntos porcentuales de los máximos posibles.

Con los resultados anteriores se concluye que las propuestas están seleccionando las variables relevantes de los *datasets* de datos crono-valuados tomando en cuenta el sentido cronológico de estos datos. Además, las propuestas desarrolladas son autónomas, es decir no se necesita de expertos en los universos de datos.

ABSTRACT

From the research question: Is it possible to incorporate the evolution sense of chronologically linked data into a feature selection technique? this research was proposed; where two techniques were used to maintain or to improve classification, keeping separation of data universe. The chronologically linked data describes a phenomenon in different number of samples while they change in time, but features are kept the same. In this research, the changes of samples over time is called evolution sense and its importance for the selection technique is conclusive. All of these evolutionary phenomena shape the chronologically linked data sets and, in this research, the environment will be presented as supervised.

The solution proposal presents two approaches: a) using Bayesian Networks to model data and the Markov blanket to select features, and b) using Logical Combinatorial to Pattern Recognition. The research contributions can be listed as: 1) mathematical formulation of Markov blanket to Dynamic Bayesian Networks, 2) Function to flatten chronologically linked data and 3) heuristic to establish a decision threshold in a similarity comparison.

Given the experimentation, it was found that the feature selection techniques here proposed, exceed the classification results obtained using all variables with at least 19% (using the Dynamic Bayesian Network and the Markov blanket that is proposed in here) and a minimum of 75% (using the flattening function). Having as comparative point an exhaustive classification exercise, it was found that the proposed techniques are situated within 10 perceptual points of the maximum possible.

Having the results, it can be concluded that the feature selection techniques are selecting the relevant features out of the chronologically linked datasets taking in consideration the evolution sense of them. Also, the proposed techniques are autonomous, it means no data experts were required.

Contenido

Capítulo 1 Introducción	13
1.1 Descripción del problema	13
1.2 Delimitación del problema	14
1.3 Objetivo.....	15
1.4 Alcances y limitaciones	15
1.5 Justificación y beneficios	16
1.6 Pregunta de investigación e hipótesis.....	16
1.7 Antecedentes del proyecto	16
1.8 Estado del arte	17
1.8.1 Estado del arte y su contexto teórico.....	20
1.8.2 Resumen de métodos estudiados y conclusiones del Estado del Arte	41
1.9 Descripción de la solución	44
1.10 Discusión	44
Capítulo 2 Propuesta de solución	45
2.1 Planteamiento de la solución	45
2.1.1 Descripción teórica de los elementos de la solución, enfoque uno - RB.....	45
2.1.2 Descripción teórica de los elementos de la solución, enfoque dos - LCRP	48
2.2 Modelo conceptual de la solución.....	50
2.3 Descripción de los elementos del modelo conceptual	51
2.3.1 Aportaciones de la investigación: Enfoque Uno - RB.....	51
2.3.2 Aportaciones de la investigación: Enfoque Dos - LCRP.....	55
2.4 Variables / criterios de control	60
2.5 Discusión	60
Capítulo 3 Análisis, diseño e implementación del sistema	61
3.1 Plataforma, lenguaje y conjunto de datos	61
3.1.1 R y R Studio	61
3.1.2 <i>Datasets</i>	61
3.1.3 Implementación de la clasificación	62

3.1.4 Interconexiones	62
3.2 Diagramas de flujo de los módulos del sistema y su correspondiente pseudocódigo.....	62
3.2.1 Función de comparación entre objetos	63
3.2.2 Umbrales de decisión para la función de comparación entre dos objetos	67
3.2.3 Heurística para umbral de decisión.....	69
3.2.4 Diseño e implementación de la sábana de Markov para redes bayesianas dinámicas	71
3.2.4 Diseño e implementación de la adaptación del algoritmo de LCRP	74
Capítulo 4 Validación y experimentación.....	76
4.1 Experimento #1.....	77
4.2 Experimento #2.....	79
4.3 Experimento #3.....	82
4.4 Experimento #4.....	83
4.5 Experimento #5.....	85
4.6 Experimento #6.....	86
4.7 Experimento #7.....	88
4.8 Experimento #8.....	92
4.9 Experimento #9.....	94
4.10 Experimento #10.....	96
4.11 Experimento #11.....	98
4.12 Experimento #12.....	99
Análisis de los resultados.....	100
Capítulo 5 conclusiones y trabajo futuro	102
Aportaciones y productos académicos	103
Trabajo futuro.....	103
Referencias.....	104
Anexo A.....	110
Anexo B	111
Anexo C	112
Anexo D.....	119
Anexo E.....	120

Índice de Figuras

Figura 1.1 Ilustración de ejemplo de cuatro gestos y sus coordenadas	14
Figura 1.2 Adaptación hecha por Ochoa Somuano en su trabajo de investigación (Ochoa Somuano, 2016)	16
Figura 1.3 Número de publicaciones por año relacionadas con la selección de variables en la última década	18
Figura 1.4 Representación visual de los datos puntuales (a) y los vinculados cronológicamente (b) como se presenta en (Cervantes Salgado & Pinto Elías, 2018).....	20
Figura 1.5 El flujo general del algoritmo propuesto en (Nassuna et al., 2020)	22
Figura 1.6 Estructura general del enfoque usado en (Hancer et al., 2018).....	24
Figura 1.7 Descripción visual del método de selección de características de conjunto aproximado de correlación propuesto por (Sridevi & Murugan, 2014).....	26
Figura 1.8 Estructura geométrica del conjunto de entrenamiento de ejemplo.....	29
Figura 1.9 Interpretación gráfica de Consensus Clustering en (Cisotto et al., 2020)	32
Figura 1.10 MLP de una capa oculta tal como se muestra en (Pedregosa et al., 2011).....	33
Figura 1.11 Diagrama de flujo del modelo de detección de intrusos basado en CNN en (Wu et al., 2018)	35
Figura 1.12 Representación visual de los datos de las redes sociales y su ilustración matricial como se muestra en (Tang & Liu, 2014).	40
Figura 2.1 Un nodo X es condicionalmente independiente de todos los otros nodos en la red dada su sábana de Markov (área gris) (Russell & Norvig, 2013)	47
Figura 2.2 Esquema de representación para la selección de variables con el enfoque LCRP presentado en (Martínez-Trinidad & Guzmán-Arenas, 2001).....	49
Figura 2.3 Mapa Conceptual de la investigación.....	50
Figura 2.4 2-TBN que define una distribución de independencia para demostrar el concepto de sábana de Markov sobre redes Bayesianas dinámicas.....	53
Figura 2.5 Descripción del resultado de la función de comparación.....	56
Figura 2.6 Dos objetos de diferente longitud pertenecientes a la misma clase.....	57
Figura 2.7 Heurística propuesta para el establecimiento del umbral de decisión	59
Figura 3.1 Diagrama de flujo de la Función de comparación entre objetos.....	63
Figura 3.2 Diagrama de flujo de Heurística propuesta para el umbral de decisión	69
Figura 3.3 Diagrama de flujo para la obtención de la sábana de Markov dinámica	71
Figura 3.4 Diagrama de flujo de la adaptación al algoritmo LCRP	74
Figura 4.1 Resumen de los resultados de la Tabla 4.7	84
Figura 4.2 Comportamiento de los resultados utilizando la selección de variables a RB y RBD.....	87
Figura 4.3 Gráfica de comportamiento archivos A1/A2.....	89
Figura 4.4 Gráfica de comportamiento archivos A1/A2.....	90

Figura 4.5 Gráfica de comportamiento archivos B1/B3.....	90
Figura 4.6 Gráfica de comportamiento archivos B1/B3.....	91
Figura 4.7 Gráfica de comportamiento archivos C1/C3.....	91
Figura 4.8 Gráfica de comportamiento archivos C1/C3.....	92
Figura 4.9. Máximos de la intersección de los tres rangos explorados	95
Figura 4.10 $TP + TN$ arriba de 1000 para las intersecciones en los rangos a, b y c a, b and c.....	96

Índice de tablas

Tabla 1.1 Co-ocurrencia para variables binarias	27
Tabla 1.2 Conjunto de entrenamiento utilizado en (Murty & Devi, 2011).....	28
Tabla 1.3 Resumen de los enfoques estudiados y sus posibilidades de adaptación con datos vinculados cronológicamente.	42
Tabla 3.1 Descripción detallada del conjunto de datos de <i>Gesture Phase Segmentation</i>	62
Tabla 4.1 parámetros utilizados en el experimento Uno.....	77
Tabla 4.2 Número de posibles combinaciones por número de variables en el <i>dataset</i>	79
Tabla 4.3 Clasificación exhaustiva con dos clases.	80
Tabla 4.4 Clasificación exhaustiva para cinco clases.	81
Tabla 4.5 máximo <i>Accuracy</i> posible de todos	82
Tabla 4.6 Resultados de la sábana de Markov a red.....	83
Tabla 4.7 Resultados obtenidos al aplicar la	85
Tabla 4.8 Medidas de clasificación a los <i>datasets</i> con formato cinco clases	86
Tabla 4.9 Resultados experimento siete	89
Tabla 4.10 Resultados al cambiar el porcentaje de promedio del umbral utilizando LCRP.....	93
Tabla 4.11 Resultados para el enfoque BN / MB para dos clases.....	97
Tabla 4.12 Resultados para el enfoque BN / MB para cinco clases.....	97
Tabla 4.13 Resultados para el enfoque LCRP utilizando la función de comparación entre objetos, dos clases.....	98
Tabla 4.14 Resultados para el enfoque LCRP utilizando la función de comparación entre objetos, cinco clases.....	98
Tabla 4.15 Resultados utilizando heurística para umbral y enfoque RB/MB.....	99
Tabla 5.1 Experimento del trabajo de investigación contrastados con los objetivos	102

CAPÍTULO 1

INTRODUCCIÓN

El presente capítulo tiene como objetivo introducir el tema de investigación y los conceptos necesarios para facilitar la lectura de los capítulos posteriores. También, pretende justificar el tema presentando el estado del arte con el cual se muestran los espacios y el área de oportunidad.

1.1 Descripción del problema

La selección de variables tiene como objetivo mejorar la calidad de la clasificación. A pesar de la vasta investigación que existe acerca de selección de variables, aún se mantienen problemas sin resolver. Entre ellos la selección de variables en datos crono-valuados.

La selección de variables en datos crono-valuados se centra en conseguir que la esencia cronológica del dato sea tomada en cuenta para la selección. Ya que esta esencia cronológica es la que logra distinguir entre eventos que poseen los mismos valores en diferentes instantes. El problema existe ya que las muestras en tales datos no son unidimensionales, pueden poseer diferente longitud y están ligados cronológicamente.

Hasta ahora, de acuerdo con el estudio de estado del arte, la esencia cronológica de los datos se ha estudiado sin que se logre conseguir la conservación o correcta manipulación de esta característica. Por lo tanto, el desarrollo de un modelo de selección de variables que trabaje los datos crono-valuados y respete el orden cronológico de los mismos se considera un problema sin resolver.

Los datos crono-valuados ofrecen un amplio rango de aplicaciones, incluyendo interacciones entre humano-robot y humano-computadora (entre otros dispositivos), seguridad, reconocimiento de lenguaje de señas (gestos), entrenamiento físico, sistemas de asistencia en hogares y medicina.

1.2 Delimitación del problema

La solución se buscó con dos enfoques: redes Bayesianas / sábana de Markov y Lógica Combinatoria para el Reconocimiento de Patrones (LCRP). Los datos que se utilizaron son datos crono-valuados existentes en algún repositorio con aprendizaje supervisado.

Datos crono-valuados

A continuación, se describen las características que hacen que los datos crono-valuados sean un problema complejo para la selección de variables. Los datos crono-valuados son piezas de información de múltiples muestras vinculadas cronológicamente que en adelante serán llamados *objetos*. Estos objetos pueden tener diferente longitud dentro de la misma clase, pero el número de atributos es el mismo para todos ellos. Un ejemplo de dichos datos son los *datasets* de *gestos*. En la figura 1.1, se ilustran cuatro gestos diferentes; en esta, el gesto *a* se describe en ocho unidades de tiempo, el gesto *b* requiere cuatro unidades de tiempo para completarse, los gesto *c* y *d* toman seis unidades de tiempo. Estas unidades de tiempo representan la duración de cada gesto.

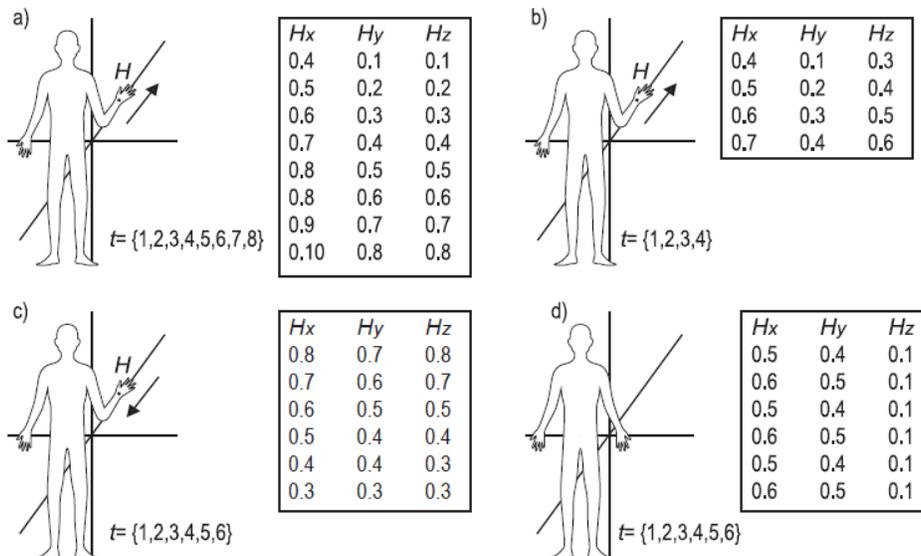


Figura 1.1 Ilustración de ejemplo de cuatro gestos y sus coordenadas

Como ilustra la figura 1.1, dos gestos de la misma clase pueden tener diferente longitud (figuras *a* y *b*). También se muestra que los elementos de dos clases de gestos diferentes (*a* y *c*) podrían describirse con los mismos valores, pero diferente orden cronológico. Finalmente, en la misma figura, se puede notar que la mediana del atributo H_x en los gestos *c* y *d* es la misma, pero pertenecen a clases diferentes.

Los datos de gestos (comprendidos entre los datos crono-valuados) se pueden resumir como:

- objetos de múltiples muestras,

- objetos con diferente longitud,
- vinculados cronológicamente.

Estas características de los datos crono-valorados se incluyen en la investigación, destacando la importancia del sentido de evolución de cada gesto como parte de la riqueza de estos datos y parte del problema a resolver.

Con este trabajo de investigación se pretende proporcionar una técnica de selección de variables que, haciendo uso de datos crono-valorados, mantenga la separación entre clases mientras observa la evolución cronológica de los datos. Los datos para la investigación fueron obtenidos del repositorio público (Dua & Graff, 2017), en particular los referenciados como *Gesture Phase Segmentation* (Madeo, Lima, & Peres, 2013), los cuales cumplen con las características mencionadas anteriormente (objetos multi-muestras, con diferente longitud y vinculados cronológicamente).

1.3 Objetivo

Desarrollar una técnica de selección de variables que logre: conservar la división de clases y tome en cuenta el sentido de evolución de los datos crono-valorados.

1.4 Alcances y limitaciones

- En esta investigación se persigue una técnica de selección de variables en un ambiente supervisado para datos crono-valorados.
- Se propuso encontrar las variables más significantes de *datasets* cronológicamente vinculados utilizando dos enfoques: el Bayesiano/sábana de Markov y el de la LCRP.
- Se propuso la formulación matemática de la sábana de Markov a redes Bayesianas dinámicas.
- Se consideró que la cardinalidad de los objetos con respecto al tiempo pueda ser variable.
- La información de los *datasets* puede contener más de dos clases, pero debe mantener el número de variables constante.
- Los enfoques propuestos no contemplan la ausencia de información, es decir, todas las variables en todos los tiempos muestreados deberán tener un valor definido.
- Los valores de los conjuntos de datos deben ser de tipo numérico.

1.5 Justificación y beneficios

La falta de una técnica de selección de variables para datos crono-valorados que tome en cuenta la evolución de la información hace de este trabajo una investigación de interés en áreas como selección de características para gestos, investigación con datos ómicos, o cualquier tipo de datos cronológicamente ligados.

1.6 Pregunta de investigación e hipótesis

¿Es posible incorporar el sentido evolutivo de los datos crono-valorados en una técnica de selección de variables? Dada la pregunta de investigación anterior, se tienen las siguientes hipótesis nulas: 1) Aún cuando se compriman los datos a un sólo valor y se utilice una técnica de selección de variables a datos puntuales, los resultados no son sobresalientes; y 2) Aún cuando se formule la sábana de Markov para redes Bayesianas dinámicas, esta no conduce a una selección de variables apropiada.

1.7 Antecedentes del proyecto

En esta sección se presenta el trabajo realizado en CENIDET que se relaciona de manera directa a la propuesta aquí presentada. Al final de la sección 1.8, estos trabajos de investigación se sintetizan junto con los trabajos del estado del arte y se muestra cómo influyen en la presente investigación.

Algoritmo para la selección de variables en descripciones crono-valoradas (Ochoa Somuano, 2016).

En este trabajo de investigación, realizado en CENIDET para tesis de doctorado, se buscó adaptar la metodología llamada *lógica combinatoria* a la selección de variables de tipo crono-valorado. La investigación está basada en el trabajo de (Ruiz-Shulcloper, Guzmán Arenas, & Martínez-Trinidad, 1999) donde se implementa un algoritmo para datos puntuales. La adaptación de Ochoa Somuano se presenta en la figura 1.2.



Figura 1.2 Adaptación hecha por Ochoa Somuano en su trabajo de investigación (Ochoa Somuano, 2016)

En el trabajo de Ochoa Somuano, los datos crono-valuados fueron evaluados primeramente en la etapa Coeficientes de Variación. Los coeficientes de variación representan el cociente entre la desviación típica y el valor de la media. Estos valores se ordenaron para aprovechar los que mayormente discriminan entre los datos. El siguiente paso fue utilizar un método que el autor llama *de derivada*, el cual se obtuvo con:

$$dy_t = y_{t+1} - y_t \quad \text{Ec. 1.1}$$

donde dy_t es la diferencia entre el dato siguiente (y_{t+1}) y el dato observado (y_t). Haciendo uso de matrices auxiliares, se establece el valor de uno cuando la diferencia es positiva y menos uno cuando la diferencia es negativa. La matriz *auxiliar* se multiplica por la Matriz de Aprendizaje Cronológica (MAC) y se obtiene una MAC *polarizada*. Después, utilizando la correlación entre variables, se obtiene la Matriz de Variables Discriminantes, la cual contiene valores de cero o uno; lo anterior gracias a un umbral de 0.8, que se establece para discriminar entre variables y su correlación. A partir de la Matriz básica se siguen los mismos pasos que en (Ruiz-Shulcloper et al., 1999) hasta la obtención de los conjuntos de variables discriminantes.

A pesar de que el autor hizo uso de los datos crono-valuados y trató de conservar la variable cronológica al polarizar la MAC, se observa que la diferencia entre los valores de dos muestras pudiera ser de cero, lo cual deja como incógnita el signo que se usaría en este caso. También, al hacer uso del proceso de coeficientes de variación, no se respeta el aspecto crono-valuado de los datos.

1.8 Estado del arte

Nota introductoria: Esta sección está apoyada en el capítulo “Feature Selection for Pattern Recognition: Upcoming Challenges” publicado por el autor del presente documento durante el periodo comprendido a la investigación. Dicho capítulo es parte del libro “Advanced Topics on Computer Vision, Control and Robotics in Mechatronics” (véase Anexo D).

Hoy, en la era del *big data*, los datos llegan en grandes formatos, desde videos de alta definición hasta publicaciones de interacción en las redes sociales, lo que dificulta que los algoritmos de reconocimiento de patrones procesen y tomen decisiones con ellos. El reconocimiento de patrones tiene como objetivo buscar en los datos regularidades que puedan catalogar automáticamente (entre otras tareas) los datos en diferentes clases.

Véase esta última afirmación con un problema específico: uno de los últimos *Smartphone* aborda la función de desbloqueo con reconocimiento facial, ya que esta tarea tiene que diferenciar un rostro de una imagen que no es un rostro, tiene que buscar características faciales, por lo que se involucra el reconocimiento de patrones.

Diferenciar imágenes faciales de imágenes no faciales requiere entrenamiento. El entrenamiento se realiza alimentando un algoritmo de reconocimiento de patrones con ejemplos representativos de

imágenes con caras e imágenes sin caras, hasta que el algoritmo pueda diferenciar entre una y otra. Este tipo de procedimiento (entrenamiento / aprendizaje) se puede implementar en muchos campos y aplicaciones que han hecho que el reconocimiento de patrones y el aprendizaje automático hayan experimentado un desarrollo extenso durante las últimas décadas (Bishop, 2016).

Durante la última década (2007-2017) ha habido una considerable cantidad de investigaciones relacionadas con la selección de variables, se encontraron más de 2000 resultados en *Google Scholar* al buscar las palabras clave "selección de variables" y al menos una de las palabras: "máquina", "Aprendizaje", "datos" y "minería". Lo cual indica que la investigación en esta área está evolucionando y aún quedan asuntos sin resolver. En la figura 1.3, se presenta un gráfico para respaldar esta información.

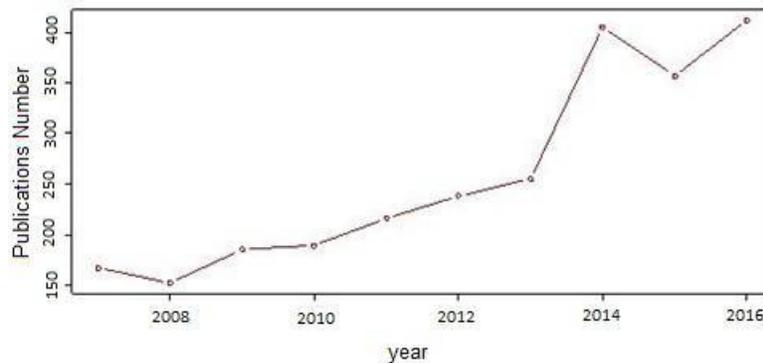


Figura 1.3 Número de publicaciones por año relacionadas con la selección de variables en la última década

Utilizando el ejemplo de reconocimiento facial en un *Smartphone*, caracterizamos los diferentes enfoques del reconocimiento de patrones para el paso de entrenamiento:

- Aprendizaje supervisado: se suministra diferentes ejemplos previamente etiquetados al algoritmo de aprendizaje; es decir, imágenes de caras y ejemplos de imágenes sin caras, diciéndole cuál es cuál, para que aprenda a reconocer cómo es una cara.
- Aprendizaje no supervisado: se introduce en el algoritmo imágenes de dos tipos, por ejemplo, caras y árboles, sin decir cuál es cuál. Por lo tanto, el algoritmo tiene que encontrar el patrón de cada clase debido a las características significativas de cada clase.
- Semi supervisado: se proporciona ejemplos de caras y algunas otras clases, pero no todas tienen la etiqueta de ser o no caras.

Proporcionar ejemplos suena tarea fácil, pero a veces estos ejemplos podrían superar el algoritmo de aprendizaje, ya sea con un exceso de variables (qué definiciones hacen que una cara parezca una cara) o con definiciones que no aportan suficiente significado al proceso de clasificación y podrían, al final, confundir la tarea de aprendizaje. Aquí es donde las técnicas de selección de variables son cruciales. La

selección de variables tiene como objetivo seleccionar un subconjunto de características¹ que se reduce en número en comparación con el original (sub-conjunto de variables), pero aún mantiene la separabilidad de clases. Para el reconocimiento facial, el número de ojos es una característica irrelevante si el objetivo es diferenciar entre el propietario del teléfono y otra persona; así, la variable *número de ojos* se puede ignorar en la lista de variables relevantes.

Se puede categorizar los métodos de selección de variables por la forma en que interactúan con los algoritmos de clasificación (A R Webb & Copsey, 2011). Este ordenamiento propone tres categorías:

1. Métodos de filtrado: estos métodos utilizan la descripción estadística para evaluar la relevancia de un atributo con otro sin tener en cuenta el algoritmo que se utilizará para una clasificación posterior.
2. Métodos de envoltura: este enfoque depende del clasificador ya que se vincula el rendimiento de este con el del algoritmo de aprendizaje.
3. Métodos integrados: tiene como objetivo incluir la selección de características en el proceso de formación. Por tanto, este método también depende del clasificador.

Para una categorización de enfoques para la selección de características, según los datos, ver el trabajo de Li (J. Li et al., 2017). En el presente documento, se trabaja sobre la categorización: 1) datos puntuales y 2) datos crono-valuados. La motivación para ordenar los enfoques en estas categorías proviene de la evolución de la recolección de datos. Muchos autores tratan los datos de la manera clásica (Hancer, Xue, Karaboga, & Zhang, 2015; He, Cai, & Niyogi, 2006; Krishna & Aramudhan, 2014; Murty & Devi, 2011; Sridevi & Murugan, 2014; Zou, Ni, Zhang, & Wang, 2015) donde los datos se dan en un formato como se muestra en Figura 1.4 (a) con una sola dimensión por observación. En el ejemplo de reconocimiento facial, la etapa de entrenamiento está dada por imágenes individuales de rostros o no rostros. Por otro lado, los datos crono-valuados vienen dados por múltiples observaciones por rostro; hay movimiento que representa acción, en este caso el reconocimiento lo hará el dueño del teléfono diciendo, por ejemplo, la palabra “hola”. En la figura 1.4 se muestra una descripción de los datos puntuales y una de datos crono-valuados. Además, en la figura 1.4 (b) se puede ver que los datos crono-valuados no solo no tienen la misma dimensión entre ellos, tienen que preservar el orden en la muestra y tiene una variable de tiempo.

¹ En este trabajo se utiliza como sinónimos los términos: variable, característica, definición y atributo

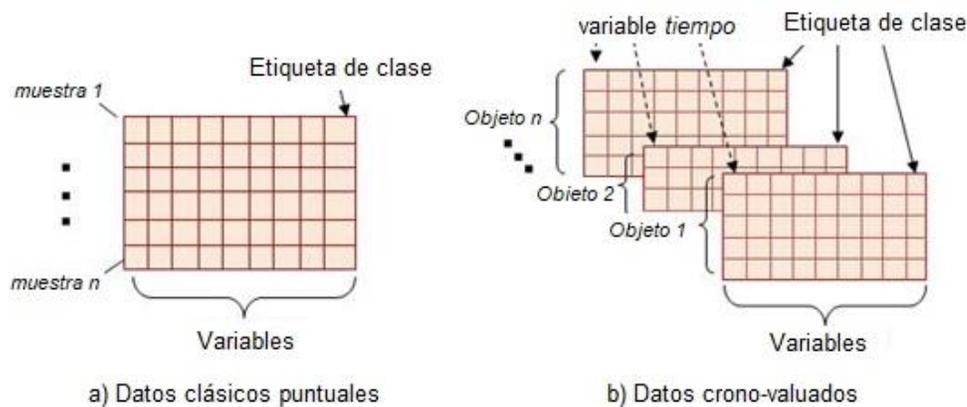


Figura 1.4 Representación visual de los datos puntuales (a) y los vinculados cronológicamente (b) como se presenta en (Cervantes Salgado & Pinto Elías, 2018)

1.8.1 Estado del arte y su contexto teórico

La selección de características ha tenido una mayor cantidad de investigación porque todos los días los dispositivos tecnológicos necesitan realizar algún tipo de tarea de reconocimiento de patrones / aprendizaje automático. Las aplicaciones son muchas y los algoritmos para hacer frente a esta exigente actividad están evolucionando; desde enfoques de datos puntuales hasta metodologías de datos vinculados cronológicamente. Se presentará una revisión del estado del arte; dividiéndolo en las metodologías en las que se basan los estudios para realizar la selección de características. Esta revisión del estado del arte no pretende ser una guía completa para realizar la selección de variables, sino buscar brechas para ofrecer un gran panorama de cómo se podrían abordar el problema de esta investigación (datos crono-valorados).

1. Métodos estadísticos

La estadística descriptiva utiliza medidas para describir datos; estas medidas tienen la intención de dar respuesta, entre otras, a preguntas como: ¿Con qué tipo de datos se está tratando? ¿Qué tan dispersos están los datos? Las medidas de tendencia central intentan dar una idea del tipo de datos dentro de un atributo y las mediciones de dispersión describen que tan dispersos están los datos dentro de un atributo (Ayala, 2015). Esta información es la línea base de selección de variables con los métodos de estadística. La selección de variables con métodos estadísticos compara los atributos sin considerar el algoritmo de clasificación, por lo que la mayoría de estos métodos se consideran métodos de selección de características de filtro (Chandrashekar & Sahin, 2014).

I.1 Medidas estadísticas

Hay algunos conceptos que necesitan descripción para facilitar la comprensión del resto de la sección. Estos comprenden las medidas: media, varianza y desviación estándar.

- Media. Es una de las medidas de tendencia central más comunes utilizadas en estadística, ayuda a localizar los datos. Para obtener el valor medio de una variable, se usa

$$\tilde{x} = \sum_{i=1}^n \frac{x_i}{n} \quad \text{Ec. 1.2}$$

donde x_i son los valores en el conjunto, cuya dimensión se expresa por n . El término media se puede expresar como promedio, lo cual da una explicación clara de la ecuación.

- Varianza. Esta medida se utiliza para conocer la dispersión de los valores dentro de una variable. Por lo tanto, la varianza mide cuánto difieren los valores de una variable de la media (Myatt, 2007), la fórmula de la varianza es

$$s^2 = \frac{\sum_{i=1}^n (x_i - \tilde{x})^2}{n} \quad \text{Ec. 1.3}$$

donde n representa el total de observaciones, difiere cuando se usa la fórmula para calcular una muestra.

- Desviación Estándar. La desviación estándar es la raíz cuadrada de la varianza y está dada por

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^n (x_i - \tilde{x})^2}{n}} \quad \text{Ec. 1.4}$$

donde x_i es el valor real de los datos, \tilde{x} es la media de la variable y n es el número de observación. Cuanto mayor sea el valor de s , más ampliamente dispersos están los valores de los datos de la variable desde y hacia la media.

I.2 Estado del arte de los métodos basados en estadística

En lugar de dar un tutorial completo de cómo se hace cada método, se pretende profundizar en la metodología para captar cualquier posibilidad de usar el método con datos crono-valorados.

Feature Selection for Abnormal Driving Behavior Recognition Based on Variance Distribution of Power Spectral Density (Nassuna, Kim, Eyobu, & Lee, 2020)

En este trabajo, los autores utilizan un método en dos etapas: a) la primera etapa transforma los datos temporales en lineales utilizando *short-time Fourier Transform* y b) la selección de variables se realiza utilizando la información de la varianza. Ver figura 1.5 donde se describe el algoritmo.

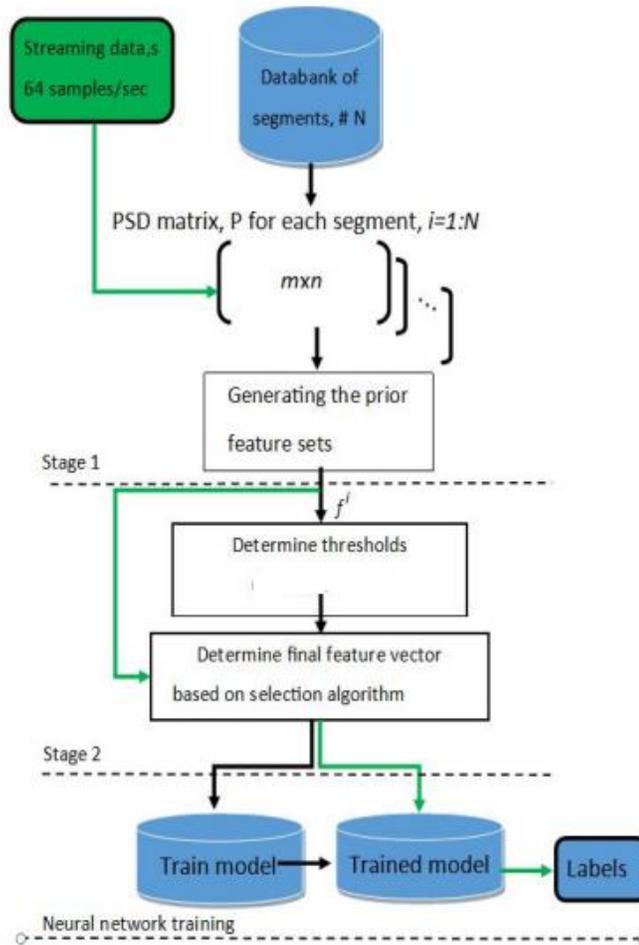


Figura 1.5 El flujo general del algoritmo propuesto en (Nassuna et al., 2020)

Aunque el trabajo mostrado toma en cuenta la evolución en el tiempo, el procedimiento que se presenta requiere que se tengan objetos de la misma longitud a través de todas las clases.

A simple low variance approach (Pedregosa et al., 2011).

La selección de variables se puede lograr utilizando estadísticas descriptivas más simples. Se han desarrollado herramientas que realizan la selección de características eliminando variables con baja variación cuyo valor no alcanza algún umbral. En el trabajo mencionado, los autores abordan las variables booleanas como variables aleatorias de Bernoulli, donde la varianza está dada por

$$var[X] = p(1 - p) \tag{Ec. 1.5}$$

donde p es el porcentaje de unos (o ceros), y la discriminación se puede hacer estableciendo un umbral, si la varianza no alcanza el umbral, se eliminará.

Se puede esperar, en este tipo de enfoque, que las variables con valor de varianza cero sean las primeras opciones a eliminar porque esto implica que la variable o característica tiene el mismo valor en todas las

muestras. Como se observa, este tipo de discriminación tiene poco que ver con la tarea de clasificación que se busca en el presente estudio.

II. Métodos basados en la teoría de la información

Considere una variable aleatoria discreta (por decir x) y luego piense en la cantidad de información que se recibe cuando se selecciona un valor específico dentro de ella, esto se llama "grado de sorpresa" (Bishop, 2016). Se comienza esta sección presentando algunos conceptos básicos de la teoría de la información antes de explorar diferentes desarrollos que la utilizan para buscar un subconjunto de características que preserven la capacidad de separabilidad entre clases (selección de variables).

II.1 Conceptos básicos

La correlación, un valor que mide la cantidad de interdependencia entre variables, es uno de los enfoques más simples para la selección de variables (A R Webb & Copsey, 2011).

- Correlación de Pearson: mide el grado de correlación lineal entre dos variables. La correlación de Pearson está dada por

$$\rho(X, Y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{[\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2]^{\frac{1}{2}}} \quad \text{Ec. 1.6}$$

donde X e Y son dos variables y \bar{x} y \bar{y} sus respectivas medias. Si dos variables están completamente correlacionadas, entonces una es redundante, por lo que podría eliminarse.

- La información mutua es una medida de correlación no lineal. Uno de sus conceptos clave es la entropía, y se define por

$$H(X) = - \sum_x p(x) \log_2(p(x)) \quad \text{Ec. 1.7}$$

aquí, la variable X tiene que tener un conjunto discreto de valores, una probabilidad asociada $p(x)$. La probabilidad de un valor viene dada por la frecuencia de ese valor en la muestra.

- La entropía de X dada Y se define como

$$H(X|Y) = - \sum_y p(y) \sum_x p(x|y) \log_2(p(x|y)) \quad \text{Ec. 1.8}$$

La información mutua es la información adicional sobre X proporcionada por Y y es la disminución de la entropía de X , definida por

$$MI(X|Y) = H(X) - H(X|Y) \quad \text{Ec. 1.9}$$

Esto podría usarse para categorizar variables para una mayor discriminación (A R Webb & Copesey, 2011). Una característica X está más correlacionada que una característica Z con la variable de clase Y si

$$MI(X|Y) > MI(Z|Y) \quad \text{Ec. 1.10}$$

La información y los conceptos presentados anteriormente darán una idea de cómo funcionan estas familias de métodos.

II. 2 Métodos basados en la teoría de la información: estado del arte

Tener un conjunto de datos que describa imágenes de rostros o no rostros con cientos de atributos requerirá que los algoritmos de reconocimiento de patrones trabajen más e incluso se confundan con algunos de ellos. Usando métodos de teoría de la información se puede reducir la cantidad de variables para entrenar el algoritmo de reconocimiento de patrones. A continuación, se explora cómo se utilizan y el tipo de datos que pueden manejar.

Differential evolution for filter feature selection based on information theory and feature ranking (Hancer, Xue, & Zhang, 2018)

En este trabajo, los autores desarrollaron un enfoque de selección de variables tipo *filter* basado en técnicas de la teoría de la información, de la clasificación de las variables y la computación evolutiva. Para llevar a cabo este enfoque, los autores propusieron la metodología de evaluación de variables (selección de variables tipo *filter*) MIRFFS. MIRFFS se basa en conceptos de relevancia mutua (marcadores *Relief* y *Fisher* (J. Li et al., 2017)) y utilizando este criterio se desarrollaron los estándares de evolución diferencial y evolución diferencial multi-objetivos. La evolución diferencial es un algoritmo de búsqueda propuesto en 1997 (Storn & Price, 1997). La evolución diferencial multi-objetivos se encarga de optimizar más de una función simultáneamente. En la figura 1.6 se muestra el diagrama del enfoque de este trabajo.

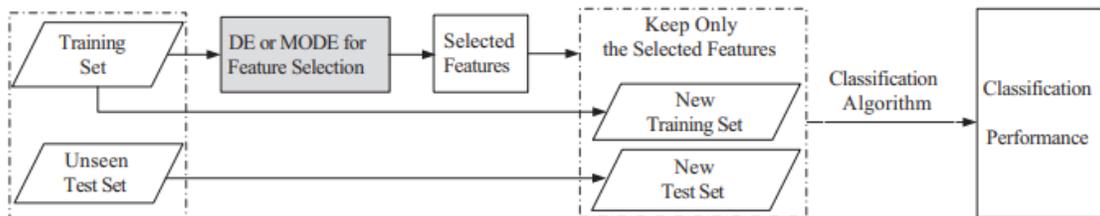


Figura 1.6 Estructura general del enfoque usado en (Hancer et al., 2018)

Ya que la información mutua no se puede utilizar en *datasets* con información continua, el uso de enfoques de la teoría de la información a los datos crono-valorados necesitan que los datos sean resumidos en una categoría.

Feature selection based on information theory for pattern classification (Krishna & Aramudhan, 2014).

En esta investigación, los autores presentan un algoritmo de selección de variables y luego crean criterios de información mutua para lograr el objetivo. Para crear los criterios, utilizaron métodos de estimación de entropía. El algoritmo se enumera en los siguientes pasos: 1. Cálculo de la información mutua. 2. Cálculo de la relación de ganancia de información, que mide la correlación entre la clase y una característica. La relación de ganancia de información viene dada por:

$$IG_R(C, F_i) = \frac{I(C, F_i)}{H(F_i)} \quad \text{Ec. 1.11}$$

donde $FS = \{x_1, x_2, x_3, \dots, x_n\}$ es el subconjunto de tamaño n y C representan la etiqueta de clase. $H(F)$ es la entropía y se puede calcular con la ecuación 1.8.

3. Descomponer FS utilizando la regla de la cadena de información mutua dada por

$$IG_R(C, F_i) = \frac{I(C, F_i)}{H(F_i)} \quad \text{Ec. 1.12}$$

4. Eliminar las funciones que no proporcionan información adicional sobre la clase.

La importancia de la estimación completa de la información mutua se aprecia cuando se emplea como criterio de selección de variables. Aunque esto parece una tarea simple, los autores sólo lo implementan con un conjunto de datos de 500 muestras y concluyen que, si aumenta el número de muestras, el tiempo de cálculo también aumentará.

Feature selection based on correlation (Sridevi & Murugan, 2014)

Los autores, logran encontrar una clasificación adecuada para el diagnóstico de cáncer de mama mediante el método combinado: correlación aproximada del conjunto y selección de variables.

El método propuesto combina dos métodos en un algoritmo de selección de variables de dos pasos. El primer paso selecciona un subconjunto de características y esto se hace aplicando el algoritmo: conjunto aproximado a datos discretos. El conjunto resultante $R1$ se establece con la variable de valor de correlación más alto, luego con un atributo con correlación promedio ($R2$) y por tercera vez ($R3$) se realiza con el atributo correlacionado más bajo. El algoritmo de selección de variables de conjunto aproximado utiliza el algoritmo QuickReduct (Hassanien, Suraj, Slezak, & Lingras, 2007). El algoritmo se puede visualizar en la figura 1.7.

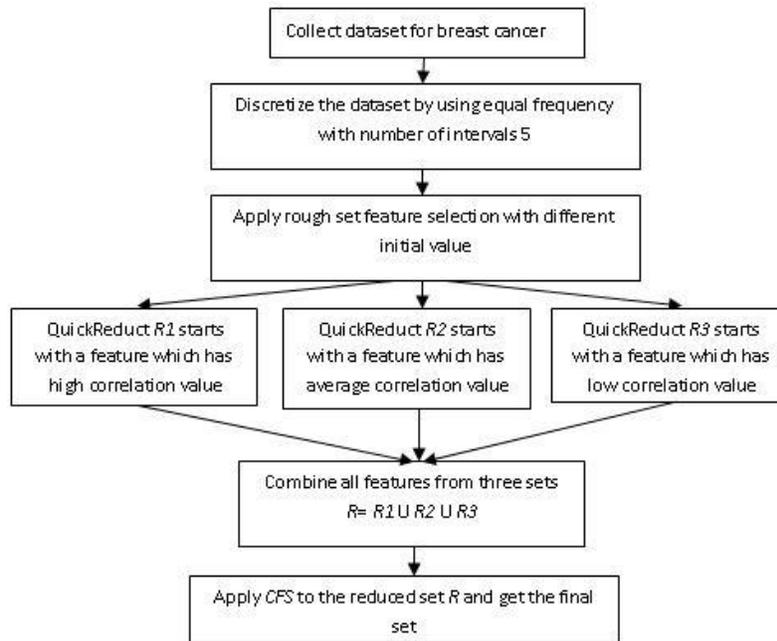


Figura 1.7 Descripción visual del método de selección de características de conjunto aproximado de correlación propuesto por (Sridevi & Murugan, 2014)

Finalmente, el segundo paso consiste en una nueva selección de los subconjuntos $R1$ a $R3$ usando la selección de variables de correlación. Como conclusión, el autor afirma que este algoritmo alcanza el 85% de exactitud (se usara el termino en inglés *accuracy* de ahora en adelante) de clasificación.

Lograr una selección de variables de datos crono-valorados con esta familia de métodos, implica el uso de técnicas probabilísticas como los modelos de Markov.

III. Métodos basados en similitudes

Los métodos basados en similitud seleccionan un subconjunto de características donde se puede preservar la similitud por pares (Liu, Wang, Zhang, Yin, & Liu, 2013). Dentro de estos métodos hay dos conceptos clave: 1) similitud de muestras por pares y 2) estructura geométrica local de datos. Estos conceptos y su apoyo teórico se describen en la siguiente sección.

III.1 Conceptos básicos de similitud

Como se indicó en la última sección, los métodos basados en similitudes se relacionan con dos conceptos importantes: similitud por pares y estructura geométrica. A continuación, se explican los conceptos para una mejor comprensión de los trabajos de investigación que siguen al presente apartado.

Similitud de muestra por pares. Se refiere a la similitud entre cada dos muestras en un conjunto de datos. La similitud da la estimación de cuán cercanas o similares son dos muestras entre sí. Muchas medidas de

distancia / similitud están disponibles en la literatura para comparar dos muestras de datos; a continuación, se presentará una lista de medidas de distancia comunes.

- Distancia euclidiana. La distancia euclidiana entre el i -ésimo y el j -ésimo objeto, como se presenta en (State, 2017), es

$$d_E(i, j) = \left(\sum_{k=1}^p (x_{ik} - x_{jk})^2 \right)^{\frac{1}{2}} \quad \text{Ec. 1.13}$$

por cada par de observaciones (i, j) .

- Distancia de Minkowski. En (State, 2017) la distancia de Minkowski se denota como

$$d_M(i, j) = \left(\sum_{k=1}^p |x_{ik} - x_{jk}|^\lambda \right)^{\frac{1}{\lambda}} \quad \text{Ec. 1.14}$$

donde $\lambda \geq 1$. Puede ser llamada la métrica $L\lambda$ dadas:

$\lambda = 1$: $L1$ métrica, *Manhattan or City-block distance*.

$\lambda = 2$: $L2$ métrica, *Euclidean distance*.

$\lambda \rightarrow \infty$: $L\infty$ métrica, *Supremum distance*. Donde

$$\lim_{\lambda \rightarrow \infty} \left(\sum_{k=1}^p |x_{ik} - x_{jk}|^\lambda \right)^{\frac{1}{\lambda}} = \max(|x_{i1} - x_{j1}|, \dots, |x_{ip} - x_{jp}|) \quad \text{Ec. 1.15}$$

- Similitud entre dos variables binarias. Las últimas tres definiciones de distancia son adecuadas para variables continuas. Pero para las variables binarias, se necesita un enfoque diferente. Primero se tiene que converger en la notación, sean p y q dos muestras binarias, la tabla 1.1 muestra todas las combinaciones posibles a sus valores que se utilizarán para encontrar la distancia entre ellas.

Tabla 1.1 Co-ocurrencia para variables binarias

	q=1	q=0
p=1	n1,1	n1,0
p=0	n0,1	n0,0

Luego, para encontrar la distancia entre las muestras p y q , hay dos medidas (State, 2017): la coincidencia simple y el coeficiente de Jaccard.

- Coincidencia simple, SMC, el coeficiente viene dado por

$$SMC = \frac{(n_{1,1} + n_{0,0})}{(n_{1,1} + n_{1,0} + n_{0,1} + n_{0,0})} \quad \text{Ec. 1.16}$$

- El coeficiente de Jaccard, J , viene dado por

$$J = \frac{n_{1,1}}{(x_{1,1} + x_{1,0} + x_{0,1})} \quad \text{Ec. 1.17}$$

Estructura geométrica local de datos. A menudo, en la literatura (He et al., 2006; Liu et al., 2013), la estructura geométrica de los datos se realiza con un gráfico, donde cada muestra se trata como un nodo y se coloca una arista entre dos muestras si son vecinas. Para averiguar si dos nodos son vecinos, se puede usar la información de la etiqueta (atributo clase para el aprendizaje supervisado) o se podría usar el algoritmo *k-Nearest Neighbor* (kNN vecino más cercano) (He et al., 2006). Usando el algoritmo kNN se coloca una arista entre los nodos i y j si x_i y x_j están "cerca".

Véase al ejemplo dado en (Murty & Devi, 2011) y su gráfico resultante. Supóngase que el conjunto de entrenamiento consta de dos variables, dieciocho muestras y tres clases, como se muestra en la tabla 1.2.

Tabla 1.2 Conjunto de entrenamiento utilizado en (Murty & Devi, 2011)

	var1	var2	class		var1	var2	class		var1	var2	class
x1	0.8	0.8	1	x2	1	1	1	x3	1.2	0.8	1
x4	0.8	1.2	1	x5	1.2	1.2	1	x6	4	3	2
x7	3.8	2.8	2	x8	4.2	2.8	2	x9	3.8	3.2	2
x10	4.2	3.2	2	x11	4.4	2.8	2	x12	4.4	3.2	2
x13	3.2	0.4	3	x14	3.2	0.7	3	x15	3.8	0.5	3
x16	3.5	1	3	x17	4	1	3	x18	4	0.7	3

Usando la información de la etiqueta del atributo clase del conjunto de datos, se obtiene un gráfico que se parece al de la figura 1.8, donde los nodos conectados (muestras) pertenecen a la misma clase.

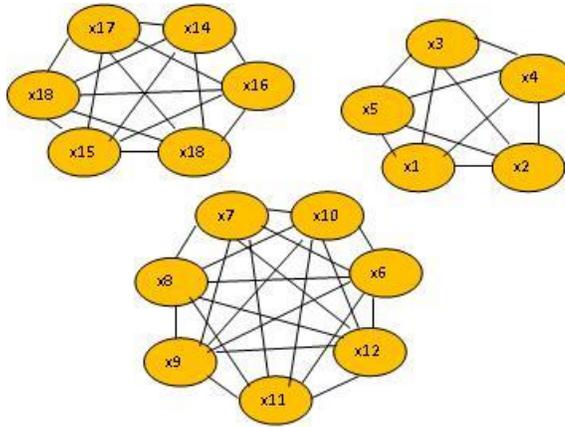


Figura 1.8 Estructura geométrica del conjunto de entrenamiento de ejemplo

Hasta ahora, se han cubierto algunos conceptos básicos de los métodos basados de similitud sin hacer ninguna selección de características. Hasta este punto, los conceptos de similitud que se presentaron sólo muestran cuán cerca están las observaciones entre sí. Por lo tanto, el trabajo de investigación que sigue dará una idea de cómo se pueden utilizar para el propósito que se persigue.

III.2 Métodos basados en similitudes: estado de la técnica

Los métodos de esta familia evalúan la importancia de las características por su capacidad para preservar la similitud de datos entre dos muestras (J. Li et al., 2017). Es decir, tienen la propiedad de seleccionar un subconjunto de atributos donde se puede preservar la similitud por comparación de partes iguales (He et al., 2006). Para el ejemplo del reconocimiento facial, las muestras de imágenes de rostros tienen valores cercanos cuando hay un rostro en la imagen, y mientras haya un rostro, de otra manera los valores serían diferentes. A continuación, se describirá el estado del arte de los métodos pertenecientes a este tipo.

Modified Dynamic Time Warping Based on Direction Similarity for Fast Gesture Recognition (Choi & Kim, 2018)

En este trabajo, los autores propusieron un Sistema de reconocimiento de gestos comparando las secuencias de las entradas de la posición de los gestos contra las entradas de los datos ya clasificados. Para la selección se utilizan las características de los movimientos del gesto utilizando el radio de la distancia Euclidiana y la distancia proporcional para calcular la diferencia. La ecuación 1.18 es la ecuación que los autores proponen para calcular la distancia entre dos elementos.

$$d(w) = (1 - \alpha)d(i, j) + \alpha s(i, j)d(i, j) \quad \text{Ec. 1.18}$$

Donde la función d calcula la distancia Euclidiana de los dos elementos de la secuencia. La función s calcula la distancia de la dirección y se computa,

$$s(i, j) = 1 - \frac{a'_j \times b'_j}{\|a'_j\| \|b'_j\| + \epsilon} \quad \text{Ec. 1.19}$$

que devuelve cero si la dirección de los dos vectores coincide y devuelve dos si las direcciones son opuestas la una a la otra. ε es una constante para prevenir que el denominador se haga cero, y α en la ecuación 1.19, es un coeficiente constante que ajusta el tamaño de la distancia direccional y la distancia euclidiana. El valor óptimo de α es cambiado en base a las características del *dataset*.

La técnica presentada en este trabajo hace uso de *interpolación* de datos cuando los objetos son de diferente longitud.

Feature Selection: Multi-Source and Multi-view Data Limitations, Capabilities and Potencial (Cherrington et al., 2019)

En este trabajo los autores presentan un censo de los trabajos de selección de variables a datos multi-fuente y multi- vista. De acuerdo a los autores, los métodos predominantes en la selección de variables a los datos multi-fuente son los basados en similitud ya que, dado que las fuentes son diferentes, se considera la similitud local de las variables al mismo tiempo que la similitud global del dataset.

El método utilizado en este trabajo para la selección de variables es capaz de modelar datos de múltiples muestras, pero no se ocupan del sentido evolutivo de los objetos.

A binary ABC algorithm based on advanced similarity scheme for feature selection (Hancer et al., 2015).

El objetivo principal de esta investigación fue proponer una variante del algoritmo binario discreto ABC (DisABC) para la selección de variables. La variante consiste en introducir un mecanismo de vecindad basado en la evolución diferencial (DE) de la búsqueda basada en similitudes de DisABC. Los principales pasos de la investigación se resumen en la siguiente lista:

1. Elija tres muestras vecinas y llámelas X_{r1} , X_{r2} y X_{r3} .
2. Calcule $\phi \times \text{Disimilaridad}(X_{r2}, X_{r2})$ usando

$$\text{Disimilaridad}(X_i, X_k) = 1 - \text{Similitud}(X_i, X_k) \quad \text{Ec. 1.20}$$

donde la similitud (X_i, X_k) representa el coeficiente de Jaccard, que se definió en la sección anterior. Y ϕ es un factor de escala aleatorio positivo.

3. Se resuelve la ecuación

$$\min \left\{ 1 - \frac{M_{11}}{M_{11} + M_{10} + M_{01}} - \Phi \times \text{Dissimilarity}(X_{r2}, X_{r3}) \right\} \quad \text{Ec. 1.21}$$

para obtener los valores M entre Ω_i y X_i

Haciendo uso de los valores M obtenidos, aplicar una selección aleatoria para obtener Ω_i

Establecer las recombinaciones entre Ω_i y X_i para obtener la nueva solución U_i , usando la ecuación

$$u_{id} = \begin{cases} \omega_{id}, & \text{si } \text{rand}(d) \leq CR \\ x_{id}, & \text{de lo contrario} \end{cases} \quad \text{Ec. 1.22}$$

donde CR es la tasa de cruce y ω_{id} representa la i -ésima dimensión de Ω_i

6. Elija la mejor solución entre X_i y U_i .

Según los autores y los resultados mostrados en el estudio, la combinación del mecanismo de búsqueda de similitudes basado en DE en el algoritmo DisABC mejora la capacidad del algoritmo en la selección de variables debido a su capacidad para eliminar características redundantes. El estudio se realizó con diferentes conjuntos de datos en (Lichman, 2013), donde los datos son puntuales.

Laplacian Score for Feature Selection (He et al., 2006)

El *Laplacian Score* es un algoritmo que se puede utilizar para la selección de variables en aprendizajes supervisados o no supervisados. Este algoritmo selecciona características que pueden preservar la separabilidad de los datos de las clases basándose en la observación de que los datos de la misma clase suelen estar cerca unos de otros. La importancia de una variable se evalúa por su poder de conservación de la localidad.

El algoritmo se basa en las siguientes afirmaciones: L_r denota la evaluación laplaciana de la r -ésima variable. f_{ri} denota la i -ésima muestra de la r -ésima variable, $i = 1, \dots, m$ y consta de los siguientes pasos:

1. Construya un gráfico vecino más cercano G con m nodos.
2. Obtenga la matriz S ; si los nodos i y j están conectados, establezca

$$S_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}} \quad \text{Ec.1.23}$$

donde t es una constante adecuada. De lo contrario, $S_{ij} = 0$.

3. Defina para la r -ésima variable

$$f_r = [f_{r1}, f_{r2}, \dots, f_{rm}]^T, D = \text{diag}(S1), 1 = [1, \dots, 1]^T, L = D - S \quad \text{Ec. 1.24}$$

4. Obtenga la evaluación laplaciana de la r -ésima variable usando

$$L_r = \frac{\tilde{f}_r^T L \tilde{f}_r}{\tilde{f}_r^T D \tilde{f}_r} \quad \text{Ec. 1.25}$$

donde \tilde{f}_r es dado por

$$\tilde{f}_r = f_r - \frac{f_r^T D 1}{1^T D 1} 1 \quad \text{Ec. 1.26}$$

Este método es un enfoque de filtro y se probó utilizando dos conjuntos de datos formados por datos puntuales. Como conclusión, este método se basa en la observación de que la estructura geométrica local es crucial para la discriminación. Los métodos de selección de características basados en similitudes se pueden aplicar al aprendizaje supervisado o no supervisado.

Feature selection for gesture recognition in Internet-of-Things for healthcare (Cisotto, Capuzzo, Guglielmi, & Zanella, 2020)

El trabajo aquí citado muestra un algoritmo para selección de variables en gestos para pacientes en cuidados hospitalarios. Los autores proponen un algoritmo que se desarrolla en dos pasos: 1) reducir el número de variables dejando las que brindan mayor información usando *Consensus clustering*, y 2) seleccionar las variables que den un desempeño mejor al clasificarlas, utilizando *Nested cross-validation*.

Como parte de la selección de variables, los autores utilizan *Consensus clustering* en cuatro pasos (ver figura 1.9):

- 1) Aplican dos algoritmos de agrupación a todo el *dataset* de manera individual
- 2) Evalúan los datos entre los dos algoritmos: utilizando matrices de similitud para cada algoritmo de agrupamiento, indicando si dos variables están asignadas al mismo grupo. Después se toma el promedio de la similitud de las variables. Finalmente se aplican umbrales para identificar sub conjuntos de variables.
- 3) Se jerarquizan los resultados de los grupos del paso anterior
- 4) Se obtienen los *centroides* de los grupos para formar el conjunto final de variables.

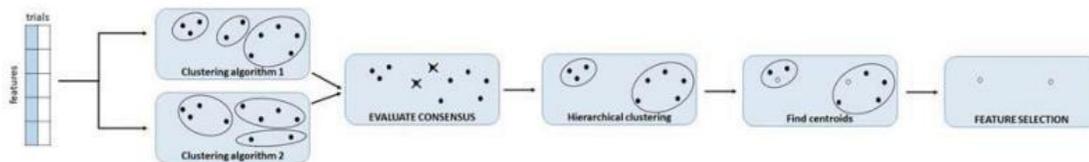


Figura 1.9 Interpretación gráfica de Consensus Clustering en (Cisotto et al., 2020)

En este trabajo se computa la distancia entre objetos para agrupar los objetos con distancias similares; sin embargo, no puede distinguir entre atributos con los mismos valores en sentido contrario.

IV. Redes Neuronales Artificiales

Con los crecientes avances en los dispositivos tecnológicos que capturan video/imágenes de alta resolución, los algoritmos de reconocimiento de patrones han sido desafiados debido a la maldición de la dimensionalidad (Svensén & Bishop, 2007). Hasta ahora se han presentado tres metodologías que pueden abordar la selección de variables, sin embargo, no todas pueden manejar datos crono-valorados. Ahora es el turno de las redes neuronales, más adelante se discutirá si pueden alcanzar el objetivo de la selección de variables en datos crono-valorados.

IV.1 Conceptos básicos

Esta sección proporciona una breve revisión del perceptrón multicapa (MLP) y *Deep Belief Networks* sin la intención de ser una guía completa (consulte las referencias para obtener más información). Se cree que las redes neuronales artificiales (ANN) manejan una mayor cantidad de datos sin comprometer demasiados recursos (Svensén & Bishop, 2007). Al final de esta rápida revisión de los enfoques de la ANN, se presentará un estado del arte.

El MLP es una estructura básica de las ANN que puede tener varias capas; su configuración radica en la idea de tener las salidas de una capa conectadas a las entradas de la siguiente capa, y entre ellas tener una función de activación. Las MLP se entrenan utilizando varios métodos de retropropagación de aprendizaje por refuerzo (Curtis, Harb, Abielmona, & Petriu, 2016). MLP es un algoritmo de aprendizaje supervisado que aprende una función $f(\cdot) : R^m \rightarrow R^o$ entrenando un conjunto de datos, donde m es el número de dimensiones por entrada y o es el número de dimensiones de salida. Dado un conjunto de características $X = x_1, x_2, \dots, x_m$ y un objetivo y , puede aprender una función no lineal para clasificación o regresión. La Figura 1.7 muestra un MLP de una capa oculta (Pedregosa et al., 2011).

En la figura 1.10, las variables están representadas en el lado izquierdo. La capa oculta (la del medio) transforma los valores de la capa izquierda con una suma lineal ponderada $w_1x_1 + w_2x_2 + \dots + w_mx_m$ seguida de una función de activación no lineal $g(\cdot) : R \rightarrow R$ (función hiperbólica, por ejemplo). La capa de salida recibe los valores de la última capa oculta y los transforma en valores de salida (Pedregosa et al., 2011).

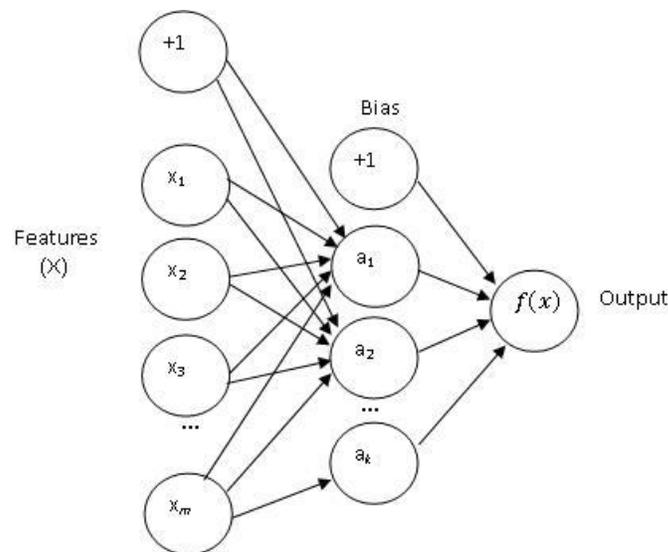


Figura 1.10 MLP de una capa oculta tal como se muestra en (Pedregosa et al., 2011)

Para modelar una tarea de clasificación usando un MLP, la ANN consistirá en una neurona de salida para cada clase, donde una clasificación exitosa produce un nivel de activación mucho más alto para la clase de neurona correspondiente (Curtis et al., 2016).

El aprendizaje profundo es una técnica relativamente nueva y ha atraído una gran atención (Zou et al., 2015), utiliza técnicas de particulares de Inteligencia Artificial a las que se harán referencia aquí como *Deep Belief Networks*. El procedimiento de aprendizaje profundo del *Deep Belief Network* consta de dos pasos: 1) resumen de características por capas y 2) ajuste fino del peso de reconstrucción (Hinton, Osindero, & Teh, 2006). En el primer paso, el *Deep Belief Network* utiliza una máquina de Boltzmann restringida (*restricted Boltzmann machine* RBM) para calcular la reconstrucción de pesos. Durante el segundo paso, *Deep Belief Networks* realiza una propagación hacia atrás para lograr los pesos deseados obtenidos en el primer paso (Hinton et al., 2006).

Poniendo en contexto, considérese v como la capa izquierda (capa visual) y h como capa intermedia (oculta). En una RBM, existe un concepto llamado "energía" que es una configuración conjunta de las capas visibles y ocultas y se define de la siguiente manera

$$E(v, h; \theta) = - \sum_{i,j} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j \quad \text{Ec. 1.27}$$

donde θ denota los parámetros (es decir, W, a, b); W denota los pesos entre los nodos visibles y ocultos; y a y b denotan el sesgo (bias) de las capas ocultas y visibles. La probabilidad conjunta de la configuración se puede definir como,

$$P_{\theta}(v, h) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta)) \quad \text{Ec. 1.28}$$

donde $Z(\theta) = \sum_{v, h} (\exp(-E(v, h; \theta)))$ es el factor de normalización. Combinando las dos últimas ecuaciones [10], se tiene

$$P_{\theta}(v, h) = \frac{1}{Z(\theta)} \exp\left(\sum_{i,j} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j\right) \quad \text{Ec. 1.29}$$

En las RBM los nodos visibles y ocultos son condicionalmente independientes entre sí, por eso, la distribución marginal de v respecto h se puede definir como

$$P_{\theta}(v) = \frac{1}{Z(\theta)} \exp[v^T W h + a^T h + b^T v] \quad \text{Ec. 1.30}$$

En el segundo paso de la *Deep Belief Network*, se aplica una retro-propagación en todas las capas para afinar el peso obtenido en el primer paso.

IV.2 Redes neuronales artificiales para la selección de variables: estado del arte

Teniendo en cuenta que el estudio del estado del arte realizado en este capítulo tiene que llevar al descubrimiento de las técnicas que pueden manejar la selección de variables de datos crono-valuados, se presentan las investigaciones recientes que tratan dicho aspecto en esta área. En la sección 1.8.2 se presenta un resumen de los métodos y una opinión personalizada de los enfoques que podrían manejar datos crono-valuados.

A novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks (Wu, Chen, & Li, 2018)

En este trabajo, los autores proponen un modelo de detección de intrusos en redes utilizando redes neuronales convolucionales. Ahí se usan las redes neuronales para seleccionar las variables automáticamente. El modelo propuesto se puede apreciar en la figura 1.11.

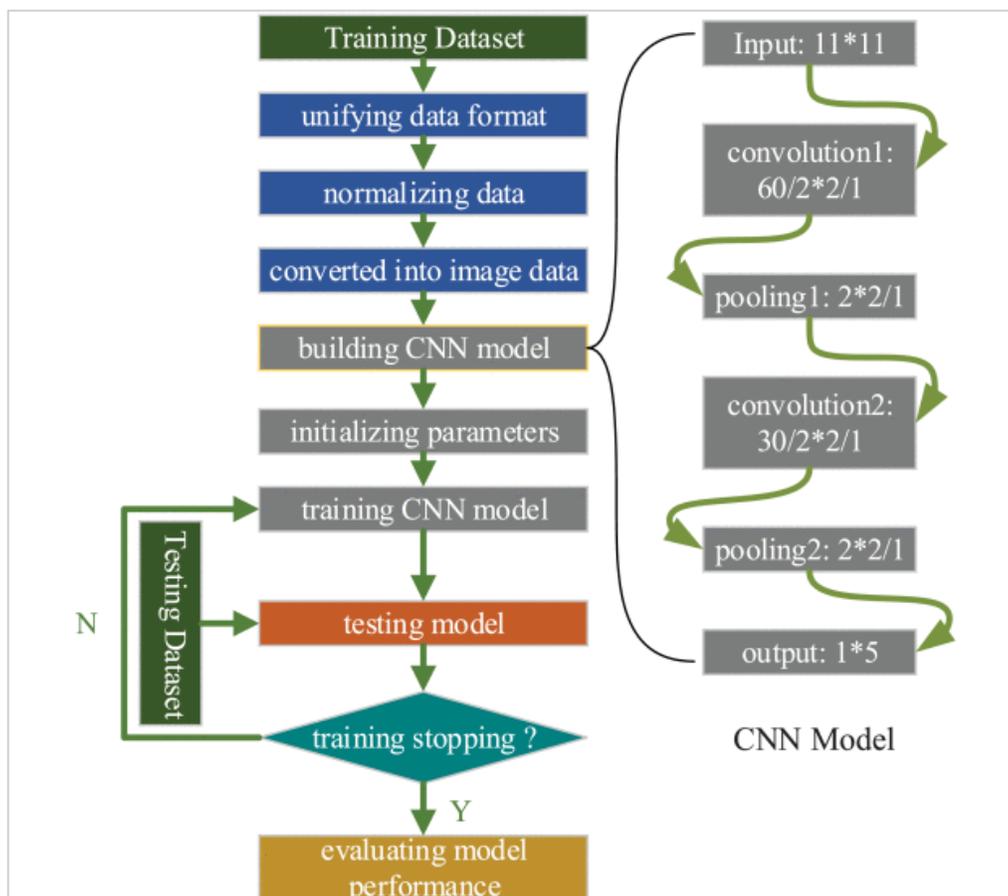


Figura 1.11 Diagrama de flujo del modelo de detección de intrusos basado en CNN en (Wu et al., 2018)

En este trabajo, la parte del pre procesamiento de datos es primordial ya que se trata de datos no lineales o puntuales. Este paso ajusta el formato de datos inicial y normaliza los valores de los datos. El pre procesamiento de datos incluye tres pasos: 1) pre procesamiento de datos simbólicos; 2) normalización de datos; y 3) convertir datos normalizados en formato de datos de imagen.

Deep learning based feature selection for remote sensing scene classification (Zou et al., 2015)

Según los autores, la selección de variables se puede lograr haciendo uso de las características más reconstruibles debido a la particularidad de mantener la característica intrínseca. Propusieron un método basado en *Deep Belief Network* con dos pasos principales. Los pasos son:

1. Aprendizaje iterativo de variables. En este paso, el objetivo es obtener ponderaciones de reconstrucción, y eso se puede hacer eliminando los valores atípicos de la variable. Los valores atípicos de la variable son aquellos con errores de reconstrucción más grandes. Se pueden identificar analizando la distribución de los errores de reconstrucción como salida del siguiente algoritmo:

a) como ingresan las entradas:

$$V = \{v_i | i = 1, 2, \dots, n\} \quad \text{Ec. 1.31}$$

Que es el vector de la variable original, η = radio de *outliers*, ε es el criterio de paro, nIt = máximo número de iteraciones.

b) iterar desde $j=1$ hasta nIt solo si:

$$\hat{e}_{j-1} - \hat{e}_j < \varepsilon \quad \text{Ec. 1.32}$$

c) finalmente, como salida obtener M , la matriz de ponderación de reconstrucción final. El error de reconstrucción de la variable v_i se define como $e = \|v'_i - v_i\|$ y el promedio de error es

$$\hat{e} = \frac{1}{n} \sum_i^n e_i \quad \text{Ec. 1.33}$$

2. Selección de variables. En este paso, la matriz de ponderaciones se utiliza para elegir las mejores variables, ya que los valores atípicos se eliminaron en el primer paso. Supongase que M es la matriz de ponderación de reconstrucción obtenida en el último paso, I es una imagen (ya que este método fue diseñado para la selección de características en imágenes) en el conjunto de datos de prueba,

$$V_N^I = \{v_i^I | i = 1, 2, \dots, N\} \quad \text{Ec. 1.34}$$

N es el número de variables extraídas de I . Como se mencionó antes, el propósito de esta investigación es seleccionar la variable con menor error de reconstrucción que están dadas por

$$V^I = \{v_i^I | e_i^I < T^I, \quad e_i^I \in E_N^I\} \quad \text{Ec. 1.35}$$

Consulte (Zou et al., 2015) para ver el conjunto completo de ecuaciones. Se puede observar que la idea principal es deshacerse de las variables que luego de pasar por la RNA, en este caso una *Deep Belief Network*, se considera que tienen un error mayor; dando una forma diferente de usar la RNA no como clasificadores sino como selectores.

Artificial Neural Networks (Murty & Devi, 2011)

En (Murty & Devi, 2011) se presenta una técnica menos complicada. Utilizando una red de retroalimentación de múltiples capas con un algoritmo de aprendizaje de retropropagación, los autores proponen la extracción del subconjunto de variables más discriminativas. Primero, se propone establecer una red más grande, luego comenzar el entrenamiento; y, a medida que avanza, recortar algunos nodos teniendo cuidado de ajustar los pesos restantes de tal manera que el rendimiento de la red no empeore durante el proceso de entrenamiento.

El criterio para eliminar un nodo, en este enfoque, viene dado por

- Se eliminará un nodo después de analizar el aumento del error causado por eliminar ese nodo específico.

El problema de la poda se fórmula en términos de resolver un sistema de ecuaciones lineales utilizando la técnica de optimización. Al igual que en el último estudio (Zou et al., 2015), el conjunto de datos considerado para este estudio específico es puntual. Dentro de este y los últimos estudios no se utilizaron ni se consideraron datos crono-valorados durante las pruebas.

V. Métodos de Sparse para la selección de variables

Dada su naturaleza, la regularización (o métodos *Sparse*) es muy adecuada para la selección de variables. Para un modelo de regularización estadístico, sólo un número relativamente pequeño de características son importantes para la variedad de datos (Hastie, Tibshirani, & Friedman, 2009). Se dice que estos métodos pueden manejar datos vinculados o datos de múltiples vistas (Tang & Liu, 2014; Wang, Nie, & Huang, 2013), por esa razón se presentarán en esta sección.

V.1 Conceptos introductorios

La regularización tiene como objetivo encontrar un modelo más simple a partir de los datos, como se explica en (Hastie et al., 2009). Para comprender este tema se tiene que introducir conceptos importantes.

En un problema de regresión lineal, sea N el número de observaciones de una variable de resultado y_i , y p las variables predictoras asociadas, $x_i = (x_{i,1}, \dots, x_{i,p})^T$. La solución al problema será predecir el resultado a partir de las características, y esto se hace para lograr dos objetivos: la predicción con datos futuros y encontrar las características que realmente son importantes para la separabilidad. Un modelo de regresión lineal supone que

$$y_i = \beta_0 + \sum_{j=1}^p x_{ij}\beta_j + e_i \quad \text{Ec.1.36}$$

donde β_0 y $\beta = (\beta_1, \beta_2, \dots, \beta_p)$ son los parámetros desconocidos, y e_i es el error.

Por lo general, el método de mínimos cuadrados se utiliza para obtener los coeficientes de los parámetros por

$$\text{minimizar}_{\beta_0, \beta} = \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \quad \text{Ec. 1.37}$$

Debido a las estimaciones de la última ecuación, que normalmente será distinta de cero, la interpretación del modelo será difícil si p es grande. Para esto, en la regresión regularizada *lasso* o *l1*, se introduce una regulación y el problema se resuelve de la siguiente manera

$$\text{minimizar}_{\beta_0, \beta} \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \text{ sujeto a } \|\beta\|_1 \leq t \quad \text{Ec. 1.38}$$

donde $\|\beta\|_1 = \sum_{j=1}^p |\beta_j|$ la norma *l1* de β , y t es el parámetro establecido para encontrar los parámetros internos.

Hasta aquí se ha mostrado la base introductoria del aprendizaje *Sparse*, a continuación, presentamos un par de estudios que darán una mejor comprensión de cómo se utiliza este método.

V.2 Metodos del aprendizaje *Sparse*: Estado del arte

Un modelo *Sparse* tiene como objetivo acercar los coeficientes de las características a cero, luego estas características se pueden eliminar. Los modelos *Sparse* son sometidos a muchas investigaciones en los últimos años (J. Li et al., 2017). Se presentan estudios recientes que llevarán a encontrar un método adecuado para manejar datos vinculados cronológicamente.

EMG Channel Selection for Improved Hand Gesture Classification (Samadani, 2018)

Este documento presenta un enfoque para identificar un conjunto mínimo de canales más destacados en los *Electromyographic signals* (EMG) para discriminar entre diferentes gestos con las manos. En el enfoque se realizan de los siguientes pasos:

- 1) codifica cada canal de observaciones EMG de series de tiempo en un HMM específico del gesto
- 2) obtiene un espacio probabilístico compartido donde las observaciones se representan en términos de distancias por pares entre los HMM específicos de la clase y del canal
- 3) calcula las dimensiones del espacio probabilístico resultante mediante regresión logística multinomial con la mínima contracción absoluta del grupo y la penalización del operador de selección (grupo Lasso),

correspondiendo cada grupo a distancias específicas del canal (distancias posteriores entre un canal y el resto de canales)

4) determina los canales salientes según las dimensiones ponderadas

5) clasifica la observación EMG según el conjunto de canales salientes.

El enfoque propuesto se puede utilizar en la fase preliminar / exploratoria de los sistemas de clasificación de gestos basados en EMG para determinar el conjunto mínimo de sensores EMG y su ubicación para un seguimiento óptimo.

Feature selectin with multi-view data: A survey (R. Zhang, Nie, Li, & Wei, 2019)

Este trabajo presenta una recopilación de los métodos de selección de variables en datos multi-vista. En él, los autores mencionan dos métodos de selección que son de mayor uso entre los trabajos de investigación recientes. Los enfoques supervisados de selección de variables tienden a relacionar los datos con la clase, para ello, los métodos tales como el Fisher Score valoran las variables de manera individual; donde la correlación de los atributos no es tomada en consideración.

Otro método utilizado para la selección de variables, de acuerdo al trabajo aquí citado, es el análisis discriminador lineal; este tipo de métodos es complementado con las penalizaciones y juntos forman los métodos *Sparsity*.

Feature Selection for Neural Networks Using Group Lasso Regularization (H. Zhang, Wang, Sun, Zurada, & Pal, 2020)

En este trabajo se propuso un método de selección de variables basado en una combinación de penalización del *Group Lasso* y redes Neuronales. La regularización propia de *Group Lasso* se utiliza para la selección de variables y los autores proponen cuatro penalizaciones para descartar los pesos adicionales necesarios para el clasificador *multi-layer perceptron* que se propuso en la metodología del mencionado trabajo.

En los trabajos relacionados a *Group Lasso*, y en general los llamados *Sparsity*, se observa que la distinción de la evolución de las variables no es soportada por la naturaleza de este tipo de métodos.

Feature Selection for social media data (Tang & Liu, 2014).

Este es un estudio de cómo los datos de las redes sociales presentan nuevos desafíos para los enfoques de selección de variables. Dado que los datos en las redes sociales tienen múltiples dimensiones, algunos enfoques para la discriminación de variables no funcionarán bien. Los datos en las redes sociales, por decir, Twitter, tienen una representación morfológica como se ve en la figura 1.12 donde los autores explican la interacción usuario-publicación y usuario-usuario. Los usuarios en las redes sociales tienen dos comportamientos: 1) seguir a otros usuarios, representados en la figura 1.12 como μ_i , y 2) generar

algunas publicaciones (la publicación es una generalización de *tweets*, *blogs* o imágenes) representadas como p_i .

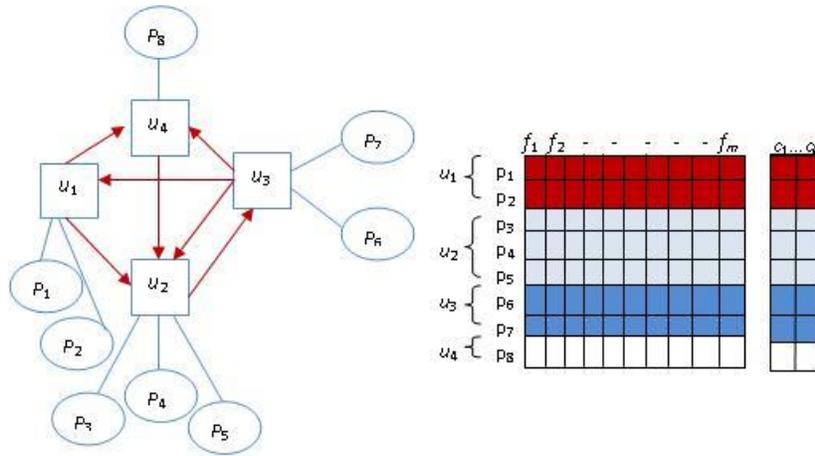


Figura 1.12 Representación visual de los datos de las redes sociales y su ilustración matricial como se muestra en (Tang & Liu, 2014).

Para modelar la hipótesis, los autores primero introdujeron la selección de variables para datos puntuales basada en la regulación de la norma $l_{2,1}$, que selecciona características a través de puntos de datos usando

$$\min_w \|X^T W - Y\|_F^2 + \alpha \|W\|_{2,1} \tag{Ec. 1.39}$$

donde $\|\cdot\|_F$ denota la norma Frobenius de una matriz y el parámetro α controla la escasez en las filas W . $W \in R^{m \times k}$ y $\|W\|_{2,1}$ es la norma $l_{2,1}$ de W y se define por

$$\|W\|_{2,1} = \sum_{i=1}^m \sqrt{\sum_{j=1}^k W^2(i,j)} \tag{Ec. 1.40}$$

Luego, los autores proponen agregar un término de regularización a la ecuación en el primer paso que fuerza la hipótesis de que las etiquetas de clase de las publicaciones del mismo usuario son similares. Y esto viene dado por

$$\min_w \|X^T W - Y\|_F^2 + \alpha \|W\|_{2,1} + \beta \sum_{u \in U} \sum_{f_i, f_j \in F_u} \|T(f_i) - T(f_j)\|_2^2 \tag{Ec. 1.41}$$

La hipótesis anterior asume que las publicaciones del mismo usuario son de temas similares. En otras palabras, las publicaciones del mismo usuario son más similares, en términos de temas, que las publicaciones seleccionadas al azar.

Dado que los autores de este estudio trabajan con datos vinculados, se podría suponer que este método podría manejar los datos vinculados cronológicamente.

Hasta aquí, se puede hacer un resumen de los métodos y el tipo de datos que utilizaron y si existe la posibilidad de ampliar el modelo para aceptar y funcionar bien con datos vinculados cronológicamente.

1.8.2 Resumen de métodos estudiados y conclusiones del Estado del Arte

En esta sección se presenta un resumen de los métodos de selección de variables presentados en la sección 1.8.1 con el fin de confirmar, o no, la posible inclusión de datos crono-valuados. El objetivo principal del capítulo uno es presentar los desafíos que enfrenta la selección de variables a medida que cambian las características de los datos.

Se presentaron en la sección 1.8.1 métodos basados en diferentes enfoques que logran la selección de variables. Ahora, se reúne la información para brindarle al lector una perspectiva de cómo esos métodos podrían usarse, o no, para la selección de variables con datos crono-valuados.

Métodos basados en estadísticas (secciones 1.7 y I.2): la investigación en (Somuano, 2016) trata datos vinculados cronológicamente, sin embargo, parece no preservar la secuencia de los mismos. El trabajo de (Pedregosa et al., 2011) utiliza la varianza más baja del conjunto de características para determinar la discriminación. Con esta técnica, la selección de variables en datos crono-valuados no se puede lograr a menos de que los objetos se comparen individualmente.

Métodos basados en la teoría de la información (sección II.2): hay dos enfoques que utilizan datos puntuales. La teoría de la probabilidad tiene técnicas que tratan con secuencias de tiempo, es decir, modelos de Markov (Svensén & Bishop, 2007), por lo que el trabajo podría surgir de estas técnicas.

Métodos basados en similitudes (sección III.2): los métodos basados en similitudes se basan en diversas métricas. Con el fin de acomodar métodos basados en similitudes para la selección de variables en datos crono-valuados, se necesitan medidas de distancias entre grupos de objetos o distribuciones (Andrew R Webb, 2003).

Selección de variables basada en redes neuronales (sección IV.2): esta familia de métodos ha utilizado datos puntuales y multi-dimensión en la literatura. Sin embargo, se necesita una investigación más profunda para explorar todas las posibilidades que ofrecen estos métodos. Además, dada la gran cantidad de investigación que se realiza en la actualidad en esta área, se considera que la solución de las hipótesis basadas en el campo de las RNA podría ser rebasada en tiempo.

Métodos *Sparse* para la selección de características (sección V.2): muestran que pueden conducir a la solución para la selección de variables en datos crono-valuados que utiliza datos vinculados y de múltiples vistas (Tang & Liu, 2014).

En la tabla 1.3, se presenta la información condensada para tener una mejor perspectiva del desafío que representan los datos crono-valuados para los enfoques de selección de variables.

Se puede concluir, a partir del análisis realizado en este estudio, que los datos vinculados cronológicamente representan un problema no resuelto y que se pueden realizar más investigación. Además, se brindan algunas pautas sobre cómo avanzar hacia el problema no resuelto.

El uso de un enfoque de extracción de características permite incorporar la propiedad de evolución de los datos crono-valorados a la transformación. Para ello, se requiere la presencia de expertos en la materia y el estudio es específico al universo de datos (Shanableh, Assaleh, & Al-Rousan, 2007)(G. Li, Zhang, Ritchie, & Griffiths, 2017)(Gupta, Jaafar, Ahmad, & Bansal, 2013).

Otra opción explorada han sido los estudios de series de tiempo (Yoon, Yang, & Shahabi, 2005), pero los datos crono-valorados no pueden tratarse como series de tiempo por las siguientes razones: los crono-valorados examinan una descripción completa del evento dentro del tiempo de observación; por tanto, el pasado o el futuro no son relevantes para describirlos. Además, los objetos de series de tiempo tienen ciertos componentes, por ejemplo: tendencia, variaciones estacionales, cíclicas y aleatorias, que pueden estar presentes o no en los crono-valorados (ejemplo de esto: movimientos humanos u objetos de interés, video vigilancia).

Tabla 1.3 Resumen de los enfoques estudiados y sus posibilidades de adaptación con datos vinculados cronológicamente.

Referencia	Método base	Tipo de datos utilizados	Comentarios
(Somuano, 2016)	Estadístico	Crono-valorado	Utiliza datos vinculados cronológicamente, sin embargo, es necesario mejorar la preservación de la secuencia.
(Nassuna et al., 2020)	Estadístico	Crono-valorados	Requiere objetos de la misma longitud
(Pedregosa et al., 2011)	Estadístico	puntual	Enfoque simple que no puede manejar datos vinculados cronológicamente
(Hancer et al., 2018)	Teoría de la información	Diversos	Requiere variables con información no continua. Pre-procesamiento para categorizar los objetos crono-valorados
(Krishna & Aramudhan, 2014)	Teoría de la información	puntual	Potencial para tratar con datos vinculados cronológicamente utilizando, por ejemplo, modelos de Markov
(Sridevi & Murugan, 2014)	Teoría de la información	puntual	Potencial para tratar con datos crono-valorados

Tabla 1.3 Resumen de los enfoques estudiados y sus posibilidades de adaptación con datos vinculados cronológicamente. Continuación.

Referencia	Método base	Tipo de datos utilizados	Comentarios
(Choi & Kim, 2018)	Medidas de Similitud	Crono- valuados	La técnica requiere objetos de la misma longitud
(Cherrington et al., 2019)	Medidas de Similitud	Multi-fuente	No se ocupa del sentido evolutivo de los datos
(Cisotto et al., 2020)	Medidas de Similitud	Crono- valuados	No distingue entre gestos en sentido contrario
(Hancer et al., 2015)	Medidas de similitud	puntual	Potencial para tratar datos crono-valuados introduciendo medidas de distancias entre grupos de objetos o distribuciones
(He et al., 2006)	Medidas de similitud	puntual	Potencial para tratar datos crono-valuados introduciendo medidas de distancias entre grupos de objetos o distribuciones
(Wu et al., 2018)	RNA	Crono- valuados	Requiere convertir los datos crono-valuados a representación puntual, se requieren objetos de la misma longitud
(Zou et al., 2015)	RNA	puntual	Potencial para tratar datos crono-valuados. Sin embargo el área es demasiado concurrida
(Murty & Devi, 2011)	RNA	puntual	Potencial para tratar datos crono-valuados. Sin embargo el área es demasiado concurrida
(Samadani, 2018)	Sparsity	Crono- valuados	No distingue el sentido evolutivo de los objetos
(R. Zhang et al., 2019)	Estadístico / Sparsity	Multi-vista	No distingue el sentido evolutivo de los objetos
(H. Zhang et al., 2020)	Sparsity	Diversos	No distingue el sentido evolutivo de los objetos
(Tang & Liu, 2014)	Sparse learning	Ligados, multi-vista	Los métodos Sparse utilizan datos ligados cronológicamente, sin embargo no son capaces de diferenciar el sentido de ellos

Para esta investigación, se examinaron las técnicas de *Group Lasso* y de *múltiples vistas*, ya que fueron señaladas por manejar estructuras y datos heterogéneos (J. Li et al., 2017). *Group Lasso* (Samadani, 2018)(Yuan, Liu, & Ye, 2011; H. Zhang et al., 2020) puede manejar datos vinculados cronológicamente, sin embargo, no puede diferenciar entre los movimientos de sentarse en una silla y pararse de ella. Las técnicas para la selección supervisada de funciones de múltiples vistas (Cherrington et al., 2019; R. Zhang et al., 2019) son capaces de modelar datos de múltiples muestras, pero no se ocupan del sentido de evolución de los gestos.

También se examinó la selección de variables con métodos estadísticos: Análisis de varianza para gestos con manos (Singha & Laskar, 2017), sin embargo en este trabajo se realiza el mismo gesto con diferente individuo, por lo que el problema a resolver, en este caso, es el número de entradas por objeto sin que la evolución en tiempo afecte.

Además se examinó la selección de variables con matrices de similitud utilizando distancia euclidiana entre elementos de los objetos (Cisotto et al., 2020). En dicho trabajo se computa la distancia entre objetos para agrupar los objetos con distancias similares; sin embargo, no puede distinguir entre atributos con los mismos valores en sentido contrario.

El objetivo de este estado del arte fue encontrar una brecha en la investigación para justificar una propuesta de datos crono-valorados para la investigación de selección de variables. Se revisó la importancia de la selección de variables para los algoritmos de reconocimiento de patrones y el trabajo realizado hasta ahora con datos puntuales y vinculados. Se proporcionó una guía rápida de conceptos básicos para introducir a los lectores en las técnicas de selección de variables. Después de resumir y analizar el estado del arte, se descubrió que el problema de los datos vinculados cronológicamente sigue sin resolverse. Se sugirieron algunas pautas sobre los métodos matemáticos apropiados para manejar el tipo de datos mencionados. Finalmente, se presentaron los desafíos de las técnicas de selección de características dada la naturaleza dinámica de los eventos (datos vinculados cronológicamente).

1.9 Descripción de la solución

Para abordar el problema, se proponen dos enfoques. El primer enfoque utiliza las redes Bayesianas (RB), así como, la sábana de Markov y el segundo la Lógica Combinatoria para el Reconocimiento de Patrones (LCRP). La contribución original en el primer enfoque es la formulación matemática para la sábana de Markov a las redes Bayesianas Dinámicas (RBD) y en el segundo es la metodología de la función de comparación requerida en el algoritmo de LCRP.

1.10 Discusión

Este trabajo de investigación tiene como meta mostrar que los dos enfoques, mencionados en la sección 1.9, pueden obtener subconjuntos que mejoren la clasificación haciendo uso de menos variables sin perder la esencia cronológica de los datos y con esto contribuir con los huecos encontrados en el área.

CAPÍTULO 2

PROPUESTA DE SOLUCIÓN

2.1 Planteamiento de la solución

La solución se planteó a partir de los alcances y limitaciones, teniendo en cuenta el objetivo general y la justificación. El planteamiento contiene dos enfoques generales: 1) Enfoque con redes Bayesianas (RB) y 2) Enfoque con Lógica Combinatoria para Reconocimiento de Patrones (LCRP).

- 1) Dentro del enfoque uno (RB) se investigaron dos diferentes soluciones: a) modelar los datos crono-valorados con RBD, seleccionando las variables más relevantes con la propuesta de sábana de Markov para RBD y b) comprimir los datos crono-valorados a una representación puntual y así modelarlos con RB, seleccionando las variables con la sábana de Markov estándar.
- 2) El enfoque dos (LCRP) hace uso de la metodología de compresión de datos utilizada en el enfoque uno, con una modificación para ajustarla al modelo de la LCRP. Dicha metodología se referirá en adelante como *Función de comparación entre objetos*. Esta función se explica en su forma teórica en la sección 2.3.2.

2.1.1 Descripción teórica de los elementos de la solución, enfoque uno - RB

El primer enfoque para la solución del problema utiliza redes Bayesianas para modelar los datos y la sábana de Markov para la selección de variables. Este enfoque, en sí mismo, se divide en dos caminos: el primero utiliza las redes Bayesianas dinámicas para modelar los datos y el segundo utiliza las redes Bayesianas estáticas.

Redes Bayesianas

Las redes Bayesianas son modelos gráficos donde los nodos representan variables aleatorias y las aristas representan las dependencias probabilísticas entre ellas (Korb & Nicholson, 2010). La estructura gráfica $\mathcal{G} = (V, A)$ de una red Bayesiana es un grafo acíclico dirigido (DAG por sus siglas en Inglés), donde V es el conjunto de nodos y A es el conjunto de arcos. El DAG define la factorización de la distribución de probabilidad conjunta de $V = \{X_1, X_2, \dots, X_v\}$, algunas veces llamada *distribución de probabilidad global*, en un conjunto de probabilidades locales, una por cada variable. La forma de la factorización es dada por la propiedad de Markov de las redes Bayesianas (Scutari, 2010), la cual dice que por cada variable aleatoria X_i , esta depende directamente sólo de sus padres $Pa(X_i)$:

$$P(X_1, \dots, X_v) = \prod_{i=1}^v P(X_i | Pa(X_i)) \quad \text{Ec. 2.1}$$

donde los nodos padres $Pa(X_i)$ son aquellos nodos a los cuales hay un arco apuntando desde el nodo de interés X_i . X_1, \dots, X_v son las variables aleatorias pertenecientes al DAG. v es el número total de variables en el DAG.

Modelo matemático de las redes Bayesianas

La correspondencia entre independencia condicional (de las variables aleatorias) y la separación gráfica (de los nodos en el grafo) ha sido extendido a una tripleta de conjuntos disjuntos de V por (Pearl, 2009) con la *d-separación*. Por lo cual los algoritmos de selección de modelos primero tratan de aprender la estructura gráfica de la red Bayesiana (de ahí el nombre de *algoritmos de aprendizaje de estructura*) y luego estimar los parámetros de las funciones de distribución local condicionadas a la estructura aprendida. Este enfoque de dos pasos tiene la ventaja ya que considera una función de distribución local a la vez, y no requiere modelar la función de distribución global explícitamente. Otra ventaja es que los algoritmos de aprendizaje son capaces de escalar para ajustar modelos altamente dimensionales sin incurrir en la llamada *maldición de dimensionalidad* (Svensén & Bishop, 2007).

Las redes Bayesianas se basan en la noción de independencia condicional; la cual define que dado un conjunto de variables, si $P(A = a | X = x, B = b) = P(A = a | X = x)$ entonces B no da información extra acerca de A , haciendo a A y B condicionalmente independientes (Koller & Friedman, 2009).

Redes Bayesianas Dinámicas

Las redes Bayesianas dinámicas (RBD) son redes probabilísticas dinámicas que modelan la evolución estocástica de un conjunto de variables en el tiempo. Al igual que las redes Bayesianas, las RBD están definidas por una estructura gráfica y un conjunto de parámetros, que juntos especifican la distribución conjunta sobre variables aleatorias.

Las RBD extienden la noción dada en la Ecuación 2.1 para modelar procesos temporales. Por simplicidad, se asume que los cambios ocurren entre puntos de tiempo discretos que están indizados por enteros no negativos. Se asume que $X = \{X_1, \dots, X_n\}$ es un conjunto de variables a las que un proceso cambia. $X_i[t]$ es la variable aleatoria que denota el valor de la variable X_i en el tiempo t , y $X[t]$ es el conjunto de variables $X_i[t]$.

Para representar las creencias de las posibles trayectorias del proceso, se necesita una distribución de probabilidad sobre las variables aleatorias $X[0] \cup X[1] \cup X[2] \cup \dots$ (Friedman, Murphy, & Russell, 1998).

Sábana de Markov

Las redes Bayesianas modelan la (in)dependencia condicional de variables aleatorias. Es decir, el DAG codifica un conjunto de realidades acerca de las relaciones entre las variables en la distribución que este modela. Los arcos representan dependencias, así el DAG es una manera simple de representar toda la información de dependencia de un sistema particular de variables (Kuschner et al., 2010).

En general una BN no tiene un nodo raíz particular; sin embargo, esta tendrá uno o más nodos que no tengan padres. Entonces, si se asume que la clase de un *dataset* es causal de la variabilidad de los atributos para un valor particular de clase, la causalidad surgirá como un vínculo en la BN (Kuschner et al., 2010).

Sabiendo la dependencia e independencia condicional entre variables, se obtiene la red Bayesiana que de forma simplificada representa la distribución de probabilidades entre variables (distribución de probabilidad conjunta). Una vez obtenida esta distribución, se utiliza la sábana de Markov para establecer cuáles variables son dependientes con la variable observada, que en este caso será la variable *clase* (Kuschner et al., 2010).

La sábana de Markov establece que un nodo es condicionalmente independiente de todos los nodos fuera del alcance de la sábana para dicha variable. Para ejemplificar esta definición referirse a la figura 2.1 donde se aprecia que la variable X es condicionalmente independiente de todos los nodos en la red menos de sus padres, hijos y los padres de sus hijos.

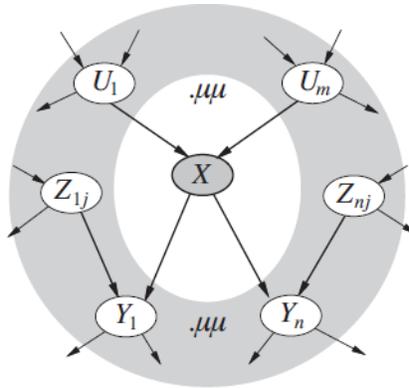


Figura 2.1 Un nodo X es condicionalmente independiente de todos los otros nodos en la red dada su sábana de Markov (área gris) (Russell & Norvig, 2013)

Por lo que se pretende utilizar esta sábana de Markov para obtener las variables más representativas (selección de variables) de la variable clase.

2.1.2 Descripción teórica de los elementos de la solución, enfoque dos - LCRP

Lógica combinatoria para el reconocimiento de patrones

La teoría de testores es una de las áreas que constituyen la base teórico-matemática del enfoque lógico matemático, se formuló como una de las direcciones científicas independientes de la cibernética matemática a mediados de los años 60, en lo que actualmente es Rusia. Su origen en 1954-1955, está vinculado a la utilización de métodos lógicos matemáticos para la localización de desperfectos en circuitos eléctricos (Ruiz-Shulcloper et al., 1999). En 1965, con los trabajos de Dmetriev Zhuravliov y Krendeleiev (Dmitriev, Zhuravlev & Krendeleiev, 1966) se abre una línea de aplicación de la teoría de testores a los problemas clásicos del reconocimiento de patrones.

La teoría de testores se ha aplicado a diferentes problemas, en la categorización de textos (Pons-Porrata, Gil-García, & Berlanga-Llavori, 2007), en la estimación de parámetros estelares (Santos, Carrasco, & Martínez, 2004), modelos de redes neuronales (Alba Cabrera & Lazo Cortés, 1998; Cortés, Díaz, & Gómez, n.d.), entre otros. Se han propuesto algoritmos para resolver problemas clásicos como el algoritmo CC (Aguila Feros & Ruiz-Shulcloper, 1984), BR (Ayaquica, 1997), CT (Bravo, 1983), BT y TB (Lazo-Cortes, Ruiz-Shulcloper, & Alba-Cabrera, 2001), entre otros (Ruiz-Shulcloper & Cortés, 1991; Ruíz-Shulcloper & others, 1985; Guillermo Sanchez-Diaz, Piza-Davila, Lazo-Cortes, Mora-Gonzalez, & Salinas-Luna, 2010; Santiesteban-Alganza & Pons-Porrata, 2003). También, se han desarrollado heurísticas para encontrar testores típicos, como los algoritmos genéticos (Alba-Cabrera, Santana, Ochoa-Rodriguez, & Lazo-Cortes, 2000; Diaz-Sanchez et al., 2011; G Sanchez-Diaz, Lazo-Cortes, & Fuentes-Chavez, 1999). Además, a partir de 1995 se han incursionado en el cálculo de testores difusos (Lazo-Cortés & Ruiz-Shulcloper, 1995; Morales-Manilla & Sanchez-Diaz, 2007).

Una de las principales ventajas de la teoría de testores es que, en caso de existir, entrega un testor típico que contiene el menor número de variables que permiten representar a un conjunto de datos (Selección de Variables).

El enfoque basado en LCRP presenta un desafío similar al presentado por las RB, se necesita una representación plana de las diferencias de los objetos (*Matriz de Diferencias* en la figura 2.2), según la versión presentada en (Martínez-Trinidad & Guzmán-Arenas, 2001). El algoritmo de la figura 2.2 se utiliza en su forma original, sin embargo, la parte de Funciones de Semejanza es remplazada por la *función de comparación entre objetos*, propuesta en esta investigación, para comprimir los datos crono-valuados.



Figura 2.2 Esquema de representación para la selección de variables con el enfoque LCRP presentado en (Martínez-Trinidad & Guzmán-Arenas, 2001)

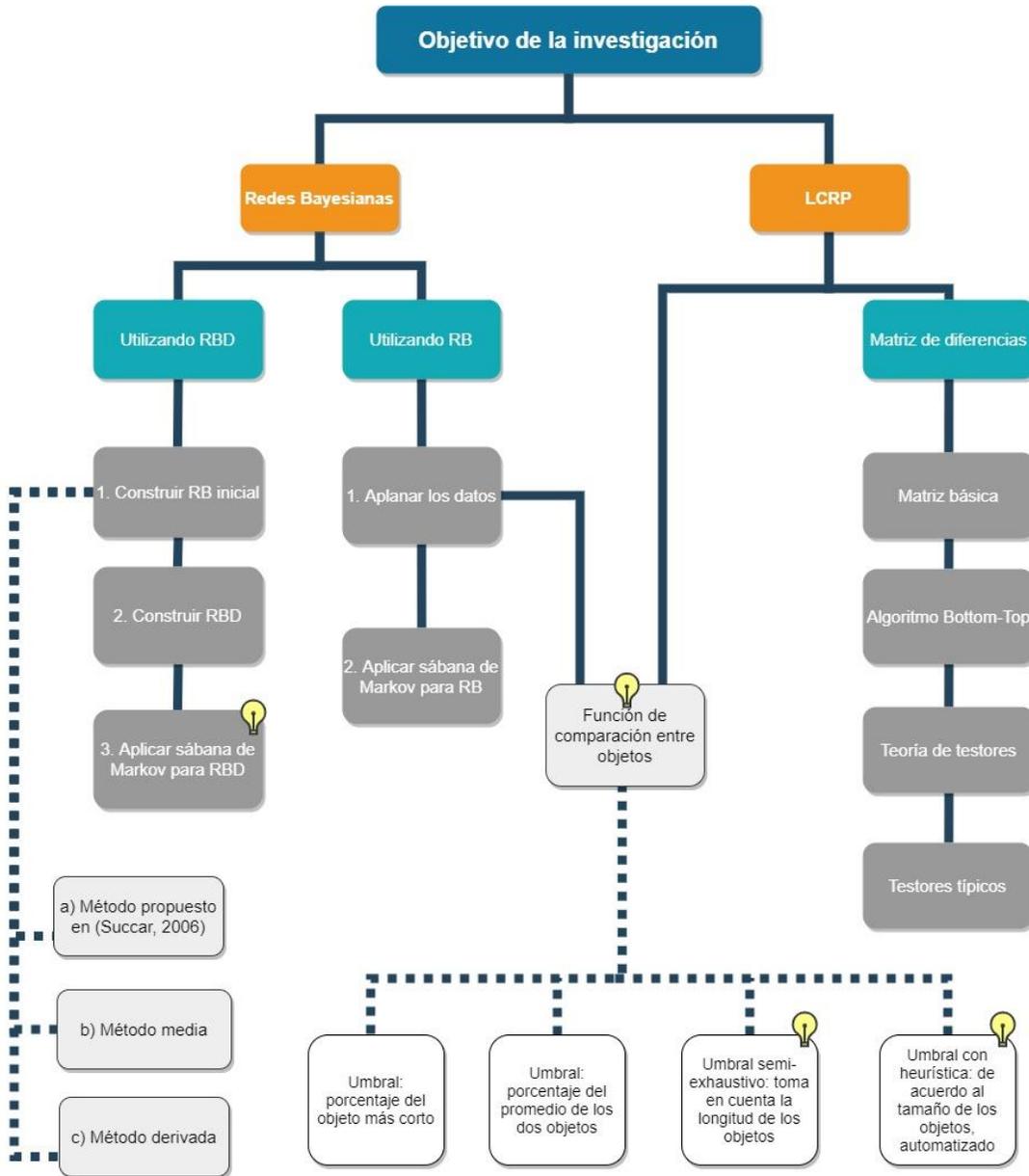
El algoritmo incluye las siguientes etapas:

1. Matriz de diferencias: es una matriz booleana que almacena la información del comparativo de las descripciones de objetos que pertenecen a la misma clase. Se almacena el valor 0 si dos descripciones son similares, en caso contrario se almacena el valor 1. Esta matriz de diferencias es el resultado de la función de comparación entre objetos.
2. La matriz básica sintetiza la información de la matriz de diferencias. La matriz básica elimina la información redundante de la matriz de diferencias. La matriz básica es binaria y se compone sólo por filas básicas de la matriz de diferencias (véase la sección *Definiciones* en el Anexo C).
3. Con el algoritmo *Bottom-Top* (BT) se calculan los *testores* y los *testores típicos* (véase la sección *Definiciones* en el Anexo c).

Dados los modelos y algoritmos, utilizados en los dos enfoques de solución, se da paso a la explicación de cómo estos se utilizaron para la selección de variables con datos crono-valorados. En la siguiente sección, 2.2, se presenta las adaptaciones, propuestas matemáticas y aportaciones hechas a dichos modelos y algoritmos; todo ello en su forma teórica y conceptual. Más adelante, en el capítulo tres se presenta el desarrollo e implementación de las ideas.

2.2 Modelo conceptual de la solución

La figura 2.3 muestra el modelo conceptual de la solución. En dicha figura se observan dos enfoques principales para solucionar el problema de investigación; así como los elementos del modelo conceptual. Además, en la misma figura, se especifican los aportes al área de investigación.



 Aporte a la investigación

Figura 2.3 Mapa Conceptual de la investigación

2.3 Descripción de los elementos del modelo conceptual

En esta sección se detallan los conceptos del mapa en la figura 2.3. Así, en esta sección se describen los dos enfoques que se siguen en la investigación, dando énfasis a las aportaciones de la investigación en su forma teórica.

2.3.1 Aportaciones de la investigación: Enfoque Uno - RB

Utilizando RBD

Con este enfoque se pueden utilizar los datos crono-valuados sin pre-procesar. Aquí, el desafío consiste en que la sábana de Markov para estas redes no tiene una definición establecida en la literatura. Por lo cual fue necesario introducir e implementar una formulación matemática para dicha sábana. Teniendo la formulación matemática se estableció la siguiente metodología:

1. Construir la red inicial. Se utilizaron tres métodos para la construcción de la red inicial:
 - a) Método propuesto en (Sucar, 2006). Este método utiliza la primera muestra del objeto para construir la red inicial.
 - b) Método Media, se utiliza la media de cada objeto para construir la red inicial.
 - c) Método derivada, se obtiene la derivada de cada objeto y se establece como parámetros para construir la red inicial.
2. Construir la red Bayesiana dinámica. Aquí se utiliza la información temporal para obtener las *interconexiones* (ver Anexo C para definición).
3. Aplicar la sábana de Markov sobre la red dinámica. En este paso de la metodología es donde se aplica la formulación matemática propuesta para la sábana de Markov a dichas redes.

Sabana de Markov a redes Bayesianas dinámicas, propuesta de formulación matemática

Para la sábana de Markov sobre redes Bayesianas dinámicas no se encontró literatura o implementación, en este trabajo se propone una formulación matemática para su implementación. La implementación está basada en las dos suposiciones de las redes Bayesianas:

- i) Si se instituye la memoria de corto plazo para obtener las tablas de probabilidad condicional (Friedman et al., 1998; Koller & Friedman, 2009), entonces utilizar sólo los tiempos ($t = 0$) y ($t = 1$) es una posibilidad viable para determinar los nodos padres, hijos y padres de los hijos de la variable clase.
- ii) Se establece la invariabilidad de la estructura de la red a través de los diferentes tiempos, entonces, utilizar los tiempos ($t = 0$) y ($t = 1$) otorga el mismo resultado que utilizar todos los tiempos para la obtención de la sábana de Markov.

Así, teniendo la red en dos tiempos ($t = 0$ y $t = 1$) se sigue la dirección de los arcos para establecer los hijos, los padres y los padres de los hijos de la variable de interés. Es decir, se tendrían los nodos de la sábana estática ($t = 0$) y los nodos que influyen en la variable de interés con la esencia cronológica (conexiones inter del tiempo $t = 1$).

Además de las suposiciones anteriores es necesario considera las siguientes ecuaciones:

- Axioma de probabilidad,

$$P(A, B) = P(A|B)P(B) \quad \text{Ec. 2.2}$$

- Regla de Marginalización de las redes Bayesianas,

$$P(X_{-i}) = \sum_{X_i} P(X_i, X_{-i}) \quad \text{Ec. 2.3}$$

donde X_{-i} representa el conjunto de variables excepto X_i .

- Definición de probabilidad condicional,

$$P(A|B) = \frac{P(A, B)}{P(B)} \quad \text{Ec. 2.4}$$

- Definición de los modelos gráficos probabilistas X_i es independiente de $X_{ancestros}$, dados los padres de X , es decir,

$$P(X_{1:n}) = P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P((X_i|Pa(X_i))) \quad \text{Ec. 2.5}$$

donde $pa(X_i)$ representa los padres de la variable X_i .

Demostración matemática

Dada la figura 2.4, se requiere verificar las variables involucradas para $P(X_i | \text{el resto de variables})$. Entonces, se tiene que demostrar:

$$P(X_i | X_{-i}, X_{-i}^t) = P(X_i | U, Y, Z, Pa(\text{hijo}^t(X_i))) \quad \text{Ec. 2.6}$$

donde $Pa(\text{hijo}^t(X_i))$ son las variables en el tiempo $t=0$ que son padres de un hijo de la variable de interés en el tiempo $t=1$. R, U, Y y Z forman el conjunto: *el resto de variables* (véase figura 2.4).

Dada la ecuación 2.4, se sustituye X_{-i}, X_{-i}^t

$$P(X_i | X_{-i}, X_{-i}^t) = \frac{P(X_i, X_{-i}, X_{-i}^t)}{P(X_{-i}, X_{-i}^t)} \quad \text{Ec. 2.7}$$

Dada la ecuación 2.3 y utilizando la red prueba en la figura 2.4:

$$P(X_i | X_{-i}, X_{-i}^t) = \frac{P(X_i, Y_i, Z_i, U_i, R_1, R_2, R_3, R_4, R_5, R_6)}{\sum_{X_i} P(X_i, Y_i, Z_i, U_i, R_1, R_2, R_3, R_4, R_5, R_6)} \quad \text{Ec. 2.8}$$

donde $R_1 \dots R_6$ son variables que no están incluidas en la sábana de Markov del tiempo $t = 0$.

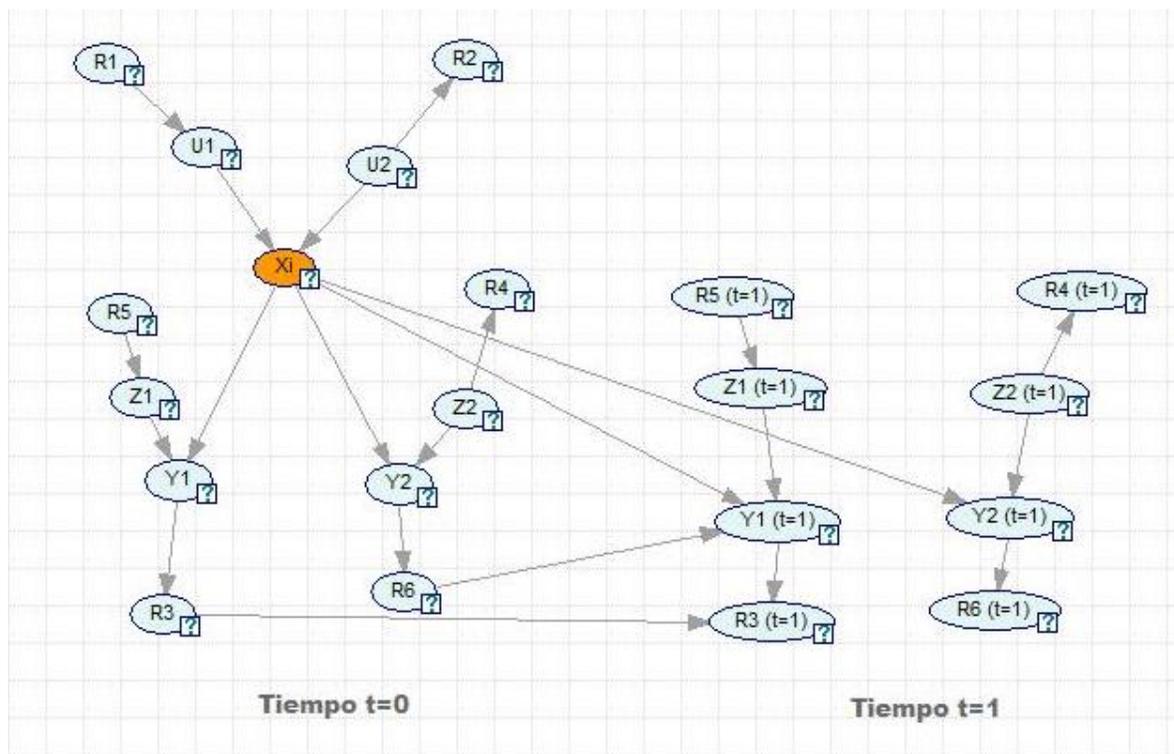


Figura 2.4 2-TBN que define una distribución de independencia para demostrar el concepto de sábana de Markov sobre redes Bayesianas dinámicas

- Definición de una red Bayesiana de dos tiempos (2-TBN): para un proceso sobre \mathcal{X} es una red Bayesiana condicional sobre \mathcal{X}' dado \mathcal{X}_t , donde $\mathcal{X}_t \subseteq \mathcal{X}$ es un conjunto de variables *interfaz*.

Las variables interfaz \mathcal{X}_t son aquellas variables cuyos valores en el tiempo t tienen efecto directo en las variables en el tiempo $t + 1$. Una 2-TBN representa una distribución condicional como:

$$P(\mathcal{X}'|\mathcal{X}) = P(\mathcal{X}'|\mathcal{X}_t) = \prod_{i=1}^n P(X'_i|Pa_{X'_i}) \quad \text{Ec. 2.9}$$

Utilizando la definición de los Modelos Gráficos Probabilistas (Ecuación 2.5), el DAG de la figura 2.4 y la ecuación 2.9, se tiene:

$$\begin{aligned} & P(X_i|X_{-i}, X_{-i}^t) \\ &= \frac{P(X_i|U)(\prod_j P(Y_j|X_i, Z)) P(R, U, Z) P(Y_1^t|X_i, Z_i, R_6) P(R_3|Y_1^t, R_3)}{\sum_{X_i} P(X_i|U)(\prod_j P(Y_j|X_i, Z)) P(R, U, Z) P(Y_1^t|X_i, Z_i, R_6) P(R_3|Y_1^t, R_3)} \end{aligned} \quad \text{Ec. 2.10}$$

Se elimina del denominador y numerador a $P(R, U, Z)$ y a $P(R_3|Y_1^t, R_3)$, quedando:

$$P(X_i|X_{-i}, X_{-i}^t) = \frac{P(X_i|U)(\prod_j P(Y_j|X_i, Z)) P(Y_1^t|X_i, Z_i, R_6)}{\sum_{X_i} P(X_i|U)(\prod_j P(Y_j|X_i, Z)) P(Y_1^t|X_i, Z_i, R_6)} \quad \text{Ec. 2.11}$$

Lo cual demuestra que la $P(X_i|X_{-i}, X_{-i}^t)$ no depende de otras variables excepto de $\{U, Y, Z, R_6\}$.

Con esta formulación matemática, la sábana de Markov para una RBD se compone de: la sábana del estado $t = 0$ más las variables del estado cero que son padres de un hijo de la variable de interés en el tiempo $t + 1$.

2.3.2 Aportaciones de la investigación: Enfoque Dos - LCRP

Utilizando LCRP y RB

En esta sección se describe el desarrollo de la metodología para la obtención de la *función de comparación entre objetos*, la cual es requerida para la solución de la investigación de los enfoques uno y dos.

La función de comparación entre objetos es utilizada en los dos enfoques. Cuando se aplica al enfoque uno, redes Bayesianas, se utiliza el número de clase como identificador de diferencia. Esto, debido a que las redes Bayesianas se basan en la noción de independencia condicional; la cual define que dado un conjunto de variables, si $P(A = a | X = x, B = b) = P(A = a | X = x)$ entonces B no da información extra acerca de A , haciendo a A y B condicionalmente independientes (Koller & Friedman, 2009). La función de comparación utiliza las diferencias para encontrar la información extra (intra clase), por lo cual se utiliza el identificador de la clase para diferencias. El resultado de la función de comparación entre objetos es una matriz de diferencias, la cual es utilizada en el enfoque uno para crear una red Bayesiana estática y posteriormente aplicar el concepto de sábana de Markov (utilizando la sábana ordinaria).

Para el enfoque dos, LCRP, la matriz de diferencias se utiliza para obtener la matriz básica para posteriormente generar los *testores* y *testores típicos* (siguiendo la metodología de (Martínez-Trinidad & Guzmán-Arenas, 2001). En el caso del enfoque dos, las diferencias son anotadas con el número uno y las similitudes con el número cero.

Antes de presentar la *función de comparación entre objetos*, se debe mencionar que dicha función da como resultado una matriz de diferencias, en la cual está explícito si dos objetos son diferentes o no, la comparación se realiza entre objetos de la misma clase y por cada variable presente en el *dataset*, el número de muestras en la matriz de diferencias está dado por,

$$DM = \sum_{i=1}^c (K_i) \times (K_i - 1) / 2 \quad \text{Ec. 2.12}$$

donde K_i es el número de objetos en la muestra para una clase y c el número de clases. El número de columnas es el mismo que el número de atributos en el *dataset*. La figura 2.5 ejemplifica esta idea de forma visual.

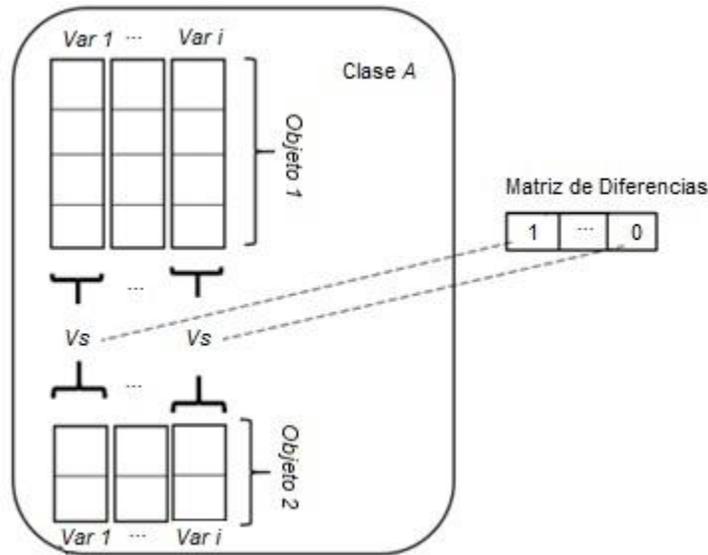


Figura 2.5 Descripción del resultado de la función de comparación

Una matriz de diferencias se puede describir visualmente con la figura 2.5, donde *Objeto 1* y *Objeto 2* pertenecen a la misma clase. Si los dos objetos se consideran similares, se coloca un cero en la matriz de diferencias, de lo contrario se escribe un identificador diferente de cero. En la misma figura se pueden apreciar dos objetos, ambos con diferente longitud con muestras vinculadas cronológicamente. La función de comparación entre objetos tiene como objetivo abordar este tipo de problemas: comparar dos objetos de diferente longitud (la mayor parte del tiempo), elemento por elemento en orden cronológico.

Función de comparación entre objetos, metodología

A continuación, se describen los pasos para la obtención de la matriz de diferencias.

1. Detección de sentido de los objetos. Este paso tiene como objetivo detectar el sentido de evolución de los objetos en cada variable para su posterior comparación. Dado que los objetos son secuencias de números reales, la evolución se expresa como,

$$EvO_i = \begin{cases} +1 & \text{if } O_{i_r} < O_{i_{r+1}} \\ 0 & \text{if } O_{i_r} = O_{i_{r+1}} \\ -1 & \text{otherwise} \end{cases} \quad \text{Ec. 2.13}$$

donde O_i representa un objeto en la muestra y r son las muestras vinculadas cronológicamente por objeto.

2. Normalización. Los valores de cada objeto se trasladaron para que el primero empiece en cero usando,

$$TO_{i_r} = O_{i_r} - O_{i_1} \quad \text{Ec. 2.14}$$

3. Escalado. El enfoque de escala adoptado fue,

$$SO_{i_r} = \frac{O_{i_r}}{\min O_i - \max O_i} \quad \text{Ec. 2.15}$$

4. Configuración de una ventana de comparación. Dado que los valores son números reales, se estableció un rango de comparación. Utilizando el valor de la pendiente del análisis de regresión de cada objeto,

$$\text{Range}O_{i_r} = O_{i_r} \pm \beta \quad \text{Ec. 2.16}$$

donde β describe la pendiente de la recta de regresión para O_i .

5. Sub-cadena común más larga (LCS). El algoritmo LCS (Gong & Chan, 2006) se utiliza para identificar cuántos elementos consecutivos de dos cadenas son similares (véase la sección *Definiciones* del Anexo C). Se adaptó el algoritmo para utilizar objetos que han sido procesados con los pasos dos y tres; luego, la comparación se diseñó para que se ejecute en el rango establecido en el paso cuatro junto con el sentido obtenido en el paso uno.

Para decidir si dos objetos son diferentes, se establece un umbral. El valor del umbral indica qué cantidad del objeto más corto debe considerarse como longitud suficiente para tomar la decisión. Un ejemplo se ilustra en la figura 2.6, donde dos objetos con diferente longitud corresponden a la misma clase, pero solo una parte de ambos se considera similar.

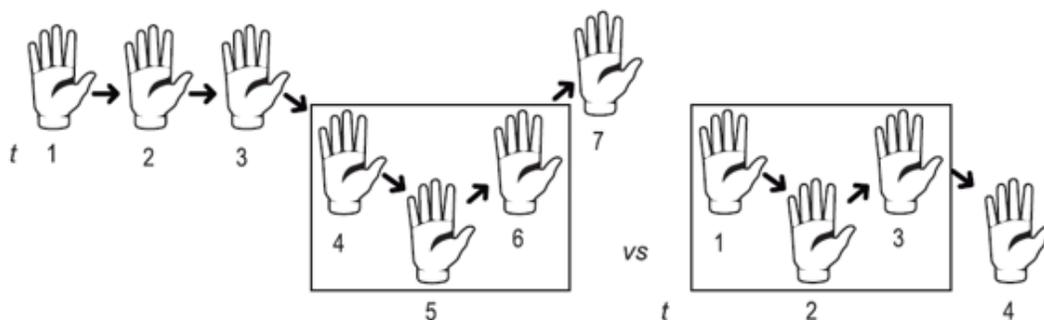


Figura 2.6 Dos objetos de diferente longitud pertenecientes a la misma clase

La decisión de si dos objetos son similares o no, se tomó en base al resultado del paso 5 y la longitud de los dos objetos, la investigación contempla cuatro posibilidades,

- I. Umbral: porcentaje del objeto más corto

- II. Umbral: porcentaje del promedio de los dos objetos
- III. Umbral empírico: toma en cuenta la longitud de los objetos, exhaustivo
- IV. Umbral con heurística: Heurística de acuerdo al tamaño de los objetos, automatizado

Establecer el umbral se convirtió en un problema en sí mismo ya que se pretende que la función se maneje automáticamente, sin necesidad de expertos.

El umbral de porcentaje del objeto más corto (primero) y el umbral de porcentaje del promedio de los dos objetos (segundo) son umbrales que se deben establecer, es decir algún experto debe establecer el porcentaje a tomarse en cuenta. De esta idea nace la necesidad del umbral con heurística y el umbral empírico.

Aquí se estudiaron cuatro formas diferentes de establecer el umbral de comparación. A continuación, se desarrolla la metodología del tercero y cuarto umbral de la lista anterior.

Umbral empírico de decisión

Este umbral se establece tomando en cuenta el tamaño de los objetos, se establece un porcentaje del objeto más corto en relación al más largo; sin embargo, la acción de establecer el porcentaje se hace de manera empírica. A continuación, los pasos del procedimiento:

Primero, se ordenan los objetos por tamaño,

$$shO = \min_r(O_i, O_j) \quad \text{Ec. 2.17}$$

$$lO = \max_r(O_i, O_j) \quad \text{Ec. 2.18}$$

donde O_i y O_j son dos objetos que se comparan y r es la longitud de cada uno.

Segundo, se establecen los porcentajes de acuerdo a los tamaños de objetos:

$$threshold = \begin{cases} 39\%shO & \text{if } shO \geq 14\%lO \\ 51\%shO & \text{otherwise} \end{cases} \quad \text{Ec. 2.19}$$

donde los porcentajes 39, 14 y 51 son ejemplo sin corresponder a ningún *dataset* en específico.

Finalmente, la decisión se toma de la forma,

$$d = \begin{cases} 0 & \text{if } result > threshold \\ C & \text{otherwise} \end{cases} \quad \text{Ec. 2.20}$$

donde *result* es el resultado del paso 5, *C* es el identificador para especificar que los dos objetos son diferentes.

Umbral con heurística de acuerdo al tamaño de los objetos, automatizado

Al contemplar la necesidad de una heurística para establecer el umbral de comparación de manera automática, se presentó la heurística que se describe en la figura 2.7.

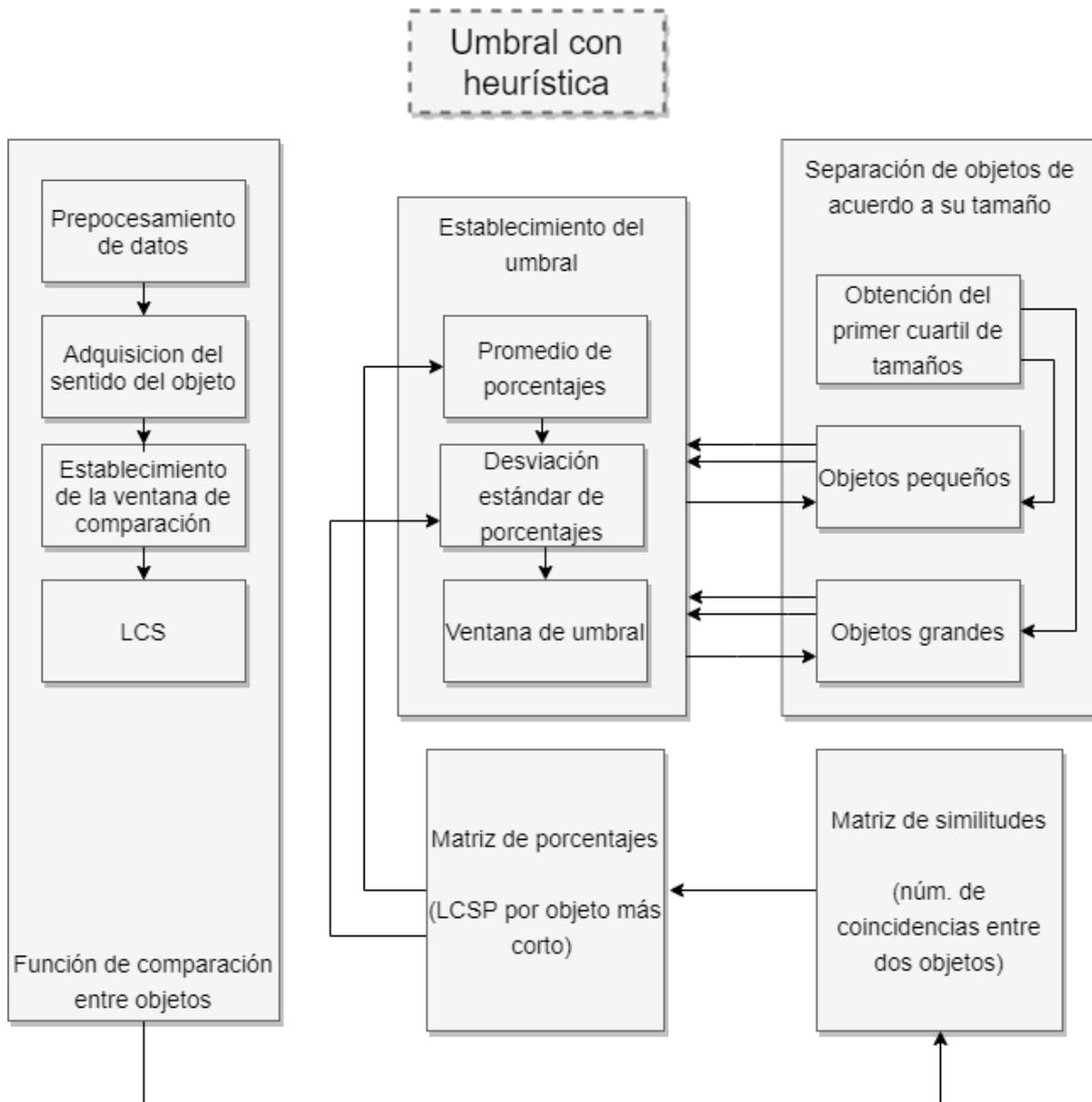


Figura 2.7 Heurística propuesta para el establecimiento del umbral de decisión

Con esta heurística se pretende acortar la búsqueda que se realiza con el umbral empírico además de no tener dependencia de expertos en el universo de datos.

2.4 Variables / criterios de control

Dado a que en la literatura revisada no hay trabajo con las mismas características que el aquí presentado, se propuso hacer una clasificación exhaustiva de los *datasets* utilizados. Esto tendrá sentido comparativo, dado que se busca que la clasificación hecha con los conjuntos de variables encontrados por los enfoques uno y dos, mejore los resultados de clasificación con todas las variables.

2.5 Discusión

Aquí se presentó los conceptos utilizados en la investigación dando como resultado el proceso teórico de la misma. Se puede ver que los desarrollos de la sección 2.3 son los que describen teóricamente las aportaciones a la investigación.

El modelo conceptual busca que la selección de variables sea independiente a expertos en el universo de los datos (crono-valorados). Con esto, se puede aplicar a cualquier conjunto de datos sin necesidad de adecuar la metodología planeada en esta sección.

CAPÍTULO 3

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se presentan los elementos de software desarrollados en esta investigación. Se inicia con el análisis de las herramientas utilizadas; después se continúa con el diseño del sistema (utilizando diagramas de flujo). Se finaliza con el desarrollo del sistema en forma de pseudocódigo.

3.1 Plataforma, lenguaje y conjunto de datos

3.1.1 R y R Studio

El desarrollo de la función de comparación, la selección de variables con redes Bayesianas, la clasificación de los atributos seleccionados y la heurística para el umbral de decisión son procedimientos que tienen fundamento estadístico y probabilístico, además de otras funciones matemáticas complejas. R es un lenguaje para computación estadística, además de tener librerías para presentar datos en diferentes gráficos de calidad (R Core Team, 2018). Se utilizó RStudio (RStudio Team, 2020) para esta investigación.

3.1.2 *Datasets*

La experimentación se realizó utilizando el conjunto de datos de gestos *Gesture Phase Segmentation* (Madeo et al., 2013) disponible en *UCI Machine Learning Repository* (Lichman, 2013). El conjunto de datos *Gesture Phase Segmentation* originalmente se obtuvo usando un sensor Microsoft Kinect® para obtener la posición de la mano izquierda, mano derecha, muñeca izquierda, muñeca derecha, cabeza y columna vertebral de tres personas en un sistema de coordenadas cartesianas para un espacio tridimensional. Cada participante tuvo que gesticular mientras contaba una historia cómica. Cada persona aparece en la lista con identificadores A, B o C; cada historia cómica se enumera con los números 1, 2 o 3 para diferenciar los archivos. Otras características se muestran en la Tabla 3.1.

Para este estudio, se utilizaron seis archivos:

- Entrenamiento (*Train*): A1, B1, C1
- Prueba (*Test*): A2, B3, C3

Tenga en cuenta que, el objetivo de clasificación para el presente estudio es por gestos, no por personas. De esta manera, se usó el archivo A1 para entrenar la máquina de aprendizaje y el archivo A2 para evaluar el modelo obtenido.

Tabla 3.1 Descripción detallada del conjunto de datos de *Gesture Phase Segmentation*

Información	Descripción
Tipo	Cronológicamente ligado
Tareas asociadas	Clasificación y agrupamiento
Número de muestras	A1 - 1747 frames, A2 - 1264 frames, A3 - 1834 frames B1 - 1073 frames, B3 - 1423 frames, C1 - 1111 frames, C3 - 1448 frames
Número de atributos y tipo de cada uno	18 atributos (double): <i>lhx, lhy, lhz, rhx, rhy, rhz, hx, hy, hz, sx, sy, sz, lwx, lwy, lwz, rwx, rwy, rwz</i> ; un atributo de marca de tiempo (integer) que se omitió para el presente estudio; y un atributo de clase (factor): <i>phase</i> .
Clases	<i>Rest, Preparation, Stroke, Hold, Retraction</i>
Donado por	Universidad de San Pablo, Brasil
Fecha de donación	18 junio 2014
Ubicación	https://archive.ics.uci.edu/ml/datasets/gesture+phase+segmentation

Otro aspecto importante para el presente estudio fue clasificar los conjuntos de datos de gestos tal como se presentan: en su formato sin procesar; esto incluye la clasificación con cinco clases. Sin embargo, como se han realizado estudios con el mismo conjunto de datos (Madeo et al., 2013; Wagner, Peres, Madeo, de Moraes Lima, & de Almeida Freitas, 2014) separando las clases en *Rest* y *NoRest*, también se decidió estudiar el conjunto de datos de la misma manera.

3.1.3 Implementación de la clasificación

Para la clasificación de las variables seleccionadas en cada enfoque (y todos los experimentos de clasificación), se implementaron Hidden Markov Model (HMM) en el entorno de RStudio con la librería *depmixS4* (Visser & Speekenbrink, 2010). Para mayores detalles de la implementación, vease Anexo A.

3.1.4 Interconexiones

Para obtener las dependencias temporales de los datos, se utilizó la librería *G1DBN* (Sophie Lèbre, 2009) para R. Las dependencias temporales en dicha librería son inferidas por medio del análisis de regresión a cada variable. El procedimiento cuenta con dos fases: en la primera se hace un análisis lineal y en la segunda se hace un análisis cuadrático y cúbico. En base a los resultados, se consideran las dependencias entre variables al asignar una calificación derivada de la prueba. Las calificaciones más pequeñas indican los arcos más significativos. Para mayores detalles de la implementación, refiérase al Anexo B.

3.2 Diagramas de flujo de los módulos del sistema y su correspondiente pseudocódigo

En las siguientes secciones se presentan los diagramas de flujo de la función de comparación entre objetos, de la *heurística para el umbral de decisión* y de la construcción de la RBD (con su correspondiente sábana).

3.2.1 Función de comparación entre objetos

Como se menciona en sección 2.3.2, la *función de comparación entre objetos* es utilizada en los dos enfoques que abarca este estudio. La figura 3.1, muestra el diagrama de flujo del análisis previo al diseño y construcción de dicha función.

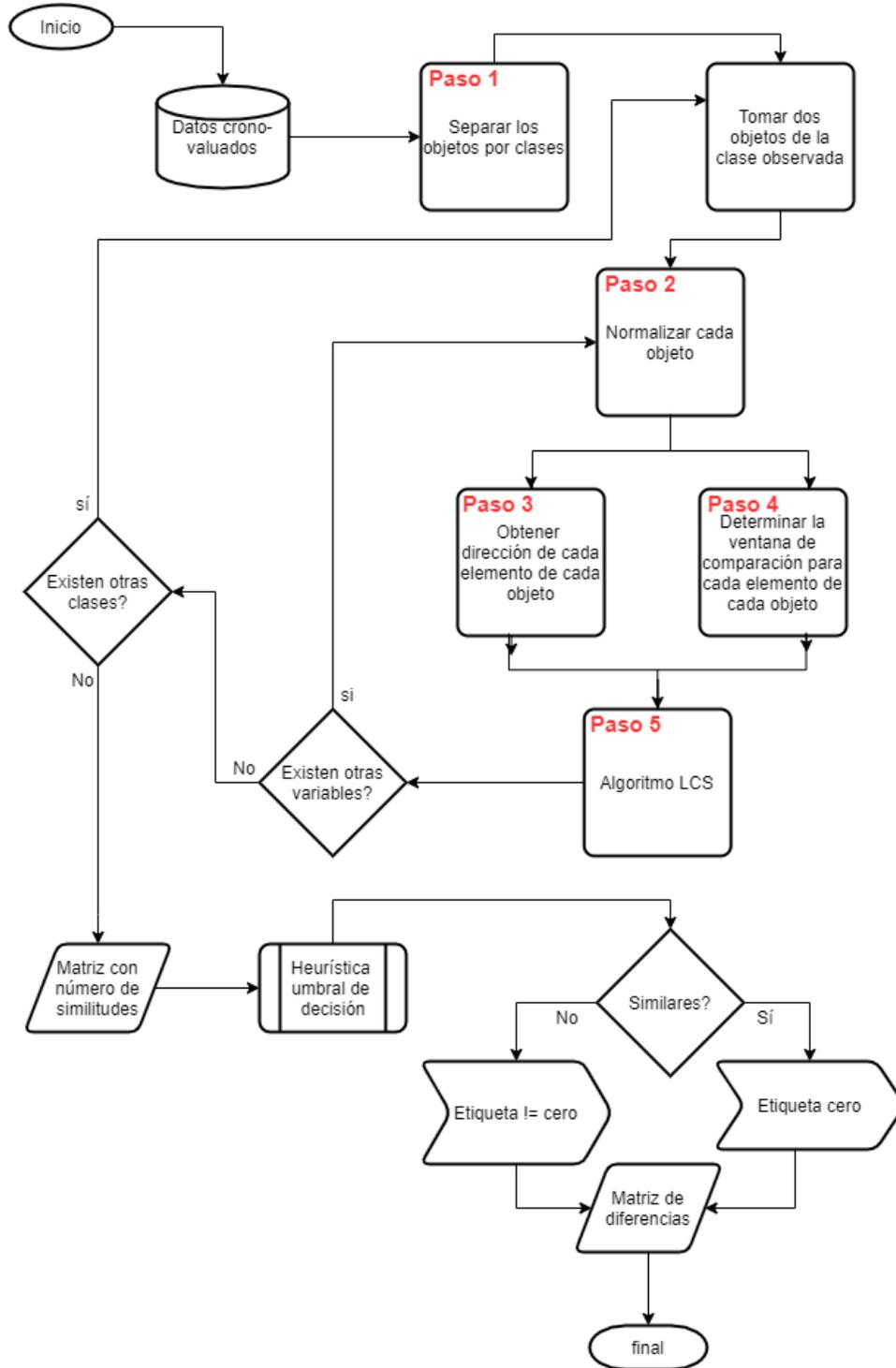


Figura 3.1 Diagrama de flujo de la Función de comparación entre objetos

En la figura 3.1 se muestra el diagrama de flujo con las generalidades de la función de comparación entre objetos. Dado que la función de comparación entre objetos es utilizada en los dos enfoques, el diagrama de flujo no contiene las fases de selección de variables y clasificación.

Pseudocódigo de la función de comparación entre objetos

Para mayor legibilidad de los pseudocódigos, estos se presentan por pasos (de acuerdo al número indicado en el diagrama de flujo correspondiente (figura 3.1).

Paso 1. Separar los objetos por clase

```
INPUT dataset con datos crono-valudos (variables y clases)
OUTPUT objetos por clase
DETERMINE primera clase del dataset
FOR cada una de las muestras del dataset
    IF muestra observada NOT EQUAL TO clase de la primera muestra THEN
        INCREMENT contador de objetos
        SET clase actual
    ENDIF
ENDFOR

#obtiene las posiciones por clase
DETERMINE las diferentes clases en el dataset
SET vectorInicios (contendrá posiciones de inicio de los objetos)
SET vectorFinales (contendrá posiciones terminales de cada objeto)
SET vectorClases (asociará cada posición de inicio y fin a una clase)
DETERMINE claseUno (primera clase del dataset)
SET agregaObjeto TO 1
SET vectorClases TO claseUno

FOR cada una de las muestras del dataset
    IF la muestra observada NOT EQUAL TO a claseUno THEN
        vectorFinales <- agregaObjeto en número de muestra
        observada menos 1
        agregaObjeto <- agregaObjeto + 1
        vectorInicios en agregaObjeto <- número de muestra
        observada
        SET claseUno TO clase en muestra observada
        SET vectorClases en agregaObjeto TO claseUno
    ENDIF
    IF número de muestra observada EQUAL TO longitud del dataset THEN
        vectorFinales en agregaObjeto <- número de muestra
        observada
    ENDIF
ENDFOR
todasPosiciones <- ( vectorInicios, vectorFinales, vectorClases)
```

Paso 2. Normalizar cada objeto

```
INPUT Objeto1 y Objeto2 de la clase observada
OUTPUT Objeto1 y Objeto2 normalizados
FOR cada uno de los objetos
    SET primerElemento de cada objeto (valor del primer elemento de cada
objeto)
ENDFOR
FOR cada uno de los objetos
    FOR uno hasta longitud de Objeto
        DETERMINE valor de cada elemento en el objeto menos
primerElemento
    ENDFOR
ENDFOR
FOR cada uno de los objetos
    SET max como el valor máximo en el objeto
    SET min como el valor mínimo en el objeto
    SET longitud como la longitud del objeto
    FOR uno a la longitud del objeto
        IF el elemento evaluado NOT EQUAL TO al primero THEN
            DETERMINE valor de cada elemento del objeto como el
valor del objeto entre max-min
        ENDIF
    ENDFOR
ENDFOR
```

Paso 3. Obtener dirección de cada elemento de cada objeto

```
INPUT Objeto1 y Objeto2 de la clase observada
OUTPUT dos objetos de la misma longitud que Objeto1 y Objeto2 con símbolos
1, -1, 0
SET Objeto1 como objSignos1 y Objeto2 como objSignos2
FOR cada objeto
    SET objSignos1 y objSignos2 en posición uno como cero (ya que la
primera posición se considera neutral)
    FOR 1 hasta la longitud del objeto
        IF la posición en la longitud NOT EQUAL TO última THEN
            IF valor del objeto en el elemento evaluado LESS THAN
el valor del siguiente elemento THEN
                objSignos en la dirección observada <- 1
            ENDIF
            IF el valor del elemento en el objeto el evaluado
GREATER THAN al valor del siguiente elemento THEN
                objSignos en la dirección observada <- -1
            ENDIF
            IF el valor del elemento en el objeto evaluado EQUAL TO
valor del siguiente elemento THEN
                objSignos en la dirección observada <- 0
            ENDIF
        ENDIF
    ENDFOR
ENDFOR
```

Paso 4. Determinar la ventana de comparación para cada elemento de cada objeto

```
INPUT Objeto1 y Objeto2 de la clase observada
OUTPUT ventana de comparación de Objeto 1 y Objeto 2
FOR cada par de objetos observados
    DETERMINE análisis de regresión lineal
    SET incremento TO pendiente
    window <- elemento +- incremento
ENDFOR
```

Paso 5. Algoritmo LCS con ediciones

```
INPUT Objeto1 y Objeto2 de la clase observada
OUTPUT número de la máxima coincidencia del algoritmo LCS
READ objeto1 y objeto2 (vectores)
DETERMINE longitud de objeto1 y objeto2
SET m como longitud de objeto1
SET n como longitud de objeto2
SET matriz para comparaciones. Número de columnas igual a n+1, número de
renglones igual a m+1
SET maxLongitud como contador de máxima longitud común TO cero
SET i TO 1
SET j TO 1
FOR i a m
    FOR j a n
        SET diferenciaObj TO objeto1 en posición i - objeto2 en
posición j
        IF diferenciaObj GREATER OR EQUAL TO window AND objSignos1
position i EQUAL TO objSingnos2posición j, THEN
            matriz posición i+1, j+1 <- matriz posición i,j mas 1
            IF matriz posición i+1,j+1 GREATER THAN maxLongitud
THEN
                maxLongitud <- matriz posición i+1,j+1
            ENDIF
        ENDIF
    ENDFOR
ENDFOR
RETURN maxLongitud
```

Las ediciones al algoritmo LCS permiten incluir la ventana de comparación de valores y el sentido de los signos obtenidos en el paso 3.

3.2.2 Umbrales de decisión para la función de comparación entre dos objetos

Esta sección contiene pseudocódigo de los diferentes Umbrales de decisión. Cada uno de los umbrales se establece para decidir si dos objetos son similares o no; en otras palabras, el algoritmo LCS da como resultado el número de coincidencias consecutivas de dos objetos, para decidir si estas coincidencias hacen o no a los objetos similares, se debe establecer un umbral.

Umbral: porcentaje del objeto más corto

```
INPUT Objeto1 y Objeto2 de la clase observada
OUTPUT umbral de decisión dados los objetos observados
FOR cada variable del dataset
  FOR cada par de objetos observados
    SET objetoMenor TO longitud del objeto de menor longitud
    SET porcentajeMenor como el porcentaje del objeto menor a
    tomar en cuenta
    umbral <- objetoMenor * porcentajeMenor
  ENDFOR
ENDFOR
```

El umbral de porcentaje del objeto más corto es relativamente sencillo, sin embargo, no toma en consideración la longitud del objeto más largo.

Umbral: porcentaje del promedio de los dos objetos

```
INPUT Objeto1 y Objeto2 de la clase observada
OUTPUT umbral de decisión dados los objetos observados
FOR cada variable del dataset
  FOR cada par de objetos observados
    SET objetoMenor TO longitud del objeto menor
    SET objetoMayor TO longitud del objeto mayor
    SET porcentajePromedio TO el porcentaje a tomar en cuenta
    umbral <- (objetoMenor + objetoMayor)/2 * porcentajePromedio
  ENDFOR
ENDFOR
```

Aunque el umbral del porcentaje del promedio de longitudes de los objetos toma en cuenta ambas longitudes, cuando los objetos son muy dispares en longitud, el promedio es una característica poco representativa.

Entonces se buscó la manera de tomar en cuenta la longitud de los dos objetos observados. Para esto se calculó el umbral empírico; este toma en cuenta la longitud de los dos objetos y la disparidad de longitud.

Umbral semi-exhaustivo: toma en cuenta la longitud de los objetos

```
INPUT Objeto1 y Objeto2 de la clase observada
OUTPUT umbral de decisión dados los objetos observados
FOR cada variable del dataset
  FOR cada par de objetos observados
    SET coincidencias como el número obtenido por el algoritmo LCS
    SET objetoMenor como el objeto de menor longitud
    SET objetoMayor como el objeto de mayor longitud
    SET porcentajeMayor como el porcentaje del objeto mayor a
    tomar en cuenta
    SET porcentajeMenor1 como el porcentaje del objeto menor a
    tomar en cuenta si los objetos son parecidos en longitud
    SET porcentajeMenor2 como el porcentaje del objeto menor a
    tomar en cuenta si los objetos son muy diferentes en longitud
    IF objetoMenor GREATER OR EQUAL TO a porcentajeMayor THEN
      SET UMBRAL TO porcentajeMenor1
    ELSE
      SET UMBRAL TO porcentajeMenor2
    ENDIF
  ENDFOR
ENDFOR
```

Con el umbral empírico se toma en cuenta la longitud de ambos objetos, así como la disparidad entre ellas; sin embargo, se debe hacer una búsqueda exhaustiva para encontrar la solución óptima.

Para dar solución a esta búsqueda exhaustiva se propone una heurística que determine el umbral de decisión de manera automática. Para ello, se tiene el diseño en la figura 3.2.

Para el umbral con heurística se toma en cuenta varios aspectos:

- Dispersión de las longitudes incluyendo *outliers*
- El porcentaje del objeto más chico de los dos objetos observados
- Variabilidad de los porcentajes según longitud de los objetos
- Ventana de porcentajes de acuerdo a la dispersión de los objetos

A continuación, se describe dicha heurística con un diagrama de flujo para después mostrar el pseudocódigo que se desarrolló.

3.2.3 Heurística para umbral de decisión

Como parte de la investigación se incluyó una heurística para el umbral de decisión. Esta heurística es la descrita en la sección 2.3.2. El diagrama de flujo de la heurística se presenta en la figura 3.2. Después se presenta el pseudocódigo de la misma.

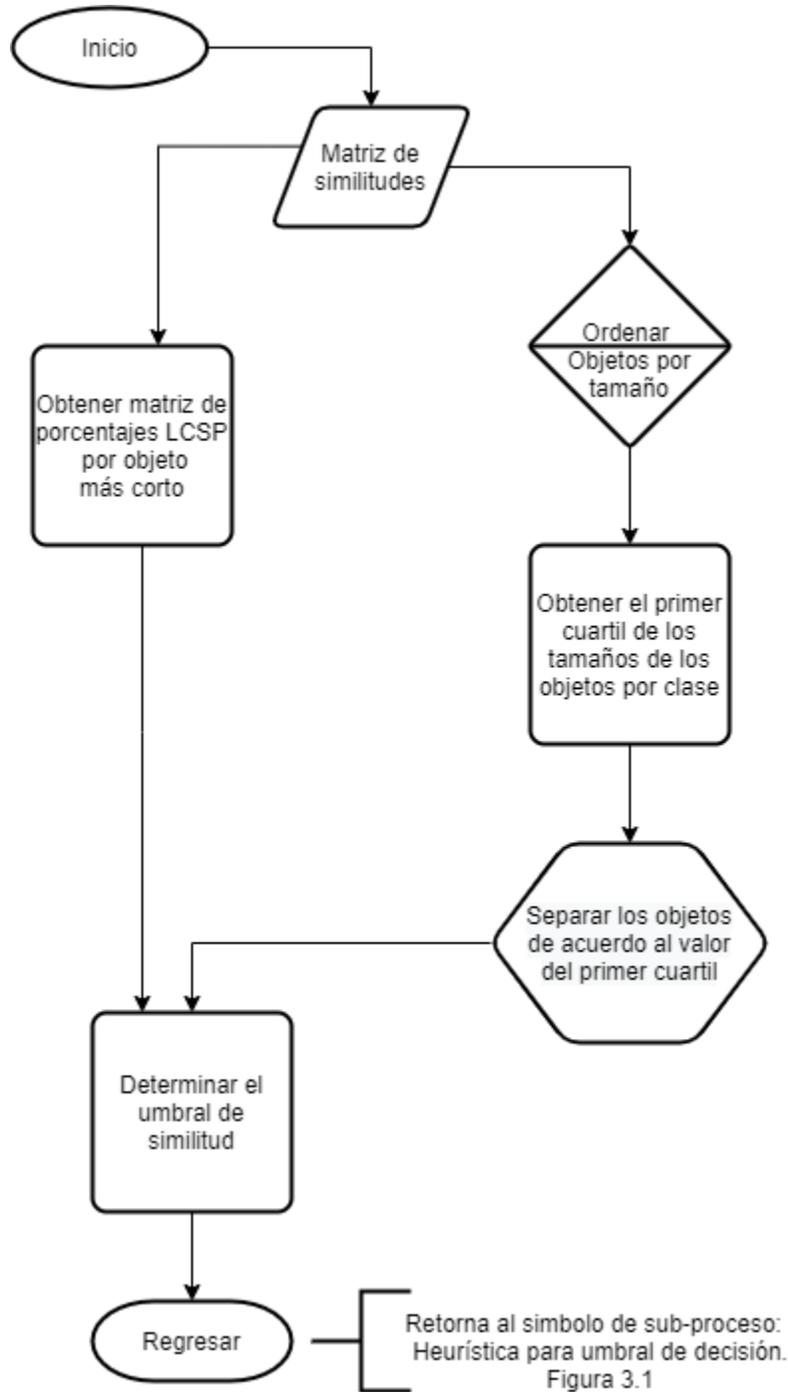


Figura 3.2 Diagrama de flujo de Heurística propuesta para el umbral de decisión

Pseudocódigo para la figura 3.2

```
INPUT matriz con número de similitudes de todos los objetos
OUTPUT umbral de decisión de acuerdo al tamaño de objetos en cada clase
READ tablaS con número de similitudes
DETERMINE número de clases presentes: niveles
FOR cada una de las muestras de tablas
    FOR cada nivel
        DETERMINE posición de último objeto de cada clase: ultimo
    ENDFOR
ENDFOR
SET listaLongitudes de los objetos de cada clase
SORT listaLongitudes de menor a mayor
DETERMINE primer cuartil de las longitudes
FOR cada una de las muestras en tablas
    FOR cada una de las variables en el dataset
        DETERMINE LCSP dado el objeto más corto
    ENDFOR
ENDFOR
FOR cada par de objetos observados en cada clase
    SET promedio de porcentaje LCSP
ENDFOR
FOR cada renglón en tabla de promedios
    FOR cada columna en tabla promedios
        DETERMINE valores para objetos del primer cuartil y el resto
    ENDFOR
ENDFOR
SET primerCuartil como primer cuartil de longitudes
SET umbralChicos como promedio de los valores para objetos en primer cuartil
más-menos desviación estándar
SET umbralGrandes como promedio de los valores para objetos fuera del primer
cuartil más-menos desviación estándar
```

Con esta heurística se cubren los requisitos de análisis y desarrollo de los módulos del sistema que utilizan el concepto de la función de comparación entre objetos. Este concepto se puede visualizar en la figura 2.1, en dicha figura se puede observar que los dos principales enfoques de la investigación hacen uso de la función. Por tal motivo, la importancia de desglosar el diseño e implementación de los módulos que la comprenden y sus diferentes variantes.

3.2.4 Diseño e implementación de la sábana de Markov para redes bayesianas dinámicas

En esta sección se muestra, por medio de diagramas de flujo y pseudocódigo, el diseño e implementación de la sábana de Markov para redes Bayesianas dinámicas. El diseño e implementación de dicha sábana está basado en la formulación matemática de la sección 2.3.1.

A continuación, con la figura 3.3, se muestra el diagrama de flujo del diseño de la sábana de Markov para redes dinámicas propuesta en este trabajo de investigación.

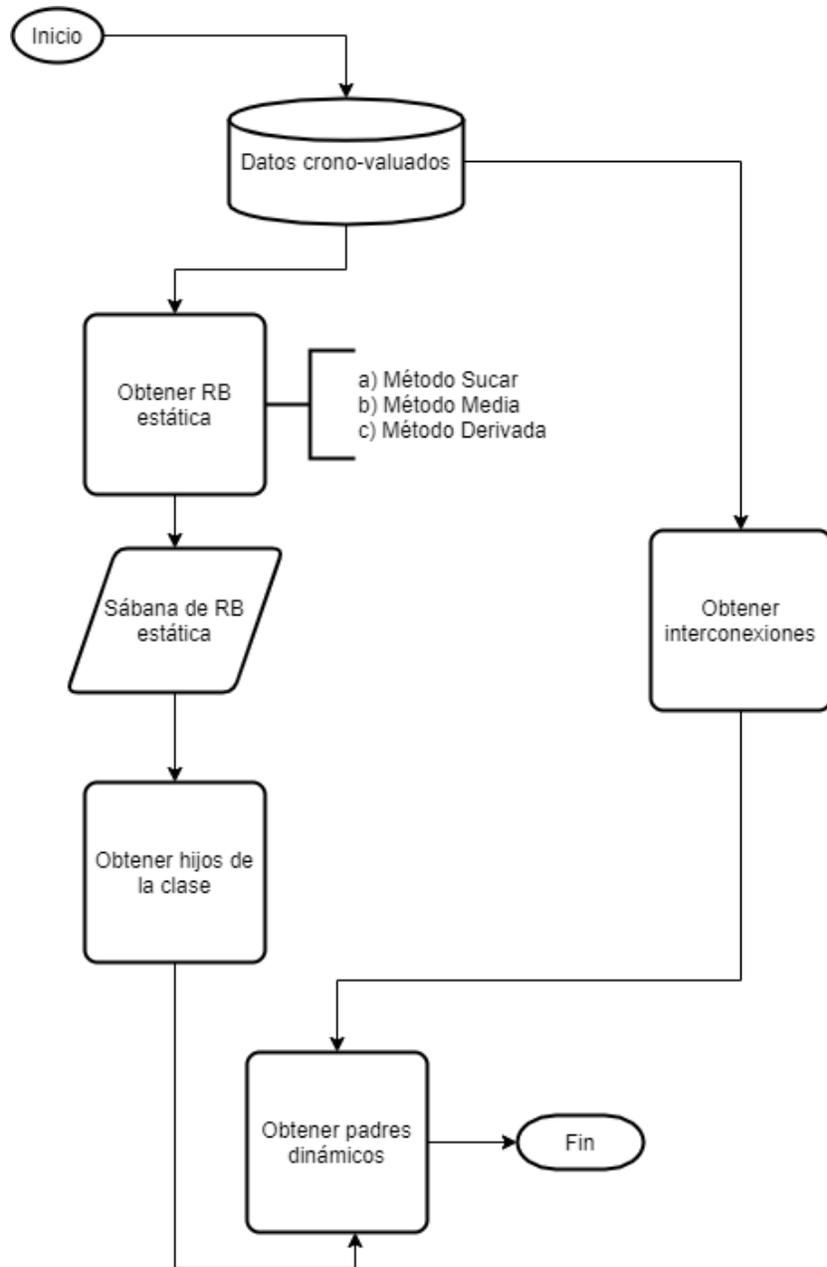


Figura 3.3 Diagrama de flujo para la obtención de la sábana de Markov dinámica

Pseudocódigo para la obtención de la sábana de Markov dinámica

```
INPUT datos crono-valorados
OUTPUT selección de variables a los datos de entrada
DETERMINE sábana de la red estática
DETERMINE interconexiones (variable y padre)
SET hijos TO hijos de la variable clase, en la sábana estática
SET contador TO 1
IF longitud de hijos NOT EQUAL TO cero
    FOR cada elemento en vector hijos
        FOR cada elemento en vector interconexiones
            IF interconexiones en padres EQUAL TO elemento de
            vector hijo THEN
                padresDinámicos <- padresDinámicos +
                interconexiones padre
            ENDIF
        ENDFOR
    ENDFOR
ENDIF
SET sabanaDinámica TO sábana AND padresDinámicos
RETURN sabanaDinámica
```

Con el anterior pseudocódigo se puede obtener la sábana dinámica, la cual es parte del enfoque uno. Véase la figura 2.1 si se requiere información acerca del mapa conceptual de la solución.

De igual forma en la figura 2.1, se puede observar que se utilizaron tres diferentes maneras de obtener la red Bayesiana estática (o inicial):

- a) Método propuesto en (Sucar,2006)
- b) Método *media*, ver anexo C para definición y fórmulas
- c) Método *derivada*, ver anexo C para definición y fórmulas

A continuación, se presenta el pseudocódigo de ellas.

Pseudocódigo para red Bayesiana estática: método propuesto en (Sucar, 2006)

```
INPUT datos crono-valorados
OUTPUT red Bayesiana estática de los datos de entrada
READ dataset con datos crono-valorados (variables y clases)
READ todasPosiciones (usar pseudocódigo de la función de comparación entre
objetos, paso 1)
SET datosRedEstatica TO dataset en primer renglón con todas las columnas
FOR cada renglón en todasPosiciones
    datosRedEstatica = dataset en la posición todasPosiciones en el
    renglón del ciclo de la primera columna, todas las columnas
ENDFOR
RETURN datosRedEstatica
DETERMINE sábanaEstatica utilizando datosRedEstatica
```

Pseudocódigo para red Bayesiana estática: método media

```
INPUT datos crono-valudos
OUTPUT red Bayesiana estática de los datos de entrada
DETERMINE númeroVariables, númeroObjetos, vectorInicioObjeto,
vectorFinObjeto, vectorClases
SET datasetMedia TO matriz con renglones = númeroObjetos y columnas =
númeroVariables +1
FOR i=uno a númeroVariables
  FOR c=uno a númeroObjetos
    SET rangoInicio TO vectorInicioObjeto en posición c
    SET rangoFin TO vectorFinObjeto en posición c
    datasetMedia posición c,i = mediana de dataset en
    rangoInicio,i hasta dataset en rangoFin,i
  ENDFOR
ENDFOR
datasetMedia en columna númeroVariables+1 = vectorClases
DETERMINE sábanaEstatica utilizando datasetMedia
```

Pseudocódigo para red Bayesiana estática: método derivada

```
INPUT datos crono-valudos
OUTPUT red Bayesiana estática de los datos de entrada
DETERMINE númeroVariables, númeroObjetos, posiciónInicial, posiciónFinal de
cada objeto
DETERMINE lista con posiciónInicial, posiciónFinal y clase de cada objeto
SET datasetDerivadas TO matriz con númeroObjetos y númeroVariables+1
SET inicios TO lista en posiciónInicial=1 y
SET finales TO lista en posiciónFinal=1
SET objetoEVA TO dataset en posición inicios hasta finales de la columna 1
FOR x=1 hasta númeroVariables
  FOR i=1 hasta númeroObjetos
    inicios <- lista posiciónInicial i
    finales <- lista posiciónFinal i
    objetoEVA <- datos posición inicios:finales,x
    SET maxIteracion TO longitud objetoEVA
    FOR j=1 hasta maxIteracion
      IF suma de diferencias de objetoEVA en nivel j EQUAL TO
      cero AND j EQUAL TO 1 THEN
        datasetDerivadas en i,x <- objetoEVA en 1
      ENDIF
      IF suma de diferencias de objetoEVA en nivel j EQUAL TO
      cero AND j NOT EQUAL TO 1 THEN
        SET g TO diferencias de objetoEVA en nivel j-1
        datasetDerivadas en i,x <- g en 1
      BREAK
      IF suma de diferencias de objetoEVA en nivel j
      NOT EQUAL TO cero AND j EQUAL TO maxIteracion THEN
        SET g TO diferencias de objetoEVA en nivel j
        datasetDerivadas en i,x <- g en 1
      ENDIF
    ENDFOR
  ENDFOR
ENDFOR
```

3.2.4 Diseño e implementación de la adaptación del algoritmo de LCRP

Después de obtener la matriz de diferencias (ver diagrama de la sección 3.2.1), se crea la matriz Básica a partir de la teoría de la sección 2.3.6. La matriz básica se introduce al algoritmo BT para obtener los testores típicos. A continuación, en la figura 3.4 se define el diagrama de flujo del procedimiento anterior; después se presenta el pseudocódigo de dicho procedimiento.

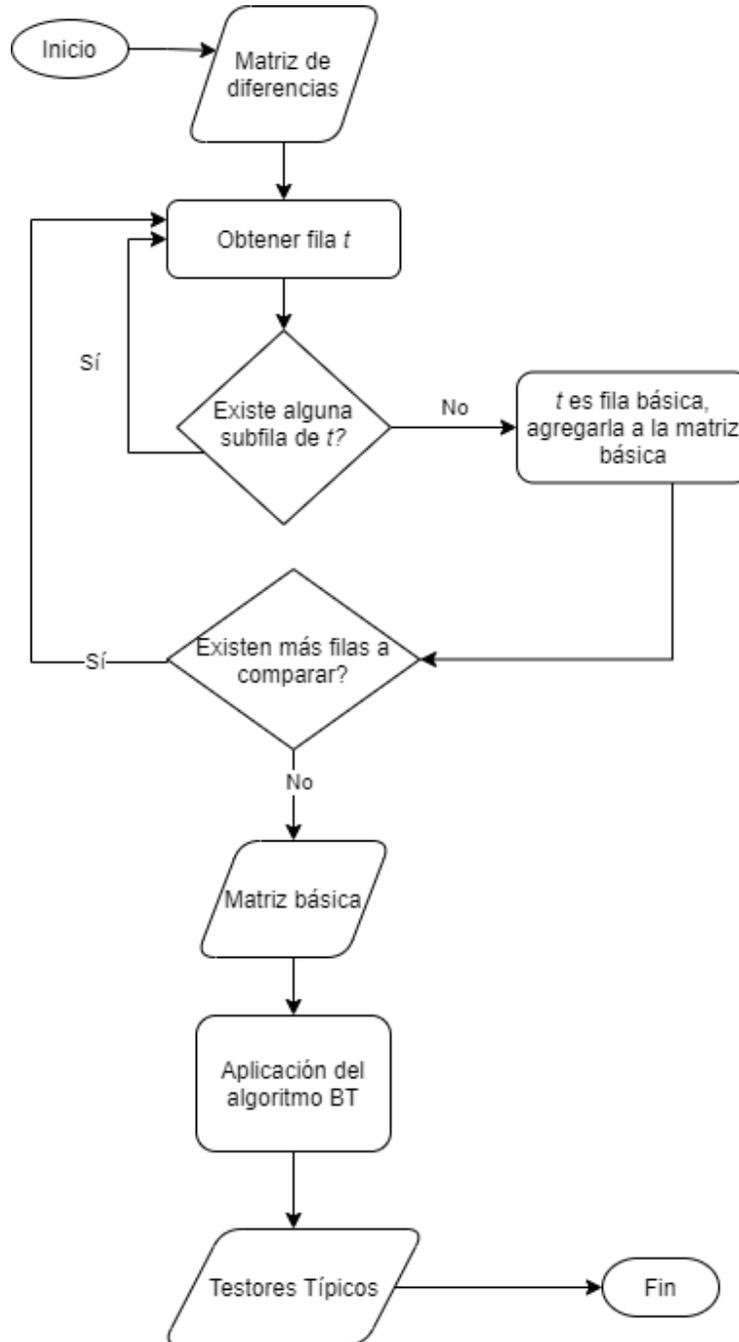


Figura 3.4 Diagrama de flujo de la adaptación al algoritmo LCRP

Pseudocódigo de implementación de la matriz básica y el algoritmo BT

```
INPUT Matriz de diferencias
OUTPUT testores típicos
FOR filas en la matriz de diferencias
  DETERMINE filas básicas utilizando:
     $\forall_j(a_{pj} = 1 \Rightarrow a_{tj} = 1)$  (en todas las columnas donde  $p$  tiene 1,  $t$  también)
     $\exists k(a_{tk} = 1 \wedge a_{pk} = 0)$  (existe al menos una columna en la que  $t$  tiene un 1 y  $p$  no lo tiene)
  ENDFOR
FOR filas en matriz básica (encontrar testores típicos (algoritmo BT))
  SET primera lista  $\alpha$  no nula, longitud = número de variables

  IF ( $\alpha$  es testor) THEN
    SET listas testores =  $2^{n-k} - 1$  tuplas siguientes ( $k$  es el subíndice del último 1 en  $\alpha$ )
  ELSE
    SET no_lista testores = listas desde  $\alpha$  hasta  $\alpha' = (\alpha'_1, \dots, \alpha'_n)$ 
  FOR ( $\alpha$  hasta  $\alpha' = (\alpha'_1, \dots, \alpha'_n)$ )
    IF ( $\alpha$  es testor) THEN
      SET listas testores =  $2^{n-k} - 1$  tuplas siguientes ( $k$  es el subíndice del último 1 en  $\alpha$ )
    ELSE
      SET no_lista testores = listas desde  $\alpha$  hasta  $\alpha' = (\alpha'_1, \dots, \alpha'_n)$ 
  ENDFOR
ENDFOR
RETURN SVD
```

CAPÍTULO 4

VALIDACIÓN Y

EXPERIMENTACIÓN

En los capítulos 2 y 3 se explicó el planteamiento y diseño de la solución para la selección de variables en datos crono-valorados. Sin embargo, es necesario hacer una serie de pruebas para comprobar que la metodología propuesta es funcional.

Para el proceso de experimentación se realizaron 12 pruebas. La cantidad de pruebas fue necesaria para comprobar los dos enfoques propuestos en sus diferentes prototipos. Cada prueba tiene un objetivo; a continuación, se presenta la perspectiva general de cada experimento y su objetivo:

- Experimento #1: tiene como objetivo validar la clasificación de los *datasets* utilizando todas las variables para todos los archivos en formato de dos y cinco clases
- Experimento #2: tiene como objetivo certificar que existe una clasificación mejor a la hecha en el experimento Uno, exhaustiva a los archivos A1 y A2 en formato dos y cinco clases
- Experimento #3: tiene como objetivo encontrar la mejor clasificación de los *datasets* utilizados en esta investigación con formato de cinco clases
- Experimento #4: tiene como objetivo la validación del método media para RBD
- Experimento #5: tiene como objetivo la validación de la propuesta de la sábana de Markov a RBD. Se utiliza el método derivada para la construcción de la red inicial
- Experimento #6: tiene como objetivo la validación de la propuesta de la sábana de Markov a RBD. Se utiliza el método propuesto en (Sucar, 2006) para la construcción de la red inicial
- Experimento #7: el objetivo es la introducción de la función de comparación entre objetos. Se utilizan las RB para modelar los datos
- Experimento #8: el objetivo es conocer el desempeño del enfoque con la LCPR utilizando la función de comparación entre objetos
- Experimento #9: el objetivo es mostrar cómo se establecieron los porcentajes de umbral de acuerdo a la longitud de los objetos
- Experimento #10: tiene como objetivo comprobar el desempeño del umbral semi-exhaustivo. Se utiliza la *función de comparación entre objetos* y las RB
- Experimento #11: tiene como objetivo comprobar el desempeño del umbral semi-exhaustivo. Se utiliza la *función de comparación entre objetos* y la LCRP
- Experimento #12: su objetivo es la validación de la heurística para el umbral de decisión para la función de comparación entre objetos. Se utiliza el enfoque dos con RB.

4.1 Experimento #1

Este experimento tiene como objetivo validar la clasificación de los *datasets* utilizando todas las variables. El experimento se diseñó de manera que se utilizarán las 18 variables con formato de dos y cinco clases en todos los archivos. Es decir, se entrenó la HMM con los parámetros mostrados en tabla 4.1, donde dos clases se refiere a *Rest* y *No Rest* (tal como se explica en la sección 3.1.2).

Tabla 4.1 parámetros utilizados en el experimento Uno

Archivo entrenamiento	Archivo Prueba	Formato
A1	A2	Dos clases
A1	A2	Cinco clases
B1	B3	Dos clases
B1	B3	Cinco clases
C1	C3	Dos clases
C1	C3	Cinco clases

El experimento pretende mostrar la clasificación con todas las variables para posterior comparación con el resto de los experimentos. La justificación recae en que la selección de variables mejora la clasificación dada con todas las variables. Para el experimento se utilizó:

- La HMM descrita en la sección 3.1.3
- Los *datasets* *A1*, *A2*, *B1*, *B3*, *C1* y *C3* descritos en la sección 3.1.2.

Los resultados obtenidos se muestran en la tabla 4.2, donde se puede observar las medidas *Accuracy*, *Precision* y *Recall* (refiérase al glosario para definición).

En la tabla 4.2 se puede apreciar que, utilizando todas las variables con formatos de dos y cinco clases, la HMM no puede distinguir la mayoría de las clases y se tienen medidas de *accuracy* muy bajas.

Como ya se mencionó, los resultados del experimento Uno (tabla 4.2) serán utilizados como comparativo para el resto de los experimentos. Se puede apreciar, con los resultados, que la medida *Accuracy* es muy bajo para la clasificación con todas las variables, siendo esto la base de partida para la selección de variables.

Tabla 4.2 Resultados de la clasificación utilizando todas las variables

Archivos	Clase	Accuracy	Precision	Recall
A1, A2	Rest	0.1313	0.1156	0.1845
	No Rest		0.1572	0.0972
A1, A2	Hold	0.0585	0.0	0.0000
	Preparation		NI	0.0000
	Rest		NI	0.0000
	Retraction		0.0806	0.8604
	Stroke		NI	0.0000
B1, B3	Rest	0.1362	0.1362	1.0000
	No Rest		NI	0.0000
B1, B3	Hold	0.1411	NI	0.0000
	Preparation		NI	0.0000
	Rest		0.1411	1.0000
	Retraction		NI	0.0000
	Stroke		NI	0.0000
C1,C3	Rest	0.2500	0.2500	1.0000
	No Rest		NI	0.0000
C1,C3	Hold	0.0994	0.0994	1.0000
	Preparation		NI	0.0000
	Rest		NI	0.0000
	Retraction		NI	0.0000
	Stroke		NI	0.0000

¹NI significa que no se identificó la clase correspondiente.

4.2 Experimento #2

Este experimento tiene como objetivo certificar que existe una clasificación mejor a la hecha en el experimento Uno. El experimento se diseñó de manera que se ejecute una clasificación exhaustiva (el conjunto potencia menos uno, el conjunto vacío no es relevante), para esto se dividió en número de variables posibles en las combinaciones. La tabla 4.3 muestra el número de combinaciones por número de elementos, que es la forma en se realizó la clasificación exhaustiva.

Tabla 4.2 Número de posibles combinaciones por número de variables en el *dataset*.

Número de variables en combinatoria	Número de combinaciones posibles
1	18
2	153
3	816
4	3060
5	8568
6	18564
7	31824
8	43758
9	48620
10	43758
11	31824
12	18564
13	8568
14	3060
15	816
16	153
17	18
18	1

La justificación del experimento nace en que la literatura consultada es nula en cuestión de experimentación con este *dataset* en selección de variables. Existen experimentos con este *dataset* que manejan extracción de variables. Por lo anterior se necesita saber cuál es el mejor resultado posible y así tener un punto de comparación que tenga los mismos parámetros (máquina de aprendizaje y selección de variables).

Para el experimento se utilizó:

- La HMM descrita en la sección 3.1.3
- Los *dataset A1* y *A2* para dos y cinco clases descritos en la sección 3.1.2.

Los resultados obtenidos se muestran en la tabla 4.4, donde se puede observar que para algunos resultados existen diversas combinaciones de variables.

Tabla 4.3 Clasificación exhaustiva con dos clases.

Número de variables	Accuracy	Precision		Recall		Subconjunto con el mayor <i>accuracy</i> en la clasificación
		Rest	NoRest	Rest	NoRest	
6	0.9051	0.8	0.9	0.8	0.9	rhy,rhz,hz,sy,sz,lwz
5	0.9027	0.8	0.9	0.9	0.8	lhz,rhy,rhz,sy,lwz,
9	0.8987	0.8	0.9	0.9	0.8	lhy,lhz,rhy,rhz,hz,sy,lwz,rwy,rwz
7	0.8964	0.8	0.9	0.8	0.8	lhy,lhz,rhy,rhz,sy,sz,lwz
4	0.8956	0.8	0.9	0.8	0.8	rhy,rhz,sy,lwz
10	0.8908	0.8	0.9	0.9	0.8	lhy,lhz,rhy,rhz,hx,hz,sy,lwz,rwy,rwz
8	0.8869	0.8	0.9	0.9	0.8	lhy,lhz,rhz,sy,sz,lwz,rwy,rwz
3	0.8782	0.8	0.9	0.9	0.8	rhz,sz,lwy
2	0.8774	0.7	0.9	0.9	0.8	rhz,lwy
11	0.8631	0.7	0.9	0.9	0.8	lhy,lhz,rhy,rhz,hx,hz,sz, lwy,lwz,rwy,rwz
12	0.8410	0.8	0.8	0.7	0.8	lhz,rhx,rhy,rhz,hx,hy, hz,sx,sy,sz,lwx,lwy,lwz,rwz
14	0.8394	0.7	0.8	0.8	0.8	lhx,lhy,lhz,rhy,rhz,hx,hy,hz,sx, sy,sz, lwz,rwy,rwz
13	0.8386	0.8	0.8	0.7	0.9	lhx,lhy,lhz,rhy,rhz,hx,hy,hz,sx, sy, lwz,rwy,rwz
1	0.7334	0.6	0.8	0.8	0.6	lhz
16	0.3291	0.3	0.1	0.8	0.0	lhx,lhy,lhz,rhx,rhy,hx,hy,hz, sy,sz,lwx,lwy,lwz,rwx,rwy,rwz
17	0.3078	0.3	0.1	0.7	0.0	lhx,lhy,lhz,rhx,rhy,rhz,hx,hy,hz, sy,sz,lwx,lwy,lwz,rwx,rwy,rwz
18	0.1313	0.1	0.1	0.1	0.0	lhx,lhy,lhz,rhx,rhy,rhz,hx,hy,hz,sx, sy,sz,lwx,lwy,lwz,rwx,rwy,rwz

¹Las medidas *Precision* y *Recall* se expresan con un dígito después del punto debido a espacio en la tabla.

Los subconjuntos de clasificación se dividieron por el número de atributos (elementos) para simplificar la presentación y la capacidad de gestión computacional.

La tabla 4.5 resume los resultados encontrados en la clasificación exhaustiva de cinco clases, es decir, *Hold*, *Preparation*, *Rest*, *Retraction* y *Stroke* usando el archivo A1 para entrenar y A2 para probar.

Con este par de clasificaciones exhaustivas se pretende hacer comparaciones para los subconjuntos que se extraigan de la selección de variables con cualquiera de los dos enfoques.

Tabla 4.4 Clasificación exhaustiva para cinco clases.

Núm varia -bles	Accu- racy	Precision					Recall					Subconjunto con el mayor <i>accuracy</i>
		1	2	3	4	5	1	2	3	4	5	
9	0.5309	0.0	0.3	0.6	1.0	NI	0.0	0.8	0.9	0.0	0.0	lhy,rhy,hx,hy,sx, sz,lwx,rwy,rwz
7	0.4889	0.0	0.3	0.5	NI	NI	0.0	0.6	0.9	0.0	0.0	lhy,lhz,rhy,hx,sx, sy,rwz
8	0.4865	0.0	0.3	0.5	0.0	NI	0.0	0.6	0.9	0.0	0.0	lhz,rhy,hx,hy,sx, lwy,lwz,rwz
6	0.4533	NI	0.2	0.6	NI	NI	0.0	0.9	0.7	0.0	0.0	lhy,hx,sx,sy,rwx, rwz
5	0.4493	0.0	0.2	0.6	NI	NI	0.0	0.4	0.9	0.0	0.0	rhy,rhz,hy,sy,lwz
10	0.4438	NI	0.2	0.7	0.1	NI	0.0	0.2	0.9	0.5	0.0	rhy,rhz,hx,hy,hz, sx,sz,lwz,rwx,rwy
11	0.4327	NI	0.1	0.8	NI	NI	0.0	0.6	0.8	0.0	0.0	hy,lhz,rhy,hx,hy, sx,sy,sz,lwz,rwy, rwz
3	0.4303	NI	0.1	0.8	0.1	0.3	0.0	0.6	0.8	0.0	0.0	lhz,sy,rwy
4	0.4303	0.0	NI	0.4	NI	0.8	0.0	0.0	0.9	0.0	0.1	lhx,lhy,sx,lwx
2	0.4233	NI	0.1	0.6	NI	NI	0.0	0.2	0.9	0.0	0.0	sx,rwy
1	0.3900	NI	NI	0.3	NI	NI	0.0	0.0	1.0	0.0	0.0	sx
12	0.3671	0.0	0.1	0.9	0.1	NI	0.0	0.7	0.6	0.3	0.0	lhy,lhz,rhx,rhy,hx, hy,hz,sx,sz, lwz,rwy,rwz
13	0.3101	NI	NI	0.3	0.1	0.0	0.0	0.0	0.7	0.2	0.0	lhx,lhz,rhx,rhy,hy, hz,sy,sz,lwx,lwy, rwx,rwy,rwz
14	0.3054	NI	NI	0.5	0.0	0.0	0.0	0.0	0.7	0.3	0.0	lhy,rhx,rhy,rhz,hx ,hy,hz,sy,sz,lwy, lwz,rwx,rwy,rwz
16	0.1091	0.0	0.1	NI	0.1	NI	0.0	0.4	0.0	0.5	0.0	lhx,lhy,lhz,rhx,rhz ,hx,hy,hz,sx, sy, lwx,lwy,lwz,rwx, rwy,rwz
17	0.0783	0.0	0.1	NI	0.0	NI	0.0	0.2	0.0	0.6	0.0	lhx,lhy,lhz,rhy,rhz ,hx,hy,hz,sx, sy,sz,lwx,lwy,lwz, rwx,rwy,rwz
18	0.0585	0.0	NI	NI	0.1	NI	0.0	0.0	0.0	0.8	0.0	lhx,lhy,lhz,rhx,rhy ,rhz,hx,hy,hz,sx, sy,sz,lwx,lwy,lwz, rwx,rwy, rwz

¹NI significa que no se identificó la clase correspondiente.

²Las medidas *Precision* y *Recall* se expresan con un dígito después del punto debido a espacio en la tabla.

Como conclusión se puede observar que la clasificación más baja en los dos experimentos es la que corresponde a la clasificación con todas las variables del dataset. Por lo tanto, cualquier selección que se pueda hacer, será para alcanzar una mejor clasificación.

También se concluye que existe sólo una combinación que logra el mejor *Accuracy* para dos y cinco clases, con esto se descubre cuáles son las variables que mejor definen al universo de datos.

4.3 Experimento #3

Este experimento tiene como objetivo encontrar la mejor clasificación a los *datasets* mencionados en la sección 3.2.1. El experimento se diseñó con las características descritas en la tabla 4.3 del experimento 2. Así, la justificación es la misma del experimento 2. Sin embargo, en el presente experimento se pone énfasis al formato de 5 clases ya que este formato alcanzó un *Accuracy* muy bajo en el experimento 2, por lo que se pretende explorar exhaustivamente todos los *datasets* para comprobar que en todos ellos el comportamiento es similar.

Para el experimento se utilizó:

- La HMM descrita en la sección 3.1.3
- Los *datasets* A1, A2, B1, B3, C1 y C3 descritos en la sección 3.1.2
- Training A1, B1 y C1. Prueba con A2, B3 y C3.

Los resultados se presentan en la tabla 4.6.

Tabla 4.5 máximo *Accuracy* posible de todos los *datasets* en formato de cinco clases

Archivos		<i>Accuracy</i> máximo
Test	Train	
A1	A2	0.5308
B1	B3	0.4058
C1	C3	0.5303

Se puede apreciar el desbalance en los *datasets* en formato de cinco clases. A pesar de esto, se considera que la información encontrada será para comprobar que tan cerca se puede llegar en la clasificación con la selección de variables. Con esto se está estableciendo que, el propósito de los experimentos posteriores se diseñó para probar la veracidad de las metodologías aquí presentadas, no para aunar en los *datasets* y sus respectivos inconvenientes.

4.4 Experimento #4

Este experimento tiene como objetivo mostrar que la selección de variables, al *dataset* de la sección 3.1.2, produce mejores medidas de clasificación que sin selección de variables. El experimento se diseñó con el enfoque de redes Bayesianas dinámicas y el método Media (ver figura 2.1). El experimento pretende validar que:

- El método Media puede ser una variante plausible al método propuesto por Sucar (Sucar, 2006).

Para el experimento se utilizó:

- Pseudocódigo para red bayesiana estática: método Media (sección 3.2.3)
- Algoritmos de construcción de redes Bayesianas usando la librería *bnlearn* para R (Scutari, 2010); PC, *Grow-Shrink* (GS), *Incremental Association Markov Blanket* (IAMB), *Fast Incremental Association* (Fast-IAMB), *Interleaved Incremental Association* (Inter-IAMB), *Incremental Association with FDR Correction* (IAMB-FDR), *Tabu Search* (Tabu), Hill Climbing (HC), *Max-Min Hill Climbing* (MMHC), *General 2-Phase Restricted Maximization* (RSMAX2), *Hybrid HPC* (H2PC), *Max-Min Parents & Children* (MMPC) y *Semi-Interleaved Hiton-PC* (SI-HITON-PC)
- Sábana de Markov utilizando la librería *bnlearn* para R
- Máquina HMM para la clasificación de las variables seleccionadas descrita en la sección 3.1.3.
- Archivos A1 y A2 con modalidad de cinco clases.

Los resultados se presentan en la tabla 4.7.

Tabla 4.6 Resultados de la sábana de Markov a red construida con el método media

Algoritmo de construcción de RB	Accuracy Red inicial o estática
PC	0.3852
GS	NS
IAMB	0.0625
Fast-IAMB	0.0625
Inter-IAMB	0.0625
IAMB-FDR	0.0625
HC	0.0949
Tabu	0.0949
MMHC	NS
RSMAX2	0.1693
H2PC	NS
MMPC	0.1392
SI-HILTON-PC	0.1693

En la tabla 4.7 el distintivo *NS* expresa que no se encontró sábana con el algoritmo especificado.

Al comparar las tablas 4.7 (del presente experimento) y 4.5 (experimento 2) se puede observar que:

- La selección de variables, hecha con el presente experimento, supera la medida *accuracy* obtenida con 18 variables (renglón 18 de la tabla 4.5).
- Dado el *accuracy* máximo para los archivos *A1*, *A2* en modalidad de cinco clases (tabla 4.5) no se justifica el uso de medidas adicionales al *accuracy* para este experimento.

Como puede verse en la tabla 4.7, las redes construidas y sus respectivas sábanas consiguen resultados de clasificación dispersos. La desviación estándar de las clasificaciones es igual a 130 y existen *outliers*, lo cual, en conjunto, ejemplifican la dispersión de resultados. Véase la figura 4.1 para el resumen de los resultados.

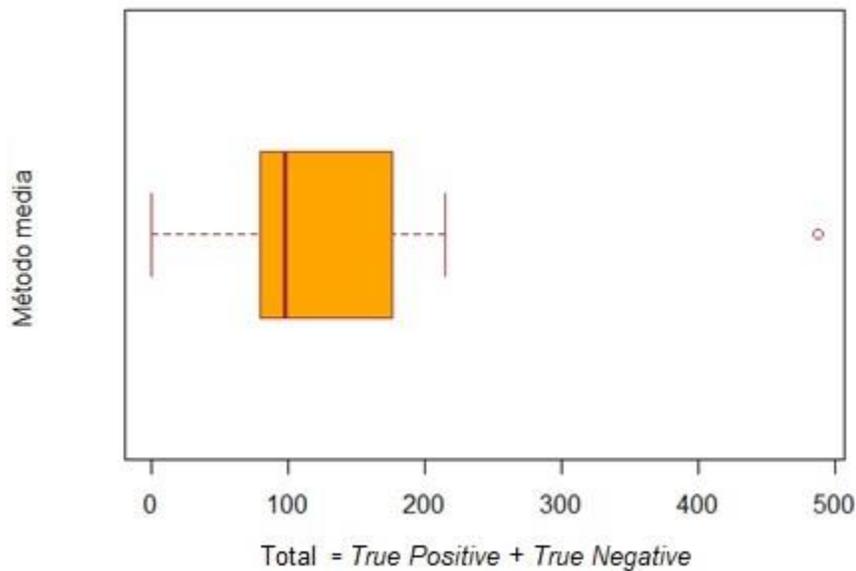


Figura 4.1 Resumen de los resultados de la Tabla 4.7

Con las validaciones anteriores se puede concluir que el método *Media* para la creación de las redes Bayesianas y la selección de variables con la sábana de Markov ordinaria, obtienen resultados dispersos no óptimos. Además, de que la información evolutiva del dato no está siendo tomada en cuenta y las redes Bayesianas obtenidas no pueden ser ajustadas por ningún parámetro dada la naturaleza su construcción.

4.5 Experimento #5

Este experimento tiene como objetivo mostrar que, utilizando la selección de variables se obtienen mejores resultados en lugar de la clasificación obtenida utilizando todas las variables. El experimento se diseñó con el enfoque de redes Bayesianas método Derivada (ver figura 2.1). El experimento pretende validar que:

- La teoría expuesta para la sábana de Markov a redes Bayesianas dinámicas es correcta
- La esencia cronológica de los datos es importante al clasificar los datos
- Se mejora la uniformidad de los resultados obtenidos por el método Media del experimento anterior.

Para el experimento se utilizó:

- Pseudocódigo para red bayesiana estática: método Derivada (sección 3.2.3)
- Algoritmos de construcción de redes Bayesianas usando la librería *bnlearn* para R (Scutari, 2010); PC, GS, IAMB, Fast-IAMB, Inter-IAMB, IAMB-FDR, Tabu, HC, MMHC, RSMAX2, H2PC, MMPC y SI-HITON-PC
- G1DBN librería de R para obtención de las interconexiones descritas en la sección 3.1.4
- Diseño e implementación de la sábana de Markov a RBD, sección 3.2.3
- Máquina HMM para la clasificación de las variables seleccionadas descrita en la sección 3.1.3
- Archivos *A1* y *A2* con modalidad de cinco clases.

Los resultados se presentan en la tabla 4.8.

Tabla 4.7 Resultados obtenidos al aplicar la propuesta de la sábana de Markov a RBD

Algoritmo	Accuracy	
	Red Inicial	Red Dinámica
PC	NS	0.1194
GS	NS	0.1194
IAMB	0.0640	0.0909
Fast-IAMB	NS	0.1194
Inter-IAMB	0.0640	0.0909
IAMB-FDR	0.0640	0.0909
HC	0.0585	0.0585
Tabu	0.0585	0.0585
MMHC	0.0640	0.0909
RSMAX2	0.0719	0.0909
H2PC	0.0585	0.0585
MMPC	0.0640	NS
SI-HILTON-PC	0.0640	NS

¹NS expresa que no se encontró sábana con el algoritmo especificado

La tabla 4.8 muestra que en el 84.6% de los casos, los datos evolutivos aumentaron la clasificación. Con ello se puede presumir que la teoría de la sábana de Markov a RBD se desempeña de forma correcta. Sin embargo, se buscará mejorar los resultados ya que, dada la medida *Accuracy* para todos los casos, se obtienen resultados no satisfactorios en comparación con los resultados exhaustivos, tabla 4.5.

4.6 Experimento #6

Este experimento tiene como objetivo validar la teoría de la sábana de Markov a RBD expresada en la sección 2.3.1, con la metodología de aprendizaje² de RBD expresada en (Sucar, 2006). El experimento se diseñó en formato de dos y cinco clases de los *datasets* descritos en la sección 3.1.2. La justificación se basa en los resultados del experimento 5, en donde se muestra un incremento de la medida *Accuracy* para los experimentos con la sábana a RBD, sin embargo, los resultados de clasificación muestran que el enfoque del método Derivada no se desempeña adecuadamente dados los resultados de la tabla 4.5 (clasificación exhaustiva con formato cinco clases).

El experimento pretende validar que:

- La teoría de sábana de Markov propuesta en esta investigación es correcta
- Las redes Bayesianas Dinámicas y sábana de Markov son una solución a la selección de variables de datos crono-valorados.

En el experimento se utilizó:

- Pseudocódigo para red bayesiana estática: método Sucar (sección 3.2.3)
- Algoritmos de construcción de redes Bayesianas usando la librería *bnlearn* para R (Scutari, 2010); PC, GS, IAMB, Fast-IAMB, Inter-IAMB, IAMB-FDR, Tabu, HC, MMHC, RSMAX2, H2PC, MMPC y SI-HITON-PC
- G1DBN librería de R para obtención de las interconexiones descritas en la sección 3.1.4
- Diseño e implementación de la sábana de Markov a RBD, sección 3.2.3
- Máquina HMM para la clasificación de las variables seleccionadas descrita en la sección 3.1.3
- Archivos *A1-A2*, *B1-B3* y *C1-C3* (entrenamiento-prueba), con modalidad de cinco clases.

Los resultados se presentan en la tabla 4.9.

Tabla 4.8 Medidas de clasificación a los *datasets* con formato cinco clases

Archivos entrenamiento	<i>Accuracy</i>	<i>Sensitivity (Recall)</i>				
		<i>Hold</i>	<i>Preparation</i>	<i>Rest</i>	<i>Retraction</i>	<i>Stroke</i>
A2	0.4098	0.0000	0.2150	0.9635	0.0000	0.0000
B3	0.3294	0.0000	1.0000	0.0000	0.0000	0.0000
C3	0.4468	0.2847	0.0798	0.8895	0.5069	0.3805

² Aprendizaje y construcción de redes Bayesianas serán utilizados como sinónimos a lo largo del documento

Con la tabla 4.9 se puede concluir que el método Sucar y la sábana propuesta en esta investigación se desempeñan mejor que el método Derivada. Para poder concluir si la selección hecha por la sábana a redes dinámicas se desempeña mejor que la sábana a las redes estáticas, se proporcionan las gráficas de las figuras 4.2.

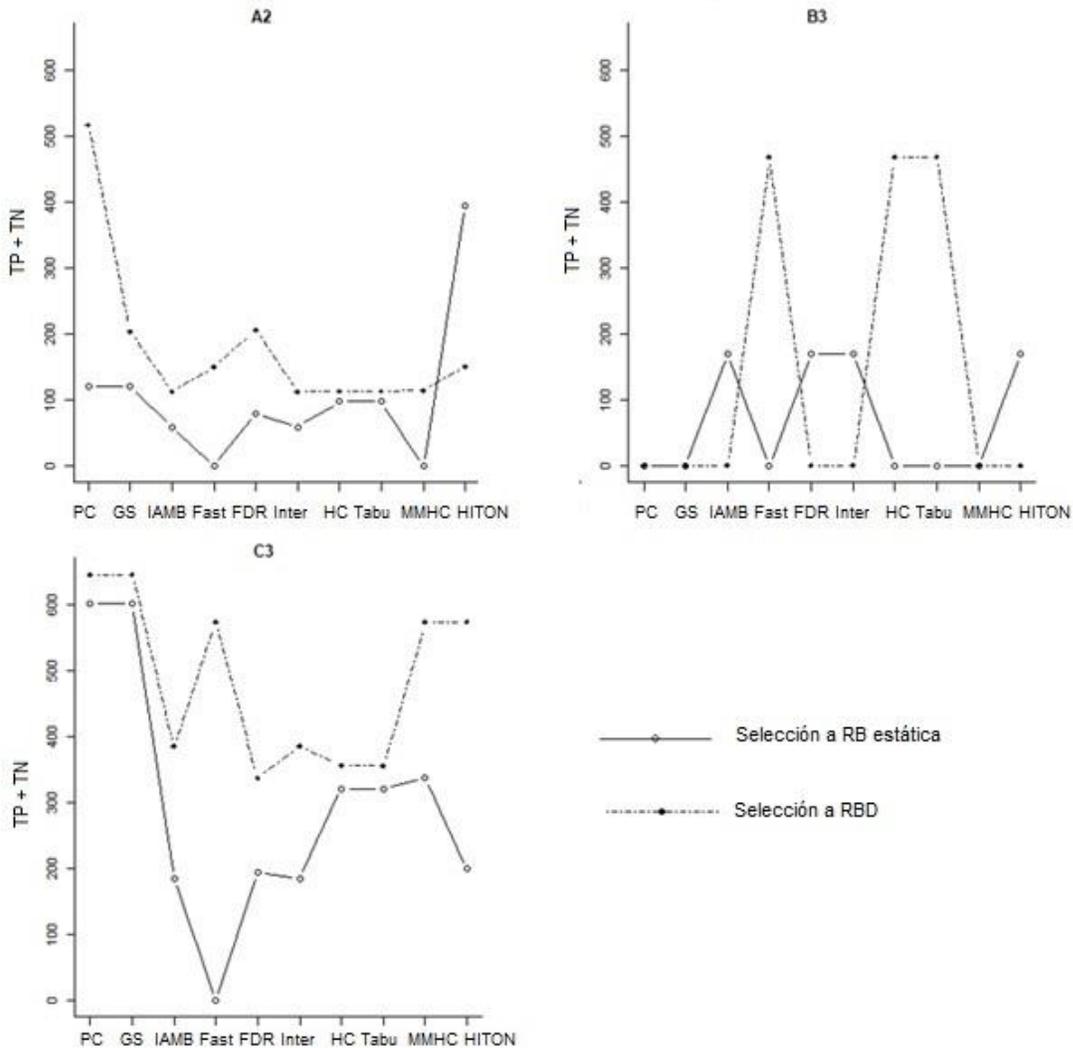


Figura 4.2 Comportamiento de los resultados utilizando la selección de variables a RB y RBD

Dadas la figura 4.2 se puede asegurar que en el 83% de los casos los resultados del *Accuracy* para la selección de variables en redes Bayesianas dinámicas, utilizando la teoría propuesta en esta investigación, superó los resultados de los obtenidos a redes estáticas. Los valores de *p-value* son los siguientes: a) archivos A1 y A2 igual a 0.00139237; b) archivos B1 y B3 igual a 0.04150943 y c) archivos C1 y C3 igual a 0.00212479.

4.7 Experimento #7

El objetivo de este experimento es la introducción de la función de comparación entre objetos. La justificación del mismo se basa en que la experimentación presentada hasta este momento ha dado resultados positivos, pero sin ser sobresalientes. Por esta razón se está introduciendo la prueba con el enfoque de redes Bayesianas con la función de comparación entre objetos.

El experimento pretende validar que:

- La función de comparación entre objetos toma en cuenta la evolución de los datos
- La función de comparación entre objetos compara dos objetos de diferente longitud
- La función de comparación entre objetos sirve para encontrar la información extra entre los objetos de una misma clase en crono-valorados
- La función de comparación entre objetos comprime los datos a un valor puntual que encapsula las propiedades de los datos
- Los métodos de aprendizaje de RB que mejor se comportan para el descubrimiento de relaciones entre variables.

Para el experimento se utilizó:

- Diagrama de flujo y pseudocódigo para la Función de comparación entre objetos (sección 3.2.1).
- Umbral de decisión (sección 3.2.2): porcentaje del objeto más corto
- Algoritmos de construcción de redes Bayesianas usando la librería *bnlearn* para R (Scutari, 2010); PC, GS, IAMB, Fast-IAMB, Inter-IAMB, IAMB-FDR, Tabu, HC, MMHC, RSMAX2, H2PC, MMPC y SI-HITON-PC
- Máquina HMM para la clasificación de las variables seleccionadas descrita en la sección 3.1.3
- Archivos *A1-A2*, *B1-B3* y *C1-C3* (entrenamiento-prueba), con modalidad de cinco clases y dos clases.

Resultados:

La evidencia y los resultados son presentados en la tabla 4.10. Se encontró el mejor *Accuracy* de clasificación para cada uno de los trece algoritmos seleccionados para este estudio. La mejor clasificación dada la sábana de Markov para el estado inicial BN se muestra en la tercera columna. El comportamiento mostrado para el mejor algoritmo de clasificación en cada prueba se presenta en las sucesivas figuras (de acuerdo a la columna cinco) y cada una de estas ilustra la clasificación de acuerdo con el umbral aplicado.

Tabla 4.9 Resultados experimento siete

Archivo	Número de clases probadas	Mejor clasificación (<i>accuracy</i>)	Algoritmo donde se encontró el mejor <i>accuracy</i>	Gráfica de comportamiento
A1/A2	5	555 /1264	<i>Max-Min Hill-Climbing</i>	Figura 4.4
A1/A2	2	1050 /1264	<i>Incremental Association with FDR / Hybrid HPC</i>	Figura 4.5
B1/B3	5	471/1424	<i>Hill-Climbing / Tabu Search</i>	Figura 4.6
B1/B3	2	1231/1424	<i>Incremental Association / Incremental Association with FDR / Interleaved Incremental Association / Max-Min Parents and Children</i>	Figura 4.7
C1/C3	5	721 /1448	<i>Hill-Climbing / Tabu Search / Max-Min Hill-Climbing / Restricted Maximization / Hybrid HPC</i>	Figura 4.8
C1/C3	2	1410/1448	<i>Hill-Climbing / Tabu Search</i>	Figura 4.9

En la tabla 4.10 se vinculan los mejores resultados, así como los algoritmos que obtuvieron dichos resultados, con una gráfica. Dicha gráfica pretende mostrar el comportamiento del *accuracy* a través de las variaciones del umbral. Como se mencionó anteriormente, el umbral se estableció como un porcentaje del objeto más corto. La figura vinculada muestra dicho porcentaje.

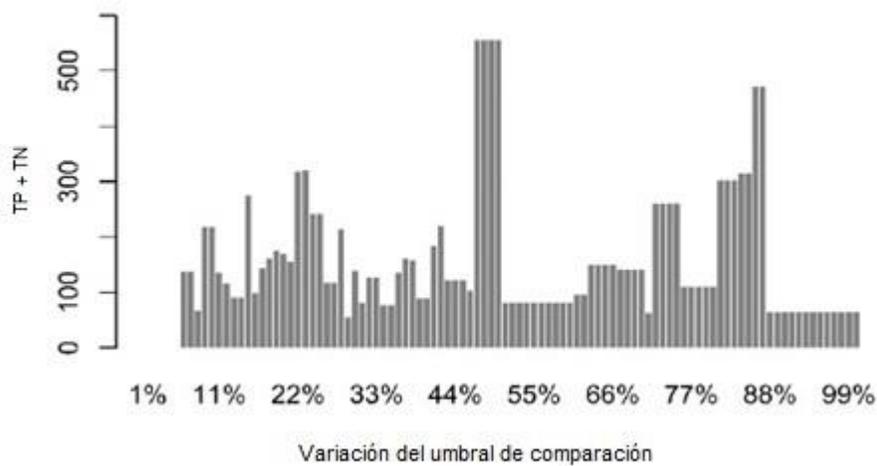


Figura 4.3 Gráfica de comportamiento archivos A1/A2

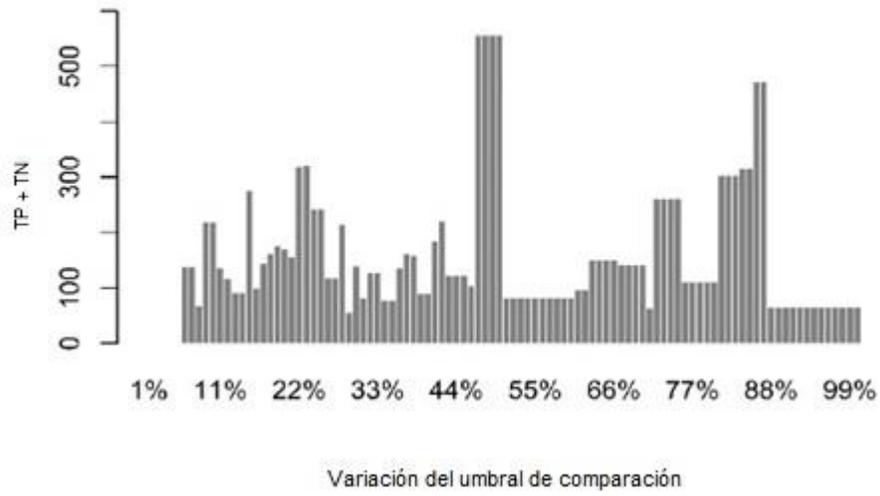


Figura 4.4 Gráfica de comportamiento archivos A1/A2

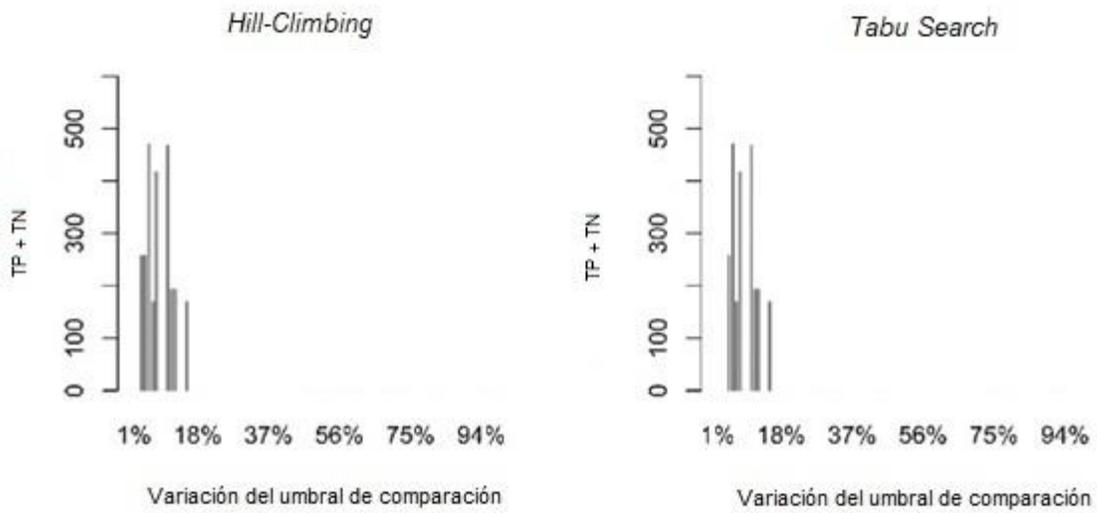


Figura 4.5 Gráfica de comportamiento archivos B1/B3

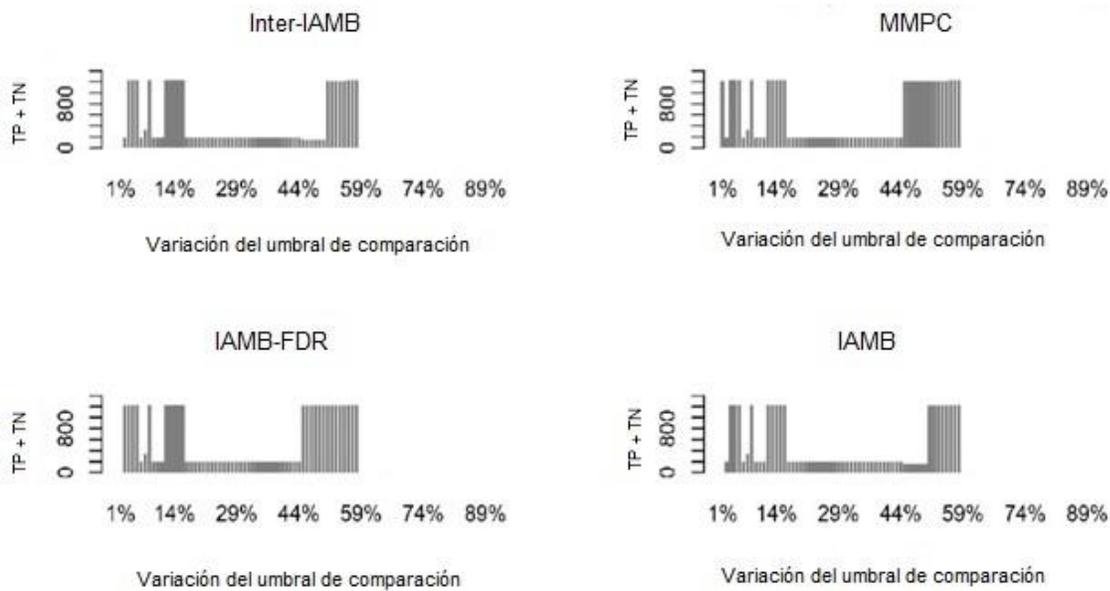


Figura 4.6 Gráfica de comportamiento archivos B1/B3

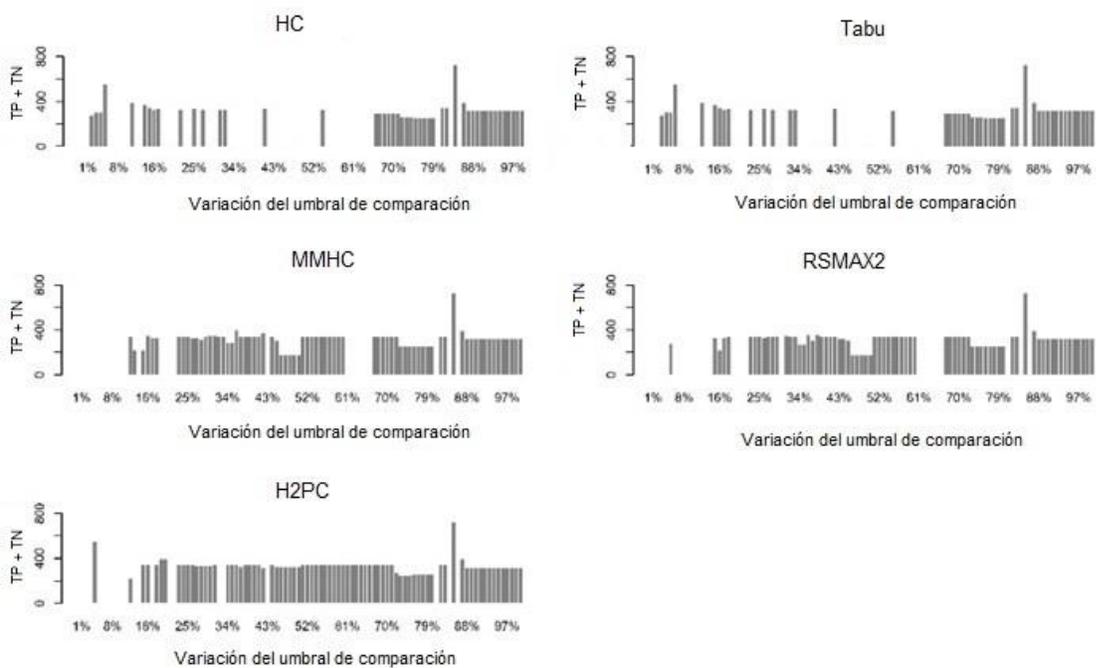


Figura 4.7 Gráfica de comportamiento archivos C1/C3

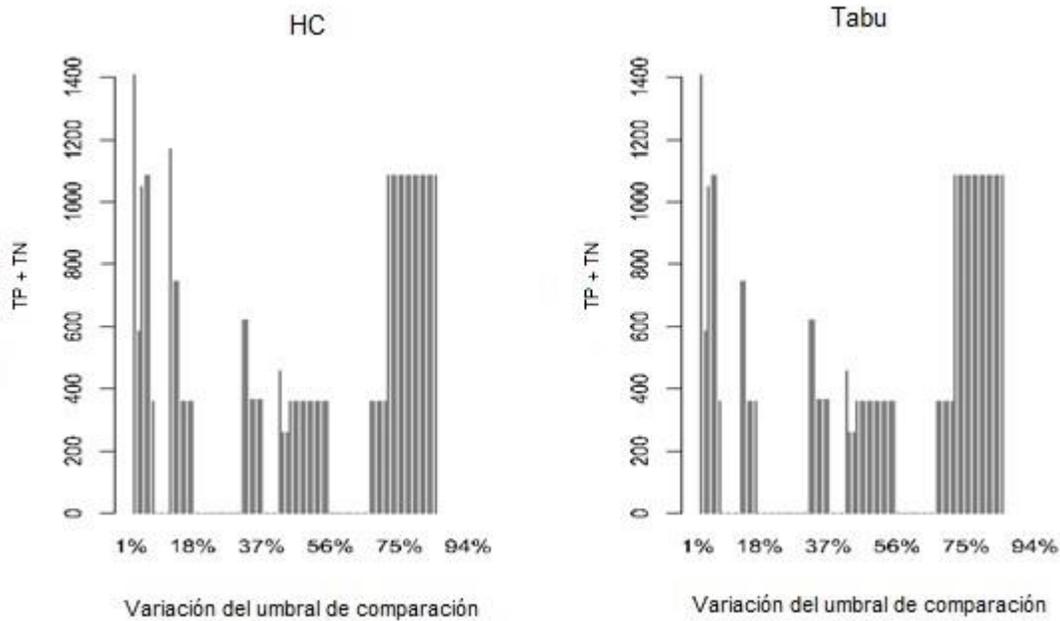


Figura 4.8 Gráfica de comportamiento archivos C1/C3

Dada la tabla 4.10 se puede asumir que la función de comparación entre objetos, propuesta en este estudio, compacta los objetos crono-valuados en una representación única conservando la particularidad evolutiva de objetos multi-muestra y comparando objetos de diferente longitud, características de los datos crono-valuados.

Con las figuras 4.4- 4.9 se puede comprobar que no hay un algoritmo específico (*constraint-based*, *score-based* o *hybrid*) que adquiera los mejores resultados, sino una combinación de ellos en cada mejor *accuracy* encontrado. Con esto, se seguirá utilizando los trece algoritmos en el resto de los experimentos. *P-values*: a) cinco clases, archivos A1 y A2 igual a 0.00015259; b) dos clases, archivos A1 y A2, igual a 0.0030861.

4.8 Experimento #8

El objetivo del presente experimento es conocer el desempeño del enfoque con la LCPR utilizando la función de comparación entre objetos. La justificación del mismo se da ya que los resultados del experimento anterior (siete) son los mejores hasta este punto y se debe verificar que la función tenga el mismo desempeño para otro seleccionador de variables.

El experimento pretende validar que:

- La función de comparación entre objetos tenga resultados similares con el enfoque LCPR y el enfoque RB/MB
- Existe diferencia en los resultados de *Accuracy* al cambiar el porcentaje del promedio de longitud al establecer el umbral.

Para el experimento se utilizó:

- Diagrama de flujo y pseudocódigo para la Función de comparación entre objetos (sección 3.2.1)
- Umbral de decisión (sección 3.2.2): porcentaje del promedio de longitud de los dos objetos
- Diseño e implementación de la adaptación del algoritmo LCRP, sección 3.2.4
- Máquina HMM para la clasificación de las variables seleccionadas descrita en la sección 3.1.3
- Archivos A1-A2 (entrenamiento-prueba), con modalidad de cinco clases y dos clases.

Resultados del experimento se presentan en la tabla 4.11.

Tabla 4.10 Resultados al cambiar el porcentaje de promedio del umbral utilizando LCRP

Porcentaje de longitud media	Número de testores	Testor con mejor clasificación	<i>Accuracy</i>
80%	20	sx; hx, sy	0.3900
70%	26	sx; hx, sy	0.3900
60%	50	lhz, sy	0.4422
20%	776	sx, lwx, lwz, rwz	0.4042
30%	315	hz, sx, rwy	0.4303
40%	169	hy, sx, sz	0.4303
50%	96	sx, rwy	0.4232
70% dos clases	18	lhz	0.7397

Dados los resultados en la tabla 4.11, se puede concluir que la variación al porcentaje del umbral representa la importancia al comparar objetos de diferente longitud, esto se puede apreciar en los resultados. Así como también, se puede establecer que la función de comparación entre objetos se comporta favorablemente con la LCRP. *P-values* de mejores resultados: a) dos clases igual a 0.02154168 y b) cinco clases igual a 0.00012589.

Dado el *accuracy* máximo para los archivos A1, A2 en modalidad de cinco y dos clases (tablas 4.5 y 4.6) no se justifica el uso de medidas adicionales al *accuracy* para este experimento.

4.9 Experimento #9

El objetivo del presente experimento es mostrar cómo se establecieron los porcentajes de umbral de acuerdo a la longitud de los objetos. Como se confirmó en los experimentos anteriores, los resultados varían de acuerdo al porcentaje que se utilice para establecer el umbral de decisión. Tal es la justificación de este experimento. Se pretende un experimento semi-exhaustivo utilizando el enfoque de las RB/MB.

La decisión de si dos objetos son diferentes se basa en: el resultado del algoritmo CLS (sección 2.3.2 paso 5, metodología; y la sección 3.2.1 paso 5 diagrama de flujo y pseudocódigo) y la longitud de los dos objetos que se comparan. Si el resultado del algoritmo CSL supera un umbral, los objetos se consideran similares.

El umbral se establece en base a la longitud de los dos objetos observados. Por dar un ejemplo: si la longitud del objeto más corto es mayor o igual al 14 por ciento del objeto más largo, el umbral se establece en 39%; de lo contrario, el umbral se establece en 51%.

Para establecer los porcentajes del umbral, se experimentó con un rango para cada uno de ellos. Los rangos en el experimento se establecieron como:

- a) 5 a 20 por ciento (*rango a*): rango que se exploró para establecer qué parte del objeto más corto debe considerarse en comparación con el más grande,
- b) 30 a 60 por ciento (*rango b*): rango cuando el objeto más corto cumple con la longitud marcada por porcentaje en *a*,
- c) 30 a 54 por ciento (*rango c*): rango cuando el objeto más corto no cumple con la longitud marcada por porcentaje en *a*.

Con el presente experimento se pretende validar que:

- El tomar en cuenta la longitud de los objetos conlleva a mejores resultados

Para el experimento se utilizó:

- Diagrama de flujo y pseudocódigo para la Función de comparación entre objetos (sección 3.2.1)
- Umbral de decisión (sección 3.2.2): Umbral semi-exhaustivo: toma en cuenta la longitud de los objetos
- Algoritmos de construcción de redes Bayesianas usando la librería *bnlearn* para R (Scutari, 2010); PC, GS, IAMB, Fast-IAMB, Inter-IAMB, IAMB-FDR, Tabu, HC, MMHC, RSMAX2, H2PC, MMPC y SI-HITON-PC
- Función de sábana de Markov incluida en la librería *bnlearn* para R

- Máquina HMM para la clasificación de las variables seleccionadas descrita en la sección 3.1.3
- Archivos A1-A2 (entrenamiento-prueba), con modalidad de dos clases.

Resultados del presente experimento:

A continuación, se muestra los resultados para los archivos A1 y A2 utilizando el enfoque LCRP. Los resultados que se muestran en la Figura 4.9, en donde el eje representa los rangos mencionados anteriormente (incisos *a*, *b* y *c* de esta sección). La Figura 4.9 representa el TP + TN máximo (verdadero positivo más verdadero negativo) encontrado en la intersección de los elementos de los tres rangos.

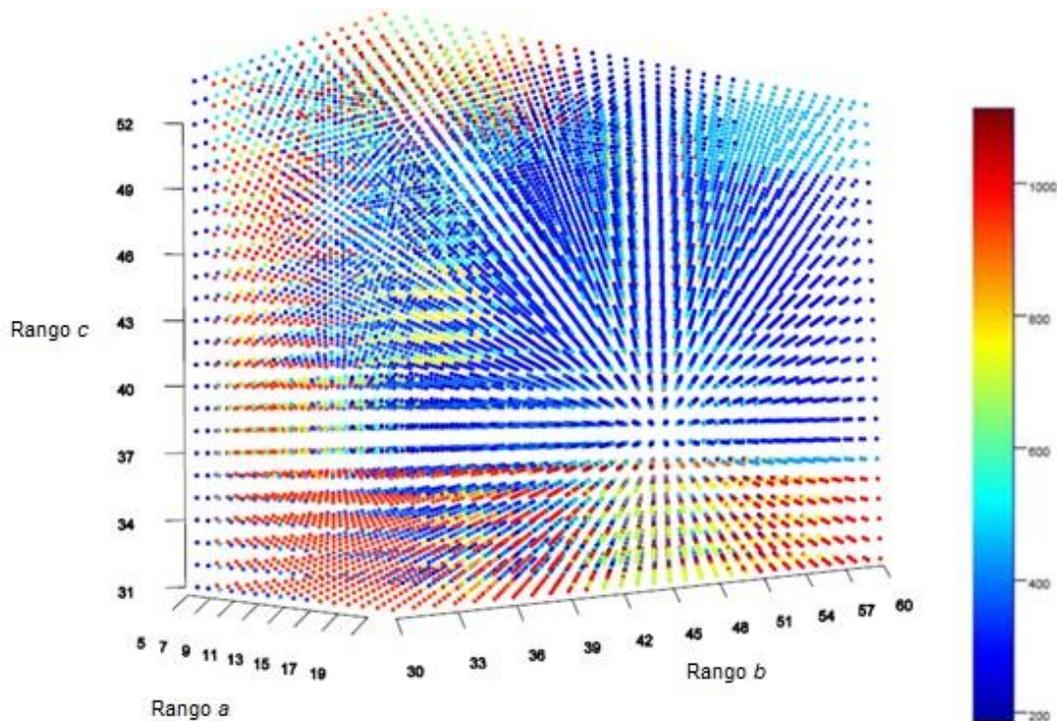


Figura 4.9. Máximos de la intersección de los tres rangos explorados

Para aumentar la claridad de los resultados de la Figura 4.9 se presenta la Figura 4.10. Donde se desglosan los resultados sobre 1000 (a), 1050 (b), 1100 (c) y 1115 (d) $TP + TN$ encontrados en los rangos establecidos anteriormente.

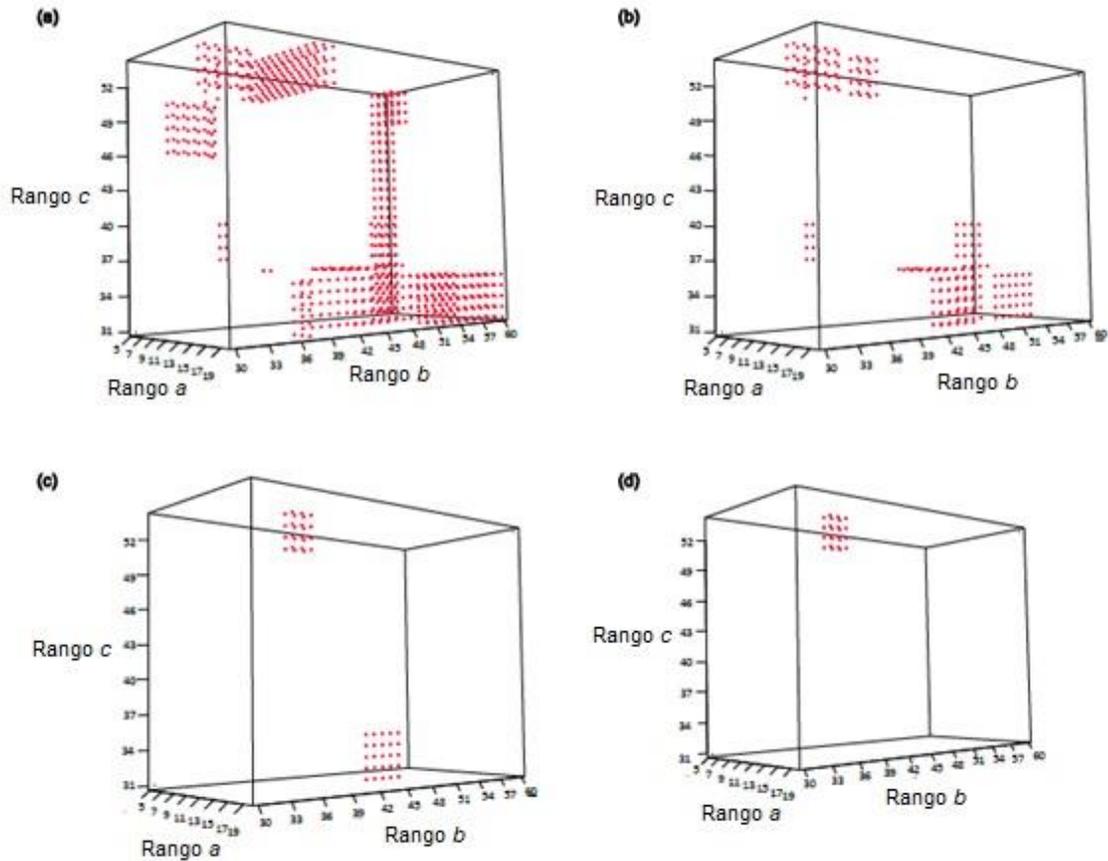


Figura 4.10 $TP + TN$ arriba de 1000 para las intersecciones en los rangos a, b y c a, b and c

Dada las figuras 4.9 y 4.10 se puede concluir que los mejores resultados de la búsqueda semi-exhaustiva para el umbral se encontraron en: rango a) 13-17; rango b) 33-39; y rango c) 49-52. Esta búsqueda semi-exhaustiva necesita ser aplicada por cada *dataset* o configuración.

4.10 Experimento #10

El presente experimento tiene como objetivo comprobar el desempeño del umbral semi-exhaustivo. LA justificación se basa en el experimento 9, donde los resultados para los archivos A1 y A2 incrementaron el *Accuracy* que se tenía de experimentos pasados. Para esto se utilizará el enfoque de las RB con la función de comparación entre objetos. Los archivos utilizados para la comprobación serán A1, A2, B1, B3, C1, y C3 para dos y cinco clases.

Con el presente experimento se pretende validar que:

- La idea de tomar en cuenta el tamaño de cada objeto para establecer el umbral es crucial para la decisión de si dos objetos son similares o no.
- Se mejoran los resultados en comparación a otros enfoques.

Para el experimento se utilizó:

- Diagrama de flujo y pseudocódigo para la Función de comparación entre objetos (sección 3.2.1)
- Umbral de decisión (sección 3.2.2): Umbral semi-exhaustivo: toma en cuenta la longitud de los objetos
- Algoritmos de construcción de redes Bayesianas usando la librería *bnlearn* para R (Scutari, 2010); PC, GS, IAMB, Fast-IAMB, Inter-IAMB, IAMB-FDR, Tabu, HC, MMHC, RSMAX2, H2PC, MMPC y SI-HITON-PC
- Función de sábana de Markov incluida en la librería *bnlearn* para R
- Máquina HMM para la clasificación de las variables seleccionadas descrita en la sección 3.1.3
- Archivos *A1-A2*, *B1-B3* y *C1-C3* (entrenamiento-prueba), con modalidad de dos clases.

Resultados del experimento se presentan en las tablas 4.12 y 4.13.

Tabla 4.11 Resultados para el enfoque BN / MB para dos clases

Archivos		Accuracy	Sensitivity Rest	Precision		Parámetros del umbral
Train	Test			Rest	No Rest	
A1	A2	0.8821	0.9148	0.8082	0.9405	a)14; b)39;c)51
B1	B3	0.8645	0.0051	1.000	0.8643	a)30; b)50;c)14
C1	C3	0.9012	0.9669	0.7277	0.9876	a)61; b)8; c)60

Tabla 4.12 Resultados para el enfoque BN / MB para cinco clases

Archivos		Accura- cy	Precision				Parámetros del umbral	
Train	Test		Hold	Prepara- tion	Rest	Retrac- tion		Stroke
A1	A2	0.4438	0.0	0.1748	0.5647	NaN	NaN	a)51; b)20; c)76
B1	B3	0.3294	NaN	0.3294	NaN	NaN	NaN	a)31; b)15; c)5
C1	C3	0.1989	0.0	0.1919	NaN	0.0	0.25	a)31; b)86; c)28

¹NaN significa que no se hicieron predicciones para la clase específica usando el umbral correspondiente.

Dadas las tablas 4.12 y 4.13 se puede corroborar que, el establecer el umbral conforme a la longitud de los objetos es un aspecto que afecta positivamente los resultados por lo que se considera continuar su exploración. *P-values*: a) cinco clases, archivos *A1* y *A2* igual a 0.00010681; b) dos clases, archivos *A1* y *A2*, igual a 0.00014877.

4.11 Experimento #11

El presente experimento tiene como objetivo comprobar el desempeño del umbral semi-exhaustivo. La justificación se basa en el experimento 10, donde los resultados incrementaron el *Accuracy* que se tenía de experimentos pasados. Para esto se utilizará el enfoque de las LCRP con la función de comparación entre objetos. Los archivos utilizados para la comprobación serán A1, A2, B1, B3, C1, y C3 para dos y cinco clases.

Con el presente experimento se pretende validar que:

- La función de comparación con el umbral semi-exhaustivo se puede considerar estable al ser validados con la LCRP como segundo selector de atributos.

Para el experimento se utilizó:

- Diagrama de flujo y pseudocódigo para la Función de comparación entre objetos (sección 3.2.1).
- Umbral de decisión (sección 3.2.2): Umbral semi-exhaustivo: toma en cuenta la longitud de los objetos
- Diseño e implementación de la adaptación del algoritmo LCRP, sección 3.2.4
- Función de sábana de Markov incluida en la librería *bnlearn* para R
- Máquina HMM para la clasificación de las variables seleccionadas descrita en la sección 3.1.3.
- Archivos *A1-A2*, *B1-B3* y *C1-C3* (entrenamiento-prueba), con modalidad de dos clases.

Los resultados del presente experimento se muestran en las tablas 4.14 y 4.15.

Tabla 4.13 Resultados para el enfoque LCRP utilizando la función de comparación entre objetos, dos clases

Archivos		<i>Accuracy</i>	<i>Sensitivity</i> <i>Rest</i>	<i>Precision</i>		Parámetros para umbral
<i>Train</i>	<i>Test</i>			<i>Rest</i>	<i>No Rest</i>	
A1	A2	0.8774	0.9331	0.7904	0.9516	a)14; b)39;c)51
B1	B3	0.8638	0	<i>NaN</i>	0.8638	a)80; b)20;c)80
C1	C3	0.9772	0.9613	0.9482	0.9870	a)61;b)15; c)20

¹*NaN* significa que no se hicieron predicciones para la clase específica usando el umbral correspondiente.

Tabla 4.14 Resultados para el enfoque LCRP utilizando la función de comparación entre objetos, cinco clases

Archivos		<i>Accuracy</i>	<i>Precision</i>					Parámetros para umbral
<i>Train</i>	<i>Test</i>		<i>Hold</i>	<i>Preparation</i>	<i>Rest</i>	<i>Retraction</i>	<i>Stroke</i>	
A1	A2	0.4304	<i>NaN</i>	0.39285	0.5419	0.1250	<i>NaN</i>	a)51; b)20; c)76
B1	B3	0.3294	<i>NaN</i>	0.3294	<i>NaN</i>	<i>NaN</i>	<i>NaN</i>	a)51; b)30; c)29
C1	C3	0.52	0.36782	0.16959	0.7558	0.38889	0.5	a)51; b)30; c)70

¹*NaN* significa que no se hicieron predicciones para la clase específica usando el umbral correspondiente.

Como se puede observar, en las tablas 4.14 y 4.15, los resultados son admisibles; lo anterior concluye que la función de comparación entre objetos (utilizando el enfoque que utiliza la longitud de los objetos para establecer el umbral) es estable entre algoritmos de selección de variables. *P-values*: a) dos clases, archivos A1 y A2, igual a 0.00020218; b) cinco clases, archivos A1 y A2, igual a 0.00037003.

4.12 Experimento #12

Al encontrar, con los experimentos 10 y 11, que el tamaño de los objetos influye al establecer el umbral, se ha propuesto una heurística para encontrar el porcentaje adecuado automáticamente; siendo así, la justificación del presente experimento. El objetivo es corroborar que la heurística propuesta obtiene resultados aceptables en comparación con los mejores encontrados hasta este momento.

Con el presente experimento se pretende validar que:

- La heurística propuesta puede encontrar los porcentajes apropiados para establecer el umbral de decisión automáticamente.

Para el experimento se utilizó:

- Diagrama de flujo y pseudocódigo para la Función de comparación entre objetos (sección 3.2.1).
- Umbral con heurística (sección 3.2.3): de acuerdo a tamaño de los objetos, automatizado
- Algoritmos de construcción de redes Bayesianas usando la librería *bnlearn* para R (Scutari, 2010); PC, GS, IAMB, Fast-IAMB, Inter-IAMB, IAMB-FDR, Tabu, HC, MMHC, RSMAX2, H2PC, MMPC y SI-HITON-PC
- Función de sábana de Markov incluida en la librería *bnlearn* para R
- Máquina HMM para la clasificación de las variables seleccionadas descrita en la sección 3.1.3.
- Archivos A1-A2 (entrenamiento-prueba), con modalidad de dos clases.

Resultados del presente experimento:

Tabla 4.15 Resultados utilizando heurística para umbral y enfoque RB/MB

Archivo		No. De Clases	Accuracy
<i>Trainnig</i>	<i>Testing</i>		
A1	A2	2	0.8651
A1	A2	5	0.4084

Dado los resultados mostrados en la tabla 4.16 se puede concluir que la heurística propuesta, para la adquisición automática de los porcentajes de la longitud de cada objeto a ser considerada para el umbral de decisión, presenta un *p-value* de 0.00059891 para el formato de dos clases y *p-value* igual a 0.00139237 para formato de cinco clases. Además, la adquisición de los parámetros necesarios se realiza en un tiempo despreciable en comparación de la búsqueda semi-exhaustiva de los experimentos 9, 10 y 11.

Análisis de los resultados

Con el experimento 1 del capítulo 4 se observa que usar todas las variables de un *dataset* resulta en *Accuracy* bajo debido a los atributos redundantes o irrelevantes que confunden al algoritmo (Koller & Sahami, 1996).

Con los experimentos 2 y 3 se comprueba que los resultados de la clasificación mejoran gracias a la selección. Debido a que los experimentos se diseñaron para encontrar la mejor clasificación de manera que la búsqueda fuera exhaustiva, se puede asegurar que los resultados comprueban que el problema de *sobre ajuste* se encuentra en las variables irrelevantes o redundantes. Este experimento sirve también para contrastar los resultados de las propuestas con los mejores resultados de los experimentos (dos y tres, capítulo 4).

El experimento cuatro muestra que los métodos de aprendizaje basados en restricciones (PC, GS, IAMB, FastIAMB, InterIAMB y AIMB-FDR) obtienen resultados similares, sin embargo se encontraron observaciones extremas (*outliers*). Los métodos basados en evaluaciones consiguen resultados iguales y los mixtos tienen la mayor variación. Con estos resultados se observan dos cosas: 1) la necesidad de utilizar los tres métodos para el aprendizaje de RB, ya que ninguno de ellos muestra un rendimiento destacado constante; y 2) los resultados muestran que los subconjuntos seleccionados no son los óptimos, esto debido a que la medida *media* no toma en cuenta la posición de los datos, lo cual al final, repercute en el aprendizaje de la RB.

El experimento cinco aplica el enfoque de las RB, en específico las RBD. En el experimento se construye la parte de las RB utilizando el método derivada. Después se utiliza la información temporal para construir la RBD. La selección de variables se obtiene implementando la formulación matemática aquí propuesta de la sábana de Markov a redes dinámicas. Con este enfoque se obtiene un incremento de *Accuracy* en el 84.6% de los ejercicios al utilizar la parte dinámica de las redes (es decir, la parte evolutiva de los datos) en lugar de solo usar la parte estática obtenida con la derivada.

La experimentación también incluyó la teoría propuesta por Sucar (Sucar, 2006). Con esta teoría y las dependencias entre tiempos se construyó la RBD; para después aplicar la propuesta de la sábana de Markov (experimento seis). Los resultados fueron sobresalientes, mejorando los resultados hasta este punto. Con estos resultados y los resultados del experimento cinco se puede concluir que la formulación matemática de la sábana es correcta.

La implementación de la función de comparación entre objetos se hace en el experimento siete, donde se encuentra que los resultados con cinco clases sobrepasan los resultados con el mismo formato utilizando las RBD (experimentos seis y siete) aunque los resultados son muy cercanos (5% de incremento en el más amplio de los resultados). En los experimentos con 2 clases se encontró que se incrementó el *accuracy* en dos de tres experimentos y en uno se mantuvo. Esto reitera que la sábana propuesta en esta tesis está funcionando de manera favorable y que la función de comparación entre objetos está haciendo una compresión de los datos tomando en cuenta la evolución en el tiempo.

Para seguir comprobando la eficacia de la función de comparación entre objetos se diseñó el experimento ocho, en donde se modifica la forma de obtener el umbral de dicha función. Con esta modificación se alcanza un *accuracy* de 0.4422 para un formato de 5 clases, el más alto de este formato hasta ahora. Con esto se concluye que el umbral juega un papel importante al hacer la selección.

Dado que el umbral de la función de comparación representa un papel importante, se experimentó con una búsqueda semi-exhaustiva; la cual derivó en aumento en el *accuracy* de la clasificación del experimento anterior en 0.0016 puntos del experimento con el mismo formato. Sin embargo, la experimentación semi-exhaustiva no encontró resultados satisfactorios para los ejercicios con este formato en cinco clases (experimento 10).

Siguiendo con la experimentación del umbral de la función de comparación, se probó con el enfoque de la LCRP y el umbral semi-exhaustivo. En dicho experimento (experimento 11) se encontró que los valores en *Accuracy* son todos aceptables, alcanzando en el mejor resultado un 97.7% de efectividad al clasificar las muestras del archivo C3 con valores de *Sensitivity* y *Precision* por arriba de 94% en cada una de las clases. Con estos resultados preliminares se puede concluir que la función está compactando los datos de forma eficiente. Sin embargo, se requiere ejecutar una búsqueda exhaustiva (semi) para los valores del umbral.

Para evitar una búsqueda semi-exhaustiva para la obtención del umbral, se implementó una heurística. Los resultados del experimento 12 avalan la eficiencia de la heurística estando por debajo solo 0.01 puntos en *Accuracy* del experimento 11 en dos clases para los *datasets A1* y *A2*.

CAPÍTULO 5

CONCLUSIONES Y TRABAJO FUTURO

El objetivo principal de la investigación es presentar una técnica de selección de variables para datos crono-valorados que tomen en cuenta la evolución de los datos. En la tabla 5.1 se muestran los objetivos planteados, los experimentos que se realizaron y los resultados de estos.

Tabla 5.1 Experimento del trabajo de investigación contrastados con los objetivos

	<i>Técnica de Selección de Variables¹</i>	<i>Considera sentido evolutivo de datos</i>	<i>Conserva rendimiento entre métricas de clasificación</i>	<i>Tiempo computacional despreciable</i>	<i>Parámetros de umbral automáticos</i>	<i>p-value menor a 0.05</i>
Experimento 1	✗	✓	⊖	✓	⊖	✗
Experimento 2	✗	✓	⊖	✗	⊖	⊖
Experimento 3	✗	✓	⊖	✗	⊖	⊖
Experimento 4	✓	✗	⊖	✓	⊖	⊖
Experimento 5	✓	✓	✗	✓	⊖	✗
Experimento 6	✓	✓	✓	✓	⊖	✓
Experimento 7	✓	✓	✗	✓	✗	✓
Experimento 8	✓	✓	✓	✓	✗	✓
Experimento 9	✓	✓	⊖	✗	✗	⊖
Experimento 10	✓	✓	✗	✗	✗	✓
Experimento 11	✓	✓	✓	✗	✗	✓
Experimento 12	✓	✓	✓	✓	✓	✓

¹ Técnicas propuestas en esta tesis, en cualquiera de las modalidades, ver sección 2.2.

✓ Afirmación

⊖ No aplica / No generado

✗ Negación

El objetivo principal de la investigación es presentar una técnica de selección de variables para datos crono-valuados que tomen en cuenta la evolución de los datos. Los resultados de los experimentos seis y doce, muestran una posibilidad real de éxito de las propuestas hechas en esta tesis. Siendo los resultados de dichos experimentos publicados en las revistas *Traitement du Signal* (publicado) y *Journal of Applied Research and Technology* (en revisión).

En lo que a las hipótesis respecta, los valores de p -value de los experimentos 7-8, 10-12 rechazan la hipótesis nula número uno de la sección 1.1.2. Los valores p -value de los experimentos 6, rechazan la hipótesis nula número 2 de la misma sección.

Se concluye que las aportaciones pueden ser aplicadas a datos crono-valuados para la selección de variables para una mejor clasificación. Además, las técnicas presentadas, pueden ser aplicadas a los datos sin la necesidad de intervención de expertos para la extracción de variables o para el establecimiento del umbral; todo esto tomando en cuenta la propiedad evolutiva de los datos.

Aportaciones y productos académicos

Se introduce una formulación matemática para la sábana de Markov en RBD; dicha sábana funge como selector de características para modelado de datos con redes Bayesianas dinámicas. Con los resultados de la aplicación de la formulación se elaboró un artículo que se encuentra en revisión por *Journal of Applied Research and Technology* con el título de *Markov Blanket for gesture feature selection: a mathematical analysis and experimentation*.

Dado el modelo matemático de dos técnicas de selección de variables, se desarrolla una metodología para compactar los datos crono-valuados a datos lineales conservando la evolución de la información cronológica cuyos resultados son avalados por la publicación *Function to Flatten Gesture Data for Specific Feature Selection Methods to Improve clasification* en la revista *Traitement du Signal*. En el anexo E se presenta el resumen de dicho artículo.

Se presenta una heurística original para el establecimiento del umbral de decisión de similitud entre dos objetos.

Se justifica la necesidad de una técnica de selección de variables con la publicación del capítulo *Feature Selection for Pattern Recognition: Upcoming Challenges* del libro *Advanced Topics on Computer Vision, Control and Robotics in Mechatronics*. En el anexo D se presenta el resumen de dicho capítulo.

Trabajo futuro

La heurística para el establecimiento del umbral permanece como trabajo futuro. A pesar del hecho de que la heurística no es un objetivo de la presente tesis (su conclusión queda fuera del alcance de este trabajo de investigación) se pretende seguir la investigación y publicar los resultados.

Referencias

- Aguila Feros, L., & Ruiz-Shulcloper, J. (1984). Algoritmo MB para la elaboración de la información k-valente en problemas de Reconocimiento de Patrones. *Revista Ciencias Matemáticas*, 5(3).
- Alba-Cabrera, E., Santana, R., Ochoa-Rodriguez, A., & Lazo-Cortes, M. (2000). Finding typical testors by using an evolutionary strategy. In *Proceedings of the 5th Ibero American Symposium on Pattern Recognition* (p. 267).
- Alba Cabrera, E., & Lazo Cortés, M. (1998). Una solución global para la utilización de los testores en problemas de reconocimiento de patrones. *Memorias Del III Taller Iberoamericano de Reconocimiento de Patrones*, 209–217.
- Amir, A., Charalampopoulos, P., Pissis, S. P., & Radoszewski, J. (2020). Dynamic and Internal Longest Common Substring. *Algorithmica*, 82(12), 3707–3743. <https://doi.org/10.1007/s00453-020-00744-0>
- Ayala, G. (2015). *Estadística Básica con R*. (U. de Valencia, Ed.).
- Ayaquica, I. O. (1997). Un nuevo algoritmo de escala exterior para el cálculo de testores típicos. *Memorias Del II Taller Iberoamericano de Reconocimiento de Patrones*, 141–148.
- Bishop, C. M. (2016). *Pattern Recognition and Machine Learning*. Springer New York. Retrieved from <https://books.google.com.mx/books?id=kOXDtAEACAAJ>
- Bouckaert, R. R. (1995). *Bayesian belief networks: from construction to inference*.
- Bravo, A. (1983). Algoritmo CT para el cálculo de los testores típicos de una matriz k-valente. *Revista Ciencias Matemáticas*, 4(2), 123–144.
- Cervantes Salgado, M., & Pinto Elías, R. (2018). Feature selection for pattern recognition: Upcoming challenges. In *Advanced Topics on Computer Vision, Control and Robotics in Mechatronics* (pp. 77–100). https://doi.org/10.1007/978-3-319-77770-2_4
- Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), 16–28.
- Cherrington, M., Lu, J., Airehrour, D., Thabtah, F., Xu, Q., & Madanian, S. (2019). Feature Selection: Multi-source and Multi-view Data Limitations, Capabilities and Potentials. In *2019 29th International Telecommunication Networks and Applications Conference (ITNAC)* (pp. 1–6).
- Choi, H.-R., & Kim, T. (2018). Modified Dynamic Time Warping Based on Direction Similarity for Fast Gesture Recognition. *Mathematical Problems in Engineering*, 2018, 2404089. <https://doi.org/10.1155/2018/2404089>
- Cisotto, G., Capuzzo, M., Guglielmi, A. V., & Zanella, A. (2020). Feature selection for gesture recognition in Internet-of-Things for healthcare. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)* (pp. 1–6).
- Cortés, M. L., Díaz, G. S., & Gómez, T. Q. (n.d.). Evaluación de la relevancia de los rasgos en un problema de clasificación supervisada a partir de todos los testores. In *IV Simposio Iberoamericano de Reconocimiento de Patrones. Conferencia Internacional CIMAF* (Vol. 99, pp. 215–222).
- Curtis, P., Harb, M., Abielmona, R., & Petriu, E. (2016). Feature selection and neural network architecture evaluation for real-time video object classification. In *2016 IEEE Congress on Evolutionary Computation*

(CEC) (pp. 1038–1045).

- Di Leo, G., & Sardanelli, F. (2020). Statistical significance: p value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach. *European Radiology Experimental*, 4(1), 18. <https://doi.org/10.1186/s41747-020-0145-y>
- Diaz-Sanchez, G., Piza-Davila, I., Sanchez-Diaz, G., Mora-Gonzalez, M., Reyes-Cardenas, O., Cardenas-Tristan, A., & Aguirre-Salado, C. (2011). Typical testors generation based on an evolutionary algorithm. In *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 58–65).
- Dimitris Margaritis. (2003). *Learning Bayesian Network Model Structure from Data*. (University of Pittsburgh). Retrieved from <https://www.cs.cmu.edu/~dmarg/Papers/PhD-Thesis-Margaritis.pdf>
- Dmitriev, A. N., Zhuravlev, Y. I., & Krendeleiev, F. P. (1966). On the mathematical principles of patterns and phenomena classification. *Diskretnyi Anal*, 7(3), 15.
- Dua, D., & Graff, C. (2017). {UCI} Machine Learning Repository. Retrieved from <http://archive.ics.uci.edu/ml>
- Friedman, N., Murphy, K., & Russell, S. (1998). Learning the structure of dynamic probabilistic networks. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (pp. 139–147).
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Inc.
- Gong, R., & Chan, T. K. Y. (2006). Syllable alignment: A novel model for phonetic string search. *IEICE Transactions on Information and Systems*, 89(1), 332–339.
- Gupta, S., Jaafar, J., Ahmad, W. F. W., & Bansal, A. (2013). Feature extraction using MFCC. *Signal & Image Processing: An International Journal (SIPIJ)*, 4(4), 101–108.
- Hancer, E., Xue, B., Karaboga, D., & Zhang, M. (2015). A binary ABC algorithm based on advanced similarity scheme for feature selection. *Applied Soft Computing*, 36, 334–348.
- Hancer, E., Xue, B., & Zhang, M. (2018). Differential evolution for filter feature selection based on information theory and feature ranking. *Knowledge-Based Systems*, 140, 103–119.
- Hassanien, A. E., Suraj, Z., Slezak, D., & Lingras, P. (2007). *Rough Computing: Theories, Technologies and Applications: Theories, Technologies and Applications*. IGI Global.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*, Springer Series in Statistics. Springer New York.
- He, X., Cai, D., & Niyogi, P. (2006). Laplacian score for feature selection. In *Advances in neural information processing systems* (pp. 507–514).
- Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527–1554.
- Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. *Journal of Chemical Information and Modeling* (Vol. 53). MIT press. <https://doi.org/10.1017/CBO9781107415324.004>
- Koller, D., & Sahami, M. (1996). Toward optimal feature selection. In *ICML*.
- Korb, K., & Nicholson, A. (2010). *Bayesian Artificial Intelligence*. *Bayesian Artificial Intelligence* (2nd ed.). CRC Press. <https://doi.org/https://doi.org/10.1201/b10391>
- Krishna, R. S. B., & Aramudhan, M. (2014). Feature selection based on information theory for pattern classification. In *2014 International Conference on Control, Instrumentation, Communication and*

Computational Technologies (ICCICCT) (pp. 1233–1236).

- Kuschner, K. W., Malyarenko, D. I., Cooke, W. E., Cazares, L. H., Semmes, O. J., & Tracy, E. R. (2010). A Bayesian network approach to feature selection in mass spectrometry data. *BMC Bioinformatics*, *11*(1), 177. <https://doi.org/10.1186/1471-2105-11-177>
- Lazo-Cortés, M., & Ruiz-Shulcloper, J. (1995). Determining the feature relevance for non-classically described objects and a new algorithm to compute typical fuzzy testors. *Pattern Recognition Letters*, *16*(12), 1259–1265.
- Lazo-Cortés, M., Ruiz-Shulcloper, J., & Alba-Cabrera, E. (2001). An overview of the evolution of the concept of testor. *Pattern Recognition*, *34*(4), 753–762.
- Lèbre, S. (2013). G1DBN: A package performing Dynamic Bayesian Network inference. Version.
- Lèbre, Sophie. (2009). Inferring dynamic genetic networks with low order independencies. *Statistical Applications in Genetics and Molecular Biology*, *8*(1), 1–38.
- Li, G., Zhang, R., Ritchie, M., & Griffiths, H. (2017). Sparsity-driven micro-Doppler feature extraction for dynamic hand gesture recognition. *IEEE Transactions on Aerospace and Electronic Systems*, *54*(2), 655–665.
- Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys*. ACM. <https://doi.org/10.1145/3136625>
- Lichman, M. (2013). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Retrieved from <http://archive.ics.uci.edu/ml>
- Liu, X., Wang, L., Zhang, J., Yin, J., & Liu, H. (2013). Global and local structure preservation for feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(6), 1083–1095.
- Madeo, R. C. B., Lima, C. A. M., & Peres, S. M. (2013). Gesture unit segmentation using support vector machines: segmenting gestures from rest positions. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing* (pp. 46–52).
- Markham, K. (2014). Simple guide to confusion matrix terminology. Retrieved from <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/>
- Martínez-Trinidad, J. F., & Guzmán-Arenas, A. (2001). The logical combinatorial approach to pattern recognition, an overview through selected works. *Pattern Recognition*, *34*(4), 741–751.
- Morales-Manilla, L. R., & Sanchez-Diaz, G. (2007). FS-EX plus: a new algorithm for the calculation of typical FS-testor set. In *Iberoamerican Congress on Pattern Recognition* (pp. 380–386).
- Murty, M. N., & Devi, V. S. (2011). *Pattern recognition: An algorithmic approach*. Springer Science & Business Media.
- Myatt, G. J. (2007). *Making sense of data: a practical guide to exploratory data analysis and data mining*. John Wiley & Sons.
- Nassuna, H., Kim, J., Eyobu, O. S., & Lee, D. (2020). Feature Selection for Abnormal Driving Behavior Recognition Based on Variance Distribution of Power Spectral Density. *IEMEK Journal of Embedded Systems and Applications*, *15*(3), 119–127.
- Ochoa Somuano, J. (2016). *Algoritmo para la selección de variables en objetos crono-valorados*. Centro Nacional de Investigación y desarrollo tecnológico.
- Oracle. (2012). Comparison: Longest Common Substring. Retrieved July 1, 2021, from <https://www.oracle.com/webfolder/technetwork/data->

quality/edqhelp/Content/processor_library/matching/comparisons/longest_common_substring.htm

- Pearl, J. (2009). *Causality*. Cambridge university press.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.
- Pons-Porrata, A., Gil-García, R., & Berlanga-Llavori, R. (2007). Using typical testors for feature selection in text categorization. In *Iberoamerican Congress on Pattern Recognition* (pp. 643–652).
- R Core Team. (2018). R: A Language and Environment for Statistical Computing. Vienna, Austria. Retrieved from <https://www.r-project.org>
- RStudio Team. (2020). RStudio: Integrated Development Environment for R. Boston, MA. Retrieved from <http://www.rstudio.com/>
- Ruiz-Shulcloper, J., & Cortés, M. L. (1991). K-testores primos. *Revista Ciencias Técnicas Físicas y Matemáticas*, 9, 17–55.
- Ruiz-Shulcloper, J., Guzmán Arenas, A., & Martínez-Trinidad, J. F. (1999). Enfoque lógico combinatorio al reconocimiento de patrones. i. *Selección de Variables y Clasificación Supervisada, IPN, México*. article.
- Ruiz-Shulcloper, J., & others. (1985). Algoritmos BT y TB para el cálculo de todos los testores típicos. *Revista Ciencias Matemáticas*, 2, 11–18.
- Russell, S., & Norvig, P. (2013). *Artificial Intelligence A Modern Approach*. Prentice Hall. <https://doi.org/10.1017/S0269888900007724>
- Samadani, A. (2018). EMG Channel Selection for Improved Hand Gesture Classification. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 4297–4300).
- Sammut, C., & Webb, G. I. (Eds.). (2010). Data Set. In *Encyclopedia of Machine Learning* (p. 261). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-30164-8_196
- Sanchez-Diaz, G, Lazo-Cortes, M., & Fuentes-Chavez, O. (1999). Genetic algorithm for calculating typical testors of minimal cost. In *Proc. of the Iberoamerican Symposium on Pattern Recognition (SIARP 1999)* (pp. 207–213).
- Sanchez-Diaz, Guillermo, Piza-Davila, I., Lazo-Cortes, M., Mora-Gonzalez, M., & Salinas-Luna, J. (2010). A fast implementation of the CT_EXT algorithm for the testor property identification. In *Mexican International Conference on Artificial Intelligence* (pp. 92–103).
- Santiesteban-Alganza, Y., & Pons-Porrata, A. (2003). LEX: A new algorithm for calculating typical testors. *Revista Ciencias Matemáticas*, 21(1), 85–95.
- Santos, J., Carrasco, A., & Martínez, J. F. (2004). Feature selection using typical testors applied to estimation of stellar parameters. *Computación y Sistemas*, 8(1), 15–23.
- Santoyo, S. (2017). A Brief Overview of Outlier Detection Techniques. Retrieved from <https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561>
- Scutari, M. (2010). Learning Bayesian networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3), 1–22. <https://doi.org/10.18637/jss.v035.i03>
- Shanableh, T., Assaleh, K., & Al-Rousan, M. (2007). Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(3), 641–650.

- Singha, J., & Laskar, R. H. (2017). Hand gesture recognition using two-level speed normalization, feature selection and classifier fusion. *Multimedia Systems*, 23(4), 499–514.
- Somuano, J. O. (2016). *Algoritmo para la Selección de Variables en Descripciones Crono-Valuadas*. Centro Nacional de Investigación y Desarrollo Tecnológico.
- Spirtes, P., Glymour, C. N., Scheines, R., & Heckerman, D. (2000). *Causation, prediction, and search*. MIT press.
- Sridevi, T., & Murugan, A. (2014). A novel feature selection method for effective breast cancer diagnosis and prognosis. *International Journal of Computer Applications*, 88(11).
- State, U. O. P. (2017). Applied Data Mining and Statistical Learning. Retrieved from <https://onlinecourses.science.psu.edu/stat857/node/3>
- Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341–359. <https://doi.org/10.1023/A:1008202821328>
- Sucar, L. E. (2006). Redes bayesianas. *BS Araujo, Aprendizaje Automático: Conceptos Básicos y Avanzados*, 77–100.
- Svensén, M., & Bishop, C. M. (2007). Pattern recognition and machine learning. Springer.
- Tang, J., & Liu, H. (2014). Feature selection for social media data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(4), 19.
- Tsamardinos, I., Aliferis, C. F., Statnikov, A. R., & Statnikov, E. (2003). Algorithms for Large Scale Markov Blanket Discovery. In *FLAIRS conference* (Vol. 2, pp. 376–380).
- Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1), 31–78.
- Verma, T., Pearl, J., & others. (1991). Equivalence and synthesis of causal models.
- Visser, I., & Speekenbrink, M. (2010). {depmixS4}: An {R} Package for Hidden Markov Models. *Journal of Statistical Software*, 36(7), 1–21. Retrieved from <http://www.jstatsoft.org/v36/i07/>
- Wagner, P. K., Peres, S. M., Madeo, R. C. B., de Moraes Lima, C. A., & de Almeida Freitas, F. (2014). Gesture unit segmentation using spatial-temporal information and machine learning. In *The Twenty-Seventh International Flairs Conference*.
- Wang, H., Nie, F., & Huang, H. (2013). Multi-view clustering and feature learning via structured sparsity. In *International conference on machine learning* (pp. 352–360).
- Webb, A R, & Copsey, K. D. (2011). *Statistical Pattern Recognition*. Wiley. Retrieved from <https://books.google.com.mx/books?id=gb79v56894oC>
- Webb, Andrew R. (2003). *Statistical pattern recognition*. John Wiley & Sons.
- Wu, K., Chen, Z., & Li, W. (2018). A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks. *IEEE Access*, 6, 50850–50859. <https://doi.org/10.1109/ACCESS.2018.2868993>
- Yaramakala, S., & Margaritis, D. (2005). Speculative Markov blanket discovery for optimal feature selection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)* (pp. 4--pp).
- Yoon, H., Yang, K., & Shahabi, C. (2005). Feature subset selection and feature ranking for multivariate time series. *IEEE Transactions on Knowledge and Data Engineering*, 17(9), 1186–1198.
- Yuan, L., Liu, J., & Ye, J. (2011). Efficient methods for overlapping group lasso. In *Advances in neural*

information processing systems (pp. 352–360).

Zhang, H., Wang, J., Sun, Z., Zurada, J. M., & Pal, N. R. (2020). Feature Selection for Neural Networks Using Group Lasso Regularization. *IEEE Transactions on Knowledge and Data Engineering*, 32(4), 659–673.

Zhang, R., Nie, F., Li, X., & Wei, X. (2019). Feature selection with multi-view data: A survey. *Information Fusion*, 50, 158–167.

Zou, Q., Ni, L., Zhang, T., & Wang, Q. (2015). Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, 12(11), 2321–2325.

Anexo A

Código de la implementación de la librería depmixS4 para los ejercicios de clasificación en este documento. La librería depmixS4 es mencionada en la sección 3.1.4 como parte de la implementación del sistema.

```
1 library(depmixS4)
2
3 a1_raw_N <- read.csv("c1_raw_N_5clases.csv")
4 a2_raw_N <- read.csv("c3_raw_N_5clases.csv")
5 datos<-a1_raw_N[,c(1:18,21)]
6 df<-a2_raw_N[,c(1:18,21)]
7 nombresvariables<-list(lhx~1,lhy~1,lhz~1,rhx~1,rhy~1,rhz~1,hx~1,hy~1,hz~1,sx~1,sy~1,sz~1,
8                        lwx~1,lwy~1,lwz~1,rwx~1,rwy~1,rwz~1)
9 listaGau<-list(gaussian(),gaussian(),gaussian(),gaussian(),gaussian(),
10              gaussian(),gaussian(),gaussian(),gaussian(),gaussian(),
11              gaussian(),gaussian(),gaussian(),gaussian(),gaussian(),
12              gaussian(),gaussian(),gaussian())
13 posiciones<-c(155,163,10,88,90,179,81,142,102,156,14,96,11,82,35,78,131,65,69)
14 SN<-c[1:18]
15
16 tryCatch({
17   mod<- depmix(nombresvariables[SB],
18              data = datos, transition = ~datos$phase, nstates = 2,
19              family= listaGau[doce],
20              ntimes = posiciones)
21
22   set.seed(3)
23
24   fm1<-fit(mod)
25   modNew<-depmix(nombresvariables[SB],data = df,transition = ~df$timestamp,
26                 nstates = 2, family = listaGau[SB])
27
28   modNew<-setpars(modNew,getpars(fm1))
29
30   resHMM<-viterbi(modNew)
31   resultados[x]<-sum(resHMM[,1]==dff,na.rm = TRUE)
32 },error=function(e){cat("error",conditionMessage(e),"\n")}
33 )
34 }
35
36 write.csv(resultados,file = "resultados.csv",row.names = TRUE,col.names = TRUE)
37
```

En las líneas 3 y 4 se leen los datos crono-valorados. En las 5 y 6 se obtienen los datos útiles. En la línea 7 se declara una lista con el nombre de las variables en el *dataset*. En la línea 9 se declara una lista con la forma de distribución de los datos que se emplean en el ejercicio, por variable. En la línea 13 se declara la variable que contiene las particiones de cada clase, es decir los renglones en donde la clase cambia. En la línea 14 se declara la variable que contiene cuáles variables se estarán clasificando, en este ejemplo se clasificarán 18. Línea 17 declara la variable que contendrá el modelo de entrenamiento, en este caso se están contemplando dos clases (*nstates*). Línea 20 se refiere a las posiciones en donde se dividen los datos por clase. La línea 22 es una función propia de la librería que optimiza los parámetros. Línea 25 crea el modelo de la HMM para los datos de prueba. Línea 28 se utiliza para ordenar los parámetros de los dos modelos en uno solo. La línea 30 utiliza el algoritmo *viterbi* para clasificar los modelos obtenidos por la HMM. La línea 31 compara las predicciones con el *ground truth*.

Anexo B

Código de la implementación de la librería G1DBN (S Lèbre, 2013) para *Rstudio* que se utilizó en esta investigación. La implementación de la librería se utilizó para la obtención de las interconexiones mencionadas en la sección 3.1.4 y cuya definición se encuentra en la sección de acrónimos.

```
1 library(G1DBN)
2
3
4 a1_raw_N <- read.csv("a1_raw_N.csv")
5 datosDBN<-a1_raw_N[,c(1:18,21)]
6
7 S1m<-DBNScoreStep1(datosDBN, method = 'ls')
8 G1m<-BuildEdges(S1m$S1ls,threshold = 0.0000000001,dec = FALSE)
9
10
11 S2m<-DBNScoreStep2(S1m$S1ls,data = datosDBN,method = 'ls',alpha1=0.5)
12 G2<-BuildEdges(S2m,threshold = 0.0000000001,dec = FALSE)
13 write.csv(G2,file = "as_2regresion5clases0000000001.csv")
14
15
```

Para la implementación de la librería G1DBN se busca obtener las dependencias más significativas. Para ello, en línea 4 se lee el *dataset*. En la línea 5 se declara la variable que contiene los datos útiles. En la línea 7 se utiliza la función `DBNScoreStep1` para obtener la matriz con las calificaciones de primer orden hechas en los datos útiles utilizando el método *Least Square* como estimador. En la línea 8 se obtienen las dependencias más significativas. En la línea 11, la función `DBNScoreStep2` infiere la calificación de cada arco en la RBD describiendo las dependencias completas entre variables. Para ello, la función `DBNScoreStep2`, utiliza los datos del *dataset*, las dependencias de primer orden (matriz obtenida con la función `DBNScoreStep1`) y el umbral *alpha* 1 para la selección de arcos en la matriz `S1m`. En la línea 12 se aplica un umbral para obtener las dependencias más significativas. En la línea 13 se obtienen los resultados que serán utilizados por la función que crea la sábana dinámica de acuerdo a la sección 3.2.3.

Anexo C

Definiciones, abreviaturas y símbolos

Definiciones

Dataset

Un *dataset* es una colección de datos que se utiliza para algún propósito específico de aprendizaje automático. Un conjunto de entrenamiento es un conjunto de datos que se utiliza como entrada a un sistema de aprendizaje, que lo analiza para aprender un modelo. Un conjunto de prueba o conjunto de evaluación es un conjunto de datos que contiene datos que se utilizan para evaluar el modelo aprendido por un sistema de aprendizaje. Un conjunto de entrenamiento se puede dividir en un conjunto de cultivo y un conjunto de poda. Cuando el conjunto de entrenamiento y el conjunto de prueba contienen conjuntos de datos separados, el conjunto de prueba se conoce como conjunto reservado (Sammut & Webb, 2010).

Interconexiones

Assumiendo que la RBD es invariante al tiempo, se puede considerar que el modelo está definido por dos estructuras: i) la estructura base, y ii) la estructura de transición. Entonces, se divide la construcción (*learning*) de una RBD en dos partes. Primero se aprende la estructura base y luego, dada la estructura base, se aprende la estructura de transición (Koller & Friedman, 2009; Sucar, 2006).

Para construir la estructura base de una RBD se puede utilizar toda la información disponible, ignorando la información temporal (o se pueden utilizar diferentes métodos como los explicados en el capítulo 2 de este documento). Esto es equivalente a aprender una RB. Para aprender la red de transición se considera la información temporal, en particular los datos de las variables en dos porciones de tiempo consecutivas: X_t y X_{t+1} . Considerando la estructura base, se pueden aprender las dependencias entre las variables t y $t+1$ (Friedman et al., 1998).

En la figura C.1 se observan los arcos de la estructura base y las dependencias entre las variables t y $t+1$. Los arcos que se obtienen en la estructura base se conocen como conexiones *intra*, y los que van de una porción de tiempo a otra son llamadas conexiones *inter*, en este trabajo de investigación son referidas como *interconexiones*.

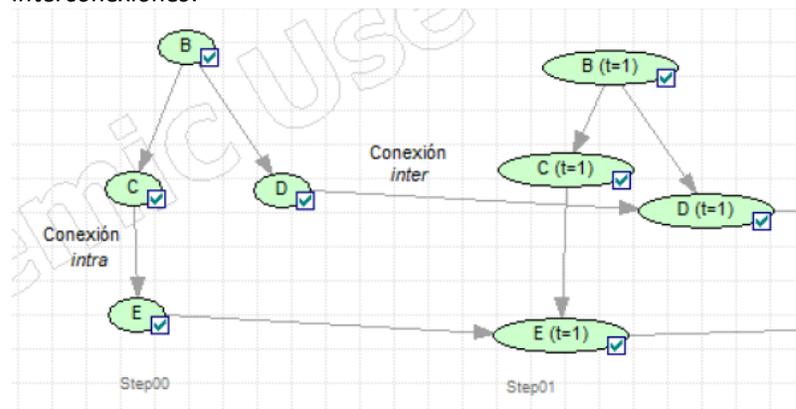


Figura C.1. RBD simple que muestra la diferencia entre los arcos de diferentes porciones de tiempo.

Filas básicas	<p>Una fila t es básica, si y solo si, en la matriz de diferencias no existe alguna fila p que sea sub fila de t. Si p y t son dos filas de la matriz de diferencias, p es sub fila de t, si y sólo si:</p> <ol style="list-style-type: none"> I. $\forall j(a_{pj} = 1 \Rightarrow a_{tj} = 1)$ (en todas las columnas donde p tiene 1, t también) II. $\exists k(a_{tk} = 1 \wedge a_{pk} = 0)$ (existe al menos una columna en la que t tiene un 1 y p no lo tiene. También se dice que t es super fila de p). <p>Ejemplos:</p> <ul style="list-style-type: none"> • Sea $t=(11001)$ y $p=(10001)$, entonces p es sub fila de t y de igual modo t es super fila de p; se puede observar que en cualquiera de las columnas en las que p tiene un 1 la fila t también lo tiene y existe al menos una columna en la que t tiene un 1 donde p no lo tiene: $\begin{array}{r} T \ 1 \ 1 \ 0 \ 1 \ 1 \\ P \ 1 \ 0 \ 0 \ 0 \ 1 \end{array}$ • Sea $t=(10101)$ y $p=(11011)$, ninguna es subfila de la otra. Existen columnas donde en una fila hay un 1 y en la otra un 0 y esa condición la cumple tanto t como p. $\begin{array}{r} T \ 1 \ 0 \ 1 \ 0 \ 1 \\ P \ 1 \ 1 \ 0 \ 1 \ 1 \end{array}$
Testores	<p>Un testor es un conjunto de columnas (atributos, variables, rasgos, etc) que permite diferenciar entre dos clases, porque ningún objeto de la clase T_0 se confunde con objeto alguno de la clase T_1.</p> <p>El conjunto $\tau = \{i_1, \dots, i_s\}$ de columnas de la tabla T, con sus respectivos rasgos x_{i_1}, \dots, x_{i_s}, se denomina testor para $(T_0, T_1) = T$, si después de eliminar de T todas las columna excepto τ, no existe fila alguna en T_0 igual a una de T_1. Al subíndice s se le denomina longitud del testor.</p>
Testores Típicos	<p>Dado un testor, se le podría quitar una columna y corroborar si sigue siendo testor, hasta que se llegue a un testor especial: testor típico; que al suprimírsele cualquiera de sus columnas o rasgos, confunde algún objeto de T_0 con algún otro de T_1 (Ruiz-Shulcloper et al., 1999).</p>
Algoritmo <i>Bottom-Top</i>	<p>La implementación de este algoritmo al presente trabajo se deriva de la matriz de diferencias y la matriz básica (explicadas en la sección 2.3.6), para aplicar a esta última el algoritmo BT, tomando en cuenta las posiciones del algoritmo de escala exterior.</p> <p>El algoritmo BT se basa en la idea de ir generando n-uplos booléanos a partir del $(0, \dots, 0, 1)$, el cual corresponde con el conjunto $\{x_n\}$, hasta llegar al $(1, 1, \dots, 1, 1)$, correspondiente al conjunto total de rasgos R. En cada caso, el algoritmo verifica si el conjunto de columnas que se corresponden con las coordenadas unitarias del n-uplo generado es un testor. Para pasar de un n-uplo a otro y con vistas a producir saltos eficientes en el orden de generación de n-uplos α, se utilizan:</p> <ol style="list-style-type: none"> Paso 1. Se genera la primera lista testor α no nula de longitud n. Paso 2. Se determina si la lista generada es una lista testor en la MB. Paso 3. Si es lista testor, se verifica si es lista testor típico, se almacena en α. Si no es lista testor, se determina la fila de la MB que provoca este hecho (de no ser la única se toma la que tenga el ultimo 1 más a la izquierda).

Paso 4. Se genera la lista siguiente a las descartadas en virtud del paso 3 y se regresa al paso 2, en caso de que la lista resulte del paso 3 no sea posterior a (1,1,...,1,1). De lo contrario termina el algoritmo.

Algoritmo de escala exterior Para el cálculo de todos los testores típicos se utilizará un algoritmo de escala exterior. Para la descripción de dicho algoritmo se usaran los n-uplos característicos $\alpha = (\alpha_1, \dots, \alpha_n)$ de cada uno de los posibles subconjuntos de rasgos, es decir, si $\alpha_j = 0$, significará que la columna $j(x_j)$ no aparece en el conjunto de columnas (rasgos) a considerar. Por orden natural de los n-uplos α se entenderá el orden creciente generado por los números naturales en su notación binaria (Ruiz-Shulcloper et al., 1999).

Accuracy Esta medida de desempeño/rendimiento se llama precisión o exactitud en español, sin embargo, se usará la palabra en Ingles para no confundirla con la medida de rendimiento *Precision*. Esta medida es la proporción de las muestras clasificadas correctamente; así, se utiliza a menudo en tareas de clasificación (Géron, 2019). En general, ¿con qué frecuencia es correcto el clasificador? (Markham, 2014).
 $Accuracy = (TP + TN) / total_muestras$
Para esta investigación la medida *accuracy* se expresará en un rango de 0 a 1.

Sensitivity Es la métrica que evalúa la habilidad del modelo al predecir los *TP* de cada categoría (clase) disponible. En otras palabras: cuando en realidad es sí, ¿con qué frecuencia predice que sí? (Markham, 2014)
 $Sensitivity = TP / sí\ real$
También conocido como *True Positive Rate*.

Precision Cuando predice sí, ¿con qué frecuencia es correcto? (Markham, 2014).
 $Precision = TP / predijo\ sí$

Algoritmos de construcción de redes bayesianas La tarea de adaptar una red bayesiana se suele denominar aprendizaje, un término tomado de la teoría de sistemas expertos y la inteligencia artificial (Koller & Friedman, 2009). Se realiza en dos pasos diferentes, que corresponden a las técnicas de selección de modelos y estimación de parámetros en modelos estadísticos clásicos.

El primer paso se denomina aprendizaje de estructuras y consiste en identificar la estructura gráfica de la red bayesiana. Idealmente, debería ser el mapa *I*-mínimo de la estructura de dependencia de los datos o, en su defecto, al menos debería resultar en una distribución lo más cercana posible a la correcta en el espacio de probabilidad. Se han propuesto varios algoritmos en la literatura para el aprendizaje de estructuras. A pesar de la variedad de antecedentes teóricos y terminología, se clasifican en tres categorías amplias: algoritmos basados en restricciones, basados en puntajes e híbridos. Como alternativa, la estructura de la red se puede construir manualmente a partir del conocimiento de dominio de un experto humano y la información previa disponible sobre los datos.

El segundo paso se llama aprendizaje de parámetros. Como su nombre indica, implementa la estimación de los parámetros de la distribución global. Esta tarea se puede realizar de manera eficiente estimando los parámetros de las distribuciones locales que implica la estructura obtenida en el paso anterior.

Los algoritmos de aprendizaje de estructuras basados en restricciones se basan en el trabajo fundamental de Pearl en mapas y su aplicación a modelos gráficos causales. Su algoritmo de causalidad inductiva (CI) (Verma, Pearl, & others, 1991) proporciona un marco para aprender la estructura de las redes bayesianas utilizando pruebas de independencia condicional.

Algunos algoritmos de aprendizaje Bayesiano basados en restricciones son:

- PC: la primera aplicación práctica del algoritmo IC (Spirites, Glymour, Scheines, & Heckerman, 2000) un procedimiento de selección hacia atrás del gráfico saturado
- Grow-Shrink (GS): basado en el algoritmo de sábana de Markov Grow-Shrink (Dimitris Margaritis, 2003), un enfoque de detección de sábana de Markov de selección directa simple
- Asociación incremental (IAMB): basado en el algoritmo de sábana de Markov de asociación incremental (Tsamardinos, Aliferis, Statnikov, & Statnikov, 2003), un esquema de selección de dos fases basado en una selección hacia adelante seguida de una hacia atrás
- Asociación Incremental Rápida (Fast-IAMB): una variante de IAMB que utiliza la selección especulativa por pasos hacia adelante para reducir el número de pruebas de independencia condicional (Yaramakala & Margaritis, 2005)
- Asociación incremental intercalada (Inter-IAMB): otra variante de IAMB que utiliza la selección progresiva hacia adelante (Tsamardinos et al., 2003) para evitar falsos positivos en la fase de detección de sábana de Markov

Los algoritmos de aprendizaje de estructuras basados en puntuaciones (también conocidos como algoritmos de búsqueda y puntuación) representan la aplicación de técnicas generales de optimización heurística al problema de aprender la estructura de una red bayesiana. A cada red candidata se le asigna una puntuación de red que refleja su bondad de ajuste, que luego el algoritmo intenta maximizar.

Algoritmos de construcción de redes bayesianas (continua...)

Algunos ejemplos de esta clase de algoritmos son los siguientes:

Algoritmos de búsqueda *greedy* como *hill-climbing* con reinicios aleatorios o búsqueda tabú (Bouckaert, 1995). Estos algoritmos exploran el espacio de búsqueda a partir de una estructura de red (generalmente el gráfico vacío) y agregan, eliminan o invierten un arco a la vez hasta que la puntuación ya no se puede mejorar.

Los algoritmos de aprendizaje de estructuras híbridas combinan algoritmos basados en restricciones y basados en puntajes para compensar sus debilidades y producir estructuras de red confiables en una amplia variedad de situaciones. Los dos miembros más conocidos de esta familia son el algoritmo *Sparse Candidate* (SC) de (Friedman et al., 1998) y el algoritmo *Max-Min Hill-Climbing* (MMHC) de (Tsamardinos, Brown, & Aliferis, 2006).

Outliers

Los valores atípicos (*Outliers*) son valores extremos que se desvían de otras observaciones sobre los datos, pueden indicar una variabilidad en una medición, errores experimentales o una novedad. En otras palabras, un valor atípico es una observación que difiere de un patrón general en una muestra (Santoyo, 2017).

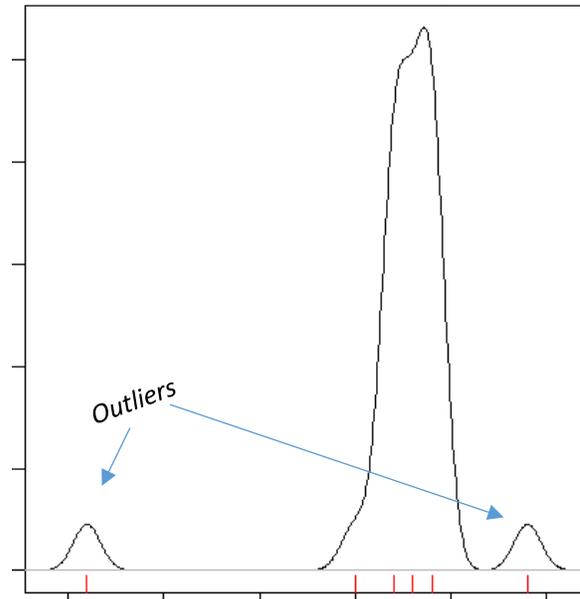


Figura C.2 Representación gráfica de los *outliers* (Santoyo, 2017)

Threshold

Umbral, valor máximo que se considera para tomar la decisión de si dos objetos son similares o no.

p-value

p-values son números entre 0 y 1 que cuantifican que tanta confianza se puede tener al interpretar una medida; en esta investigación cuantifica la confianza de que la medida *accuracy* obtenida en los experimentos (núm. 6-8, 10–12) sea suficientemente alta; la confianza tiene una relación con *p-value*: entre más pequeño el valor, más es la confianza en que los resultados no son producto del azar. En la práctica, el umbral utilizado es 0.05 para tener la suficiente confianza de que los resultados no son producto del azar (Di Leo & Sardanelli, 2020). Así, en esta investigación el *p-value* ayuda a decidir si se debería, o no, rechazar las Hipótesis Nulas presentadas en el capítulo uno. La fórmula para obtener el *p-value* en los experimentos es la siguiente:

$$p - value = p(accuracy) + p(accuracy mayor)$$

*Longest
Common
Substring (LCS)*

Dadas dos cadenas S y T , cada una de longitud como máximo n , el problema de LCS es encontrar la subcadena común más larga en S y T . Este es un problema clásico en ciencias computacionales con solución igual a $\mathcal{O}(n)$ – *tiempo* (Amir, Charalampopoulos, Pissis, & Radoszewski, 2020). El LCS compara dos valores de cadenas y prescribe si son iguales determinando la longitud más larga de caracteres (valores) que son comunes en ambas cadenas (subcadena), ya sea que la subcadena representa el total o una parte de los valores de la cadena (Oracle, 2012).

Se utiliza la comparación de subcadena común más larga (LCS) para encontrar coincidencias entre los valores de cadena donde puede haber 'ruido' al principio o al final de la cadena que es difícil de ignorar en una comparación mediante la eliminación de palabras, o cuando se sabe que es probable que los valores de cadena con una secuencia común de caracteres sobre una cierta longitud estén relacionados, por ejemplo, para coincidir con "Nomura Securities Co., Ltd." con "Nomura Investor Relations Co., Ltd." con una subcadena común más larga de 6 caracteres "Nomura" (Oracle, 2012).

La comparación de subcadena común más larga se usa a menudo en las reglas de coincidencia que se encuentran en la parte inferior de la tabla de decisiones para encontrar y revisar posibles coincidencias que tienen similitud pero que no han podido coincidir con otras reglas, tal vez debido a problemas de ordenación o debido a un exceso de 'ruido'.

Esta regla de coincidencia tiene flexibilidad a ediciones (Amir et al., 2020) así, en esta investigación, se propone incluir no solo el valor de cada cadena sino que también, una ventana de valores y un sentido de evolución

Entre las medidas de similitud, en especial LCS, existe la comparación llamada *Longest Common Substring Percentage*. La comparación de porcentaje de subcadena común más larga determina la similitud de dos valores entre sí al encontrar la subcadena común más larga entre dos valores y relacionar su longitud, en caracteres, con la longitud en caracteres del valor de entrada más largo o más corto (Oracle, 2012).

En términos matemáticos, la comparación de porcentaje de subcadena común más larga utiliza la siguiente fórmula para calcular sus resultados:

$$LCSP = \left(\frac{LCS}{M}\right) \times 100$$

donde:

LCSP = Porcentaje de la cadena común más larga

LCS = La cadena común más larga

M = la longitud máxima o mínima de las dos cadenas siendo comparadas

Así, las cadenas "ABC" y "ABCD": LCS = 3, M = 4 (si se utiliza la longitud de la cadena más larga), y por consecuencia LCSP = 75%. Si se utilizará la longitud de la cadena más corta, LCSP = 100%.

Se utiliza la comparación Porcentaje de subcadena común más larga (LCSP) donde la subcadena común más larga (LCS) puede no dar resultados tan precisos, porque simplemente puede coincidir con una palabra o palabras largas en un valor dado sin tener en cuenta otros datos en el valor. Por ejemplo, los valores "Arden Design Birmingham" y "Britannia Design Birmingham" tienen una subcadena común más larga de 17 caracteres, lo que indica un resultado sólido. Sin embargo, tienen un porcentaje de subcadena común más largo de solo el 63%, un resultado mucho más débil.

Método Media	<p>Se obtiene la media de cada objeto para construir la red inicial. La media es la media aritmética (promedio) y se calcula sumando un grupo de números y dividiendo a continuación por el recuento de dichos números (Office, 2021). La media es una medida de posición central y su fórmula matemática se expresa por,</p> $Media(X) = \bar{x} = \frac{\sum_{i=1}^N X_i}{N}$ <p>Siendo (X_1, X_2, \dots, X_N) el conjunto de observaciones.</p>
Método Derivada	<p>Se obtiene la derivada de cada objeto. Este método computa la diferencia entre par de elementos consecutivos en el objeto,</p> $der(X) = (x_2 - x_1, x_3 - x_2, \dots, x_n - x_{n-1})$ <p>donde X representa el objeto crono-valuado y x_i los elementos consecutivos de X. Se computa en forma recursiva hasta el orden de la diferencia: $n - 1$, siendo n la longitud del objeto (R Core Team, 2018).</p>

Abreviaturas

LCRP	Lógica combinatoria para el reconocimiento de patrones
RB	Red Bayesiana
RBD	Red Bayesiana Dinámica
TP +TN	True Positive + True Negative
SVD	Subconjunto de Variables Discriminantes
BT	Algoritmo Bottom Top
MD	Matriz de Diferencias
MB	Matriz básica
HMM	Máquina de aprendizaje basada en los modelos ocultos de Markov

Símbolos

α	Lista testor correspondiente a una fila de la MD
β	Pendiente correspondiente al análisis lineal de una serie
\forall_j	Para todos los valores en columna j
a_{pj}	Valor en la columna j fila p
\exists_k	Existe al menos en la columna k
shO	Objeto observado más corto
lO	Objeto observado más largo
d	Resultado de la decisión de similitud entre dos objetos
C	Identificador de la clase a la que pertenece el objeto
$RangeO_{i,r}$	Rango de comparación para el elemento r del objeto i
$SO_{i,r}$	Elemento r del objeto i escalado entre cero y uno
$TO_{i,r}$	Elemento r del objeto i transportado a cero
EvO_i	Serie evolutiva del objeto i

Anexo D

Resumen y evidencia de la publicación del capítulo *Feature Selection for Pattern Recognition: Upcoming Challenges* parte de la investigación realizada en este documento.

Chapter 4 Feature Selection for Pattern Recognition: Upcoming Challenges



Marilu Cervantes Salgado and Raúl Pinto Elías

Abstract Pattern recognition is not a new field, but the challenges are coming on the data format. Today's technological devices provide a huge amount of data with extensive detail evolving the classical pattern recognition approaches for dealing with them. Given the size of and quantity of descriptors data possess, traditional pattern recognition techniques have to draw on feature selection to handle problems like the excess of computer resources and dimensionality. Feature selection techniques are evolving, as well, for data related reasons. Chronologically linked data brings new challenges to the field. In the present chapter, we expose the gap in feature selection research to handle this type of data, as well as give suggestions of how to perform or pursue an approach to chronologically linked data feature selection.

Keywords Feature selection · Pattern recognition · Chronologically linked data

4.1 Introduction

Today, in the big data era, data are coming at large formats, from high-definition video to interaction posts on social media, making it hard for pattern recognition algorithms to process and make decisions with them. Pattern recognition aims to search into data for regularities that can automatically classify (among other tasks) the data into different classes.

Let's look at this last statement with a specific problem: one of the latest smartphone tackles unlocking function with facial recognition, since this task has to differentiate a face from non-face image, it has to search for facial characteristics, thus pattern recognitions is involved.

M. Cervantes Salgado · R. Pinto Elías (✉)
Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET),
Cuernavaca, Morelos, Mexico
e-mail: rpinto@cenidet.edu.mx

© Springer International Publishing AG, part of Springer Nature 2018
O. O. Vergara Villegas et al. (eds.), *Advanced Topics on
Computer Vision, Control and Robotics in Mechatronics*,
https://doi.org/10.1007/978-3-319-77770-2_4

77

Anexo E

Resumen y evidencia de la publicación del artículo *Function to Flatten Gesture Data for Specific Feature Selection Methods to Improve Classification*. Parte de la investigación realizada en esta tesis.



Traitement du Signal

Vol. 38, No. 4, August, 2021, pp. 929-935

Journal homepage: <http://ieta.org/journals/ts>

Function to Flatten Gesture Data for Specific Feature Selection Methods to Improve Classification



Marilú Cervantes Salgado*, Raúl Pinto Elías, Andrea Magadán Salazar

Department of Computer Science, National Center for Research and Technological Development, Cuernavaca 62490, Morelos, Mexico

Corresponding Author Email: marilu.cersa@cenidet.edu.mx

<https://doi.org/10.18280/ts.380402>

ABSTRACT

Received: 19 May 2021

Accepted: 25 July 2021

Keywords:

Bayesian networks, chronologically linked data, feature selection methods, gesture classification, Logical Combinatorial to Pattern Recognition, Markov blanket

Gestures are pieces of information with characteristics such as: multiple and chronologically linked samples with different length. The gesture characteristics mentioned before make classification, of this type of data, a challenging task. We studied the effects of flattening gesture data. We proposed a function to represent gestures in a flat format taking in consideration the evolution sense they possess. The function's main goal is to compare gestures intra class to spot differences. This function is described step by step and then its outcome is used as input to two feature selection methods (Bayesian network / Markov blanket and Logical Combinatorial to Pattern Recognition). After, with the subsets obtained, we trained Hidden Markov Models machines. We found that applying our methodology to gesture data, the subset of attributes obtained (feature selection) were able to classify with accuracies of 0.88 and 0.87 of a maximum of 0.90. The maximum accuracy was obtained from an exhaustive classification exercise we performed in order to compare our results. These findings suggest that our methodology can be applied over raw data (gesture data or