Instituto Politécnico Nacional

Centro de Investigación en Computación

Visualización de objetos multivariados utilizando agrupación de variables

T E S I S

Que para obtener el grado de:
Maestro en Ciencias de la Computación

P R E S E N T A :

Lic. Rodolfo Antonio Vilchis Mompala

Directores de tesis

Dr. Gilberto Lorenzo Martínez Luna
Dr. Adolfo Guzmán Arenas

2013

SIP-14 bis

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 12:30 horas del día 23 del mes de septiembre de 2013 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Visualización de objetos multivariados utilizando agrupación de variables"

Presentada por el alumno:

VILCHIS	**MOMPALA**	**RODOLFO ANTONIO**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

B	1	1	0	8	4	9

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

Dr. Gilberto Lorenzo Martínez Luna

Dr. Adolfo Guzmán Arenas

Dr. Serguei Pavlovich Levashkine

Dr. Alexander Gelbukh

Dr. Jesús Guillermo Figueroa Nazuno

Dr. Marco Antonio Moreno Ibarra

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas

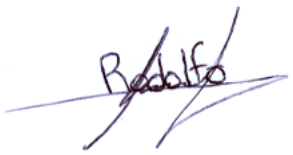
ESTADOS UNIDOS MEXICANOS
INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION EN COMPUTACION
DIRECCION

INSTITUTO POLITÉCNICO NACIONAL
SECRETARIA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el dia 4 del mes Noviembre del año 2013, el (la) que suscribe Vilchis Mompala Rodolfo Antonio alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro B110849, adscrito al Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Gilberto Lorenzo Martínez Luna y Dr. Adolfo Guzmán Arenas y cede los derechos del trabajo intitulado Visualización de objetos multivariados utilizando agrupación de variables, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, Delegación Gustavo A. Madero, C.P 07738, México D.F. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Rodolfo Antonio Vilchis Mompala

Resumen

La visualización de la información es una técnica muy usada para analizar las relaciones entre las variables de un conjunto de datos. Éstas pueden ser tanto numéricas como simbólicas y al generar una visualización la mayoría de las veces se muestran tres variables, y seis como máximo, usando color, forma y tamaño. Después de todo, el ojo humano puede percibir con claridad datos en papel o en pantalla con dos dimensiones (ejes), máximo tres. Sin embargo, en conjuntos multivariados es común tener más de cinco dimensiones, por lo cual al visualizarlos, el usuario no detecta cómo varían las variables a través de los datos, ni las relaciones entre estas. Para resolver este problema, se suelen usar varias gráficas y tablas. Esto da una visión fragmentada de cuáles objetos tienen qué valores en cuáles atributos. El propósito de este trabajo es transmitir la mayor cantidad de información posible presente en un conjunto de datos de tal forma que sea fácilmente comprensible por el ser humano, es decir, agilizar la detección de las relaciones entre las variables numéricas y simbólicas. El trabajo presenta un nuevo método para mostrar en una sola gráfica tantas variables como sea posible, de modo que el usuario tiene una visión más integral de los datos. El sistema desarrollado, automáticamente escoge el mayor número posible de variables a mostrar (dado unos parámetros) y las agrupa para que la comprensión se efectúe sobre un mayor número de variables. Para ello, se hace uso de la regresión lineal, utilizando dos métodos, los mínimos cuadrados y *Multivariate Adaptive Regression Splines* (MARS). La idea es encontrar comportamientos monótonos (crecientes y decrecientes) entre las variables, para poder graficarlas en un mismo eje cartesiano, cada variable con una escala diferente. Si hay variables constantes o casi constantes (varían muy poco) estas se muestran en la visualización con una etiqueta. Aquellas variables que no poseen un comportamiento monótono con otras se tratan de ajustar mediante un particionado (reduciendo la precisión) sobre alguno de los ejes cartesianos.

Para las variables simbólicas, se busca una partición de dos o tres conjuntos de tal forma que encajen (se particionen) sobre algún eje, lo que permitirá graficarlas. Si existen variables simbólicas sobrantes, es decir, no se ajustaron mediante un particionado, se seleccionan dos de ellas y se muestran mediante el color y forma, siempre y cuando cumplan con algunas restricciones.

Con las técnicas empleadas, un conjunto de 3194 registros con 52 variables fue po-

sible mostrarlo con nueve de sus 52 variables, otro conjunto de 4898 registros y 12 atributos se mostró con ocho de sus 12 atributos y otros conjuntos han mostrado buena visualización. En general, ambos métodos dan buenos resultados, bajo ciertas condiciones es mejor usar mínimos cuadrados y en otras MARS. Para las variables simbólicas en algunos casos se logró encontrar una partición dando buenos resultados en la visualización.

Palabras clave: Visualización de la información; mínimos cuadrados; minería de datos; *Multivariate Adaptive Regression Splines*.

Clasificación ACM:
H. Information Systems / H.5 Information Interfaces and Presentation / H.5.2 User Interfaces

Abstract

Information visualization is a very useful technique to analyze the relationship between the variables of a data set. Each object in the data set can have numeric and symbolic attributes. When a multivariate data set is visualised only three attributes (variables) or at most six attributes are displayed using colors, shapes and sizes. This is because the human eye can only perceive with ease limited 2D or 3D data in paper or on screen. Nevertheless, for multivariate objects, it is common to have more than five variables and the significance or the relationships among the variables are lost in translation when observed separately.

The purpose of this work is to identify and present what is considered less complex relationships between some of the variables in a data set in such manner that it is easily understood by the user, and to facilitate the detection of the relationship among numeric and symbolic (qualitative) variables. This work presents a new method to display, in a single graph, as many variables as possible so that the user has a more holistic view of the data. The system developed automatically chooses the maximum number of variables to show (given some parameters) and groups the variables that behave similarly. For this, it uses linear regression, following two methods, the least squares (LS) and Multivariate Adaptive Regression Splines (MARS). The basic principle is to find monotonic behaviors (increasing and decreasing) among the variables to graph them on the same Cartesian axis, each variable with a different scale. Variables that are constant or almost constant are shown in the visualization with a label. Variables that do not have a monotonic behavior with others will be adjusted by partitioning (reducing accuracy) to any of the Cartesian axes (if possible).

Symbolic variables are searched to find an order of values which would be generated by a numeric variable that would result in a partition of the symbolic variable values on the numeric variable in order to graph it. Symbolic variables that have not an order with any numeric variable are displayed using colors and shapes, provided they comply with certain restrictions.

Tests were performed on ten data sets, one with synthetic data and the rest with real data. For a set of 3194 records (objects) with 52 variables, it was possible to display nine of its 52 variables in a single graph. For another data set of 150 objects and five attributes, it was possible to display all five attributes in a single graph.

Other data sets have shown good visualization. In general, both methods give good results, under certain conditions is better to use least squares and other MARS. For symbolic variables it was possible in some cases to find a partition giving good results in the visualization.

Keywords: Information visualization; least squares; data mining; Multivariate Adaptive Regression Splines.

ACM Classification:
H. Information Systems / H.5 Information Interfaces and Presentation / H.5.2 User Interfaces

Agradecimientos

Esta tesis está dedicada a todas aquellas personas que participaron directa o indirectamente; leyendo, opinando, brindándome paciencia y ánimos. Sin ellos, este trabajo no habría sido posible.

En primer lugar, a mis padres, Guadalupe y Jesús, a quienes agradezco de todo corazón por su apoyo, cariño y comprensión.

Al resto de mi familia, muchas gracias por todo el apoyo brindado a lo largo de esta travesía, en especial al Dr. Pablo Galaviz.

A Gabriela Mayén por el apoyo que me ha dado durante estos años juntos y a su familia, con mucho cariño.

Agradezco a mis directores de tesis, Dr. Gilberto Lorenzo Martínez Luna y Dr. Adolfo Guzmán Arenas por haber confiado en mí, por su paciencia y tiempo tomado para la realización de esta tesis.

A los doctores Serguei Pavlovich Lavashkine, Alexander Gelbukh, Jesús Guillermo Figueroa Nazuno y Marco Antonio Moreno Ibarra quienes formaron parte de mi comité tutorial y jurado, gracias por todo su apoyo.

A mis amigos y compañeros, en especial a Yazmín y Carlos, gracias por su amistad.

A todos mis profesores, quienes compartieron su conocimiento para que esto pudiera ser posible.

Gracias a todos.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Actualmente la velocidad de crecimiento y el volumen almacenado de datos hace prácticamente imposible para los analistas de datos poder extraer conclusiones, tendencias o patrones a partir de los datos simples [3] cuando éstos son muy numerosos. Es por ello que surge la necesidad de buscar alternativas que permitan poder obtener ágilmente esta información. Una de las alternativas es la minería de datos. Otra alternativa es la visualización de la información, objeto de esta tesis.

El principal objetivo de la visualización es la representación perceptual adecuada tanto de los datos con parámetros múltiples como de las tendencias y relaciones que existen entre ellos. Su propósito es la asimilación rápida de información o monitoreo de grandes cantidades de datos [4].

En la actualidad el número de herramientas de visualización está creciendo debido a que es un área de investigación activa.

El estudio realizado por Larkin y Simón [5] muestra la utilidad de las técnicas de visualización de la información. Expone que la representación visual ayuda a las personas a la utilización de complejas inferencias perceptuales ya implementadas en su corteza visual. Sin embargo, para que la búsqueda del conocimiento mediante la visualización sea adecuada, la herramienta debe tener un diseño correcto que permita mejorar las posibilidades de detección de patrones, reducir la búsqueda de información, entre otros [4].

Un ejemplo de los problemas que se tiene a la hora de visualizar datos los podemos encontrar en un estudio realizado a una base de datos con *tweets* donde se mencionan a los candidatos a la presidencia de México para las elecciones de 2012. Dicho estudio tiene como finalidad el identificar el sentimiento de cada *tweet* y determinar si dicho sentimiento es positivo o negativo hacia el candidato[1]. Cabe mencionar que el estudio únicamente utiliza herramientas de base de datos y dicho procedimiento es sistemático.

Una vez teniendo la información procesada se prosiguió a generar las redes. Cada red constaba de la información recabada por semana de los *tweets* y que corresponden a un mismo sentimiento, es decir, cada red representaba *tweets* positivos o negativos.

El primer problema que surgió al generar dichas redes es la cantidad de *tweets*, debido a que el programa usado (Redes2005) únicamente soporta 250 *tweets*, por lo que en algunos casos hubo que tomar una muestra del total de estos.

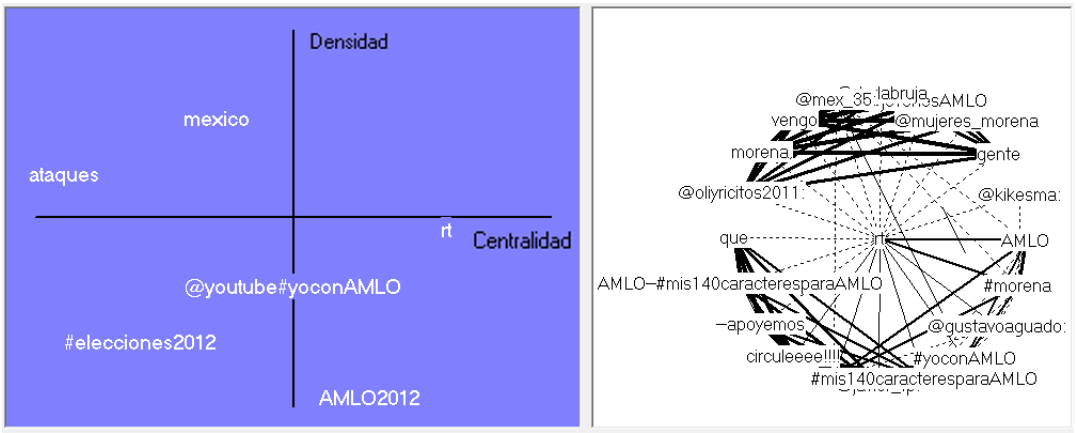


Figura 1.1: Red saturada con nodos encimados que dificulta la obtención de información. Del lado izquierdo se muestra el diagrama estratégico [1]. Del lado derecho se muestra la red que relaciona los términos. Línea más gruesa indica una relación más fuerte entre esos términos. Línea punteada una relación débil. Una relación fuerte significa que dos términos aparecen con mayor frecuencia y una relación débil que aparecen con mucha menor frecuencia que en la relación fuerte. Todo esto gira en base a un centro de interés, en este caso es la palabra "rt".

El segundo problema era determinar los parámetros correctos de tal forma que el grafo resultante no estuviera saturado y permitiera recabar información. En la

[1]Al momento de escribir esta tesis, el artículo se encuentra en desarrollo y se titula "Análisis de sentimientos por medio de funciones de bases de datos" escrito por el Dr. Gilberto Lorenzo Martínez Luna

figura 1.1 se observa un grafo que está saturado con nodos encimados y que es difícil el obtener información[2]. Sin embargo, modificando parámetros que permitan discriminar algunos nodos se genera una nueva red donde es más fácil el obtener información como se observa en la figura 1.2.

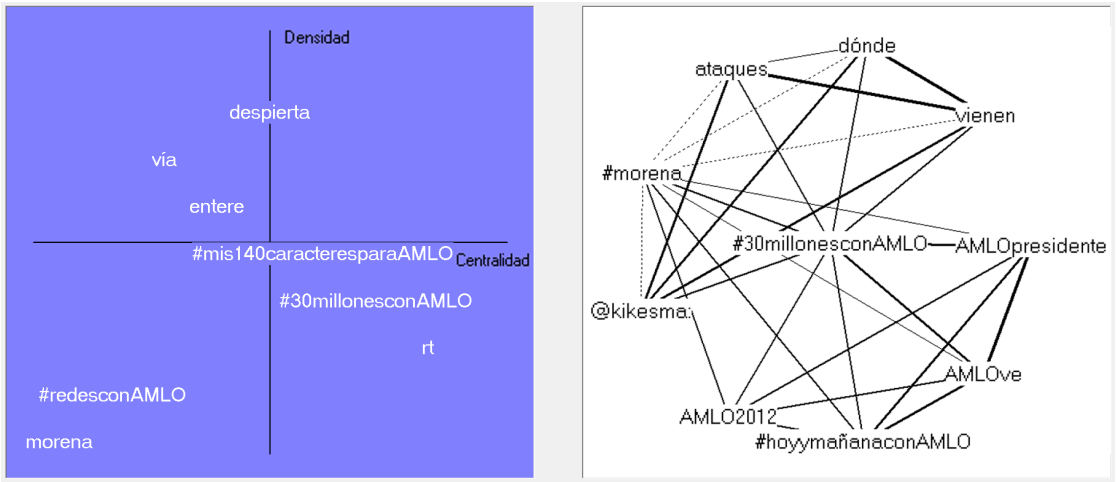


Figura 1.2: Red no saturada. Generada con la misma información que la red anterior pero distintos parámetros.

El estudio anterior muestra que un usuario a menudo obtiene resultados satisfactorios (gráficas fáciles de entender) solo después de ensayar lenta y manualmente con varios parámetros hasta lograr su objetivo, o cansarse y rendirse. En la figura 1.3 se observan cuatro gráficas, cada una con diferente combinación de variables de un conjunto de datos. Para generar estas gráficas el usuario debe seleccionar las variables a mostrar y en que ejes las quiere lo cual puede ser tedioso si el conjunto de datos tiene muchas variables o incluso el usuario puede no saber exactamente que variables graficar, lo cual resulta en pruebas de ensayo y error hasta obtener la información deseada o cansarse. Por eso una de las características construidas en el software desarrollado es la selección automática de variables a mostrar, para reducir la sintonización manual (basada en la experiencia del analista) de parámetros y variables.

La organización de este trabajo está dada de la siguiente manera:

- En lo que resta de este capítulo se definirá el problema a resolver, su justifica-

[2]En esta imagen el cuadrante izquierdo es llamado diagrama estratégico y el cuadrante derecho mapa estratégico o red. Para más información [1]

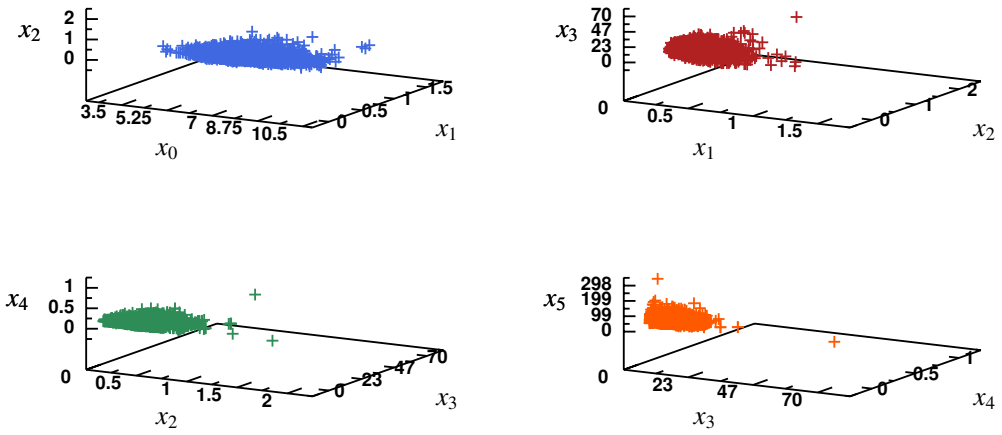


Figura 1.3: Múltiples gráficas del mismo conjunto de datos pero diferente combinación de variables. Este conjunto de datos tiene 11 variables numéricas, por lo que hay 165 combinaciones posibles (sin contar el orden) y que puede ser tedioso para el usuario generar tantas gráficas.

ción y el producto a entregar así como los objetivos, tanto particulares como generales y se darán algunas definiciones.

- En el capítulo 2 se detalla todo el marco teórico y estado del arte, definiendo lo que es visualización de la información así como la minería de datos, se dará un panorama general de todo esto así como algunos antecedentes y trabajos relevantes sobre visualización del conocimiento.

- En el capítulo 3 se analiza y diseña un método exploratorio que busca reducir el espacio de análisis a un especialista y mostrar una visualización que agilice la localización de información interesante[3], lo cual sirve de base para el desarrollo de un software que implanta este método.

- En el capítulo 4 se menciona la forma de implantar la aplicación, se describe la base de datos y la arquitectura que tendrá el sistema.

- En el capítulo 5 se efectúan pruebas al sistema para verificar que resuelven los problemas planteados.

[3]De aquí en adelante, se considera información interesante a aquellas relaciones existentes entre las variables, estas relaciones son los comportamientos monótonos entre las variables numéricas y las particiones de los valores de las variables, tanto numéricas como simbólicas.

- El capítulo 6 está dedicado a las conclusiones y a dar un panorama general de los posibles trabajos futuros que podrían desarrollarse.

1.1. Definiciones

- Objetos multivariados: Es un conjunto de datos con muchos atributos o variables (más de tres) que pueden ser numéricas o simbólicas.
- Dimensión: Son las propiedades, atributos o variables de los objetos.
- Variable simbólica: Variable cuyos valores no son numéricos. Por ejemplo, la variable color o la variable profesión. También llamada variable cualitativa, variable no numérica, variable categórica.

1.2. Planteamiento del problema

Como se ha mencionado, la gran cantidad de datos así como las relaciones entre ellos, dificulta en gran medida el análisis de la información. Es por ello que se requieren de herramientas que permitan un análisis ágil de estos datos. Una manera de lograr eso, es mediante el uso de grafos. Sin embargo, surgen dos preguntas:

1. ¿Cuál es la mejor forma de organizar las variables para ser desplegadas en los ejes cartesianos?
2. ¿Cómo determinar cuál es la mejor manera de visualizar un conjunto de datos?

La respuesta no es nada trivial. En la tesis doctoral de Charles Kemp [6] se da una posible respuesta a la pregunta (2). Para lograrlo, primero define un conjunto de seis formas base y se calcula un valor para cada una de ellas el cual indica la probabilidad de que esa forma sea la adecuada para esos datos. Al final, se ordenan las probabilidades y el valor más alto representará la mejor forma para ese conjunto de datos. Al ser un método probabilístico, puede haber errores, es decir, que la forma descubierta no sea la indicada para los datos. Sin embargo, el autor comenta que en caso de haber errores (la forma resultante no es la mejor que representa los datos),

la forma descubierta es muy similar a la forma que mejor representa al conjunto de datos.

A diferencia de Kemp, nuestro enfoque es desplegar solamente datos en un espacio cartesiano tridimensional, pero estudiando con cuidado qué variables desplegar con cuáles otras y a cuáles asignarles color y forma, es decir, se propone una posible respuesta a la pregunta (1). Para ello se plantea resolver tres problemas principalmente:

- Encontrar la mejor manera de organizar las variables de un conjunto de datos en base a ciertos comportamientos y características.
- El despliegue simultáneo de variables tanto numéricas como cualitativas (simbólicas o no numéricas)
- Obtener una visualización que permita la comprensión fácil de las interdependencias existentes en las variables contenidas en los datos, y cómo se aglomeran éstos en qué regiones, de ser el caso.

1.3. Objetivos

1.3.1. Objetivos generales

- Determinar agrupaciones de las variables tanto numéricas como simbólicas en un espacio tridimensional de un conjunto de datos multivariados.
- Generar la gráfica resultante y verificar que el usuario entienda el o los fenómenos que ocurren.

1.3.2. Objetivos particulares

- Generar conjuntos de datos sintéticos cuya estructuración se conoce para probar el sistema.

- Verificar si en conjuntos de datos reales que de alguna manera se intuye su estructura, el algoritmo agrupa las variables de la forma esperada o en una forma aceptable (fácil de entender).
- Usar técnicas sólidas basadas en principios matemáticos y en la percepción visual humana, a fin de probar el algoritmo por medio de un software que pueda desplegar con éxito una variedad de conjuntos de datos reales y no solo sea útil para unos cuantos casos de prueba escogidos ad hoc.
- Mostrar de manera gráfica las relaciones encontradas entre las variables.

1.4. Justificación

La gran cantidad de información así como las relaciones entre ellos, dificulta en gran medida el análisis de la información, en particular el descubrimiento del conocimiento. Actualmente para lograr esto, se hace uso de herramientas OLAP[4], de visualización de la información, algoritmos de minería de datos [8], entre otras.

Hasta este momento, no hay un método que permita disminuir el espacio de búsqueda de información interesante y que se muestre en un espacio tridimensional.

Por lo anterior, se ve la necesidad de desarrollar un método que disminuya el espacio de búsqueda de las variables interesantes, es decir, variables con comportamientos monótonos o cuyos valores se particionan respecto a otra variable, ya sea numérica o simbólica.

1.5. Productos a entregar

Este trabajo tiene como finalidad diseñar un método que permita determinar la mejor manera de organizar (agrupar) las variables de un conjunto de datos, el cual es la base para la construcción de un software (prototipo) que muestra de forma

[4] Acrónimo en inglés de procesamiento analítico en línea (On-Line Analytical Proccessing). Es un sistema interactivo que permite a un analista ver diferentes resúmenes de datos multidimensionales [7]

gráfica al usuario estas agrupaciones en un espacio tridimensional. De esta tesis se desprende un artículo titulado "*Visualization of the largest number of properties of multivariate objects using meta-dimensional grouping*" el cual al momento de imprimir este trabajo se encuentra en fase de evaluación por parte de la revista "Information Visualization".

Capítulo 2

Marco Teórico y estado del arte

Los avances tecnológicos de hoy en día, en particular con los referentes al despliegue de imágenes en 2D o 3D, permiten el uso de las capacidades visuales de los humanos para obtener conocimiento mediante imágenes. Si la información a visualizar posee relaciones entre sus elementos, los datos pueden ser representados mediante nodos y dichas relaciones mediante aristas [9]. Esto hace pensar, que la información puede almacenarse en una base de datos de tipo relacional para posteriormente visualizarse como un grafo.

Por esta razón, es necesario aclarar algunos conceptos de bases de datos y minería de datos así como de la visualización de la información, como son su historia, sus objetivos entre otros.

2.1. Minería de Datos

En [10] se define el concepto de minería de datos como "el descubrimiento eficiente de información valiosa, no-obvia de una gran colección de datos", cuyo objetivo "es ayudar a buscar situaciones interesantes con los criterios correctos, complementar una labor que hasta ahora se ha considerado "intelectual" y de alto nivel, privativa de los gerentes, planificadores y administradores. Además, de realizar la búsqueda fuera de horas pico, usando tiempos de máquina excedentes" [11].

Lo anterior se consigue mediante programas llamados "mineros" (más información de estos programas [11]) cuyos algoritmos buscan tendencias, anomalías, desviaciones o situaciones interesantes que son desconocidas [11, 12]. Esto ayuda al gerente o directivo a tomar decisiones que mejoren el rumbo de la empresa o institución a su mando.

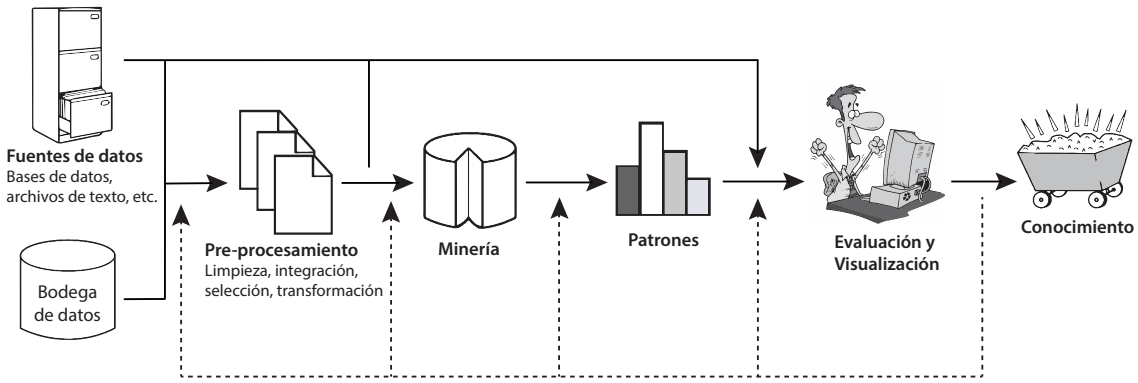


Figura 2.1: Proceso de la minería de datos para el descubrimiento de conocimiento. Las líneas punteadas indican la posibilidad de retroalimentar nuevamente el proceso. La etapa de conocimiento es el resultado de todo el proceso donde el usuario deberá tomar acciones según el nuevo conocimiento adquirido.

El proceso de la minería de datos es un ciclo ya que los resultados pueden retroalimentar nuevamente dicho proceso como se observa en la figura 2.1. Consta de los siguientes pasos [8]:

1. **Limpieza:** Remover ruido e inconsistencias en los datos.
2. **Integración:** Las fuentes de datos múltiples se pueden combinar.
3. **Selección:** Los datos relevantes para el análisis son obtenidos de la base de datos.
4. **Transformación:** Los datos se transforman en formas apropiadas para la minería.
5. **Minería:** Se aplican métodos para la extracción de patrones.
6. **Evaluación de patrones:** Identificar si los patrones obtenidos representan conocimiento basado en algunas medidas de interés.

7. **Presentación del conocimiento:** Usar técnicas de visualización de la información para presentar el conocimiento al usuario.

Los pasos 1 al 4 son llamados pasos de pre-procesamiento para preparar los datos para la minería.

2.1.1. Patrones frecuentes en Minería de Datos

Los patrones frecuentes son patrones (implícitos, no triviales) que aparecen frecuentemente en un conjunto de datos. Estos patrones pueden ser conjuntos de elementos, subsecuencias o subestructuras. Por ejemplo, si leche y pan aparecen frecuentemente juntos en un conjunto de compras entonces la pareja (leche, pan) forma un patrón frecuente. Una subsecuencia, tales como "comprar primero una computadora, luego una cámara de video y por ultimo una tarjeta de memoria", si esto ocurre frecuentemente en un historial de compras en una base de datos entonces se dice que eso forma un patrón secuencial. Una subestructura puede referirse a diferentes estructuras tales como un subgrafo, subárbol o una *sublattice* que pueden ser combinados con un conjunto de elementos o subsecuencias. Así, si una subestructura aparece frecuentemente entonces se dice que se ha encontrado un patrón estructurado [8, 13].

Uno de los primeros análisis en minería de datos fue el realizado por Agrawal en 1993 [14]. Él analizó una "cesta de compras" (figura 2.2) para ver los hábitos de los clientes a la hora de hacer las compras, es decir, buscaba asociaciones entre los diferentes elementos que los clientes ponían en la "cesta de compra". Por ejemplo, si los clientes colocaban leche se preguntaba ¿Cuál era la probabilidad de que también comprarán cereal? Esta información era de utilidad para los minoristas al organizar los productos que frecuentemente se compraban juntos en las estanterías de tal manera que sus ventas aumentarán.

2.2. Visualización de la información

La visualización de la información como rama de estudio de las ciencias de la computación es relativamente nueva. Sin embargo, antes de la llegada de esta área

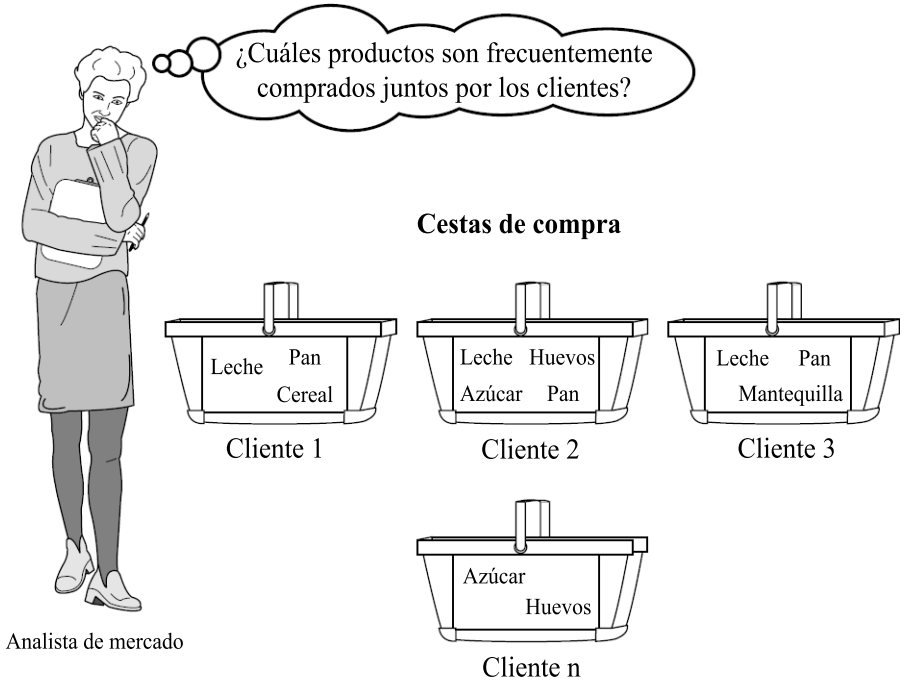


Figura 2.2: Análisis de los hábitos de compras de los clientes (patrones frecuentes).

había técnicas para analizar los datos y poder descubrir tendencias, patrones, relación etc. Pese a esto, estas técnicas en ocasiones no son suficientes o son tediosas. Estas son:

- Tradicional: Consiste en hacer consultas a los datos de cosas que ya se conocen, por ejemplo, consultar por todos los *tweets* de una cierta fecha en particular, o los últimos comentarios puestos en Facebook por un cierto usuario entre muchas más. Sin embargo, esta técnica no permite encontrar información oculta a menos que los datos sean pocos o las tendencias sean muy obvias.
- Minería de datos: Actualmente es muy utilizada, sobre todo en el área de la inteligencia de negocios (*business intelligence*). Hace mucho uso de la estadística.
- InfoVis: Es una técnica que pregunta al experto del dominio lo interesante o relevante de los datos a analizar para con ello poder crear aplicaciones interactivas que permitan ver esas situaciones, sin embargo, es poco escalable a diferencia de la minería de datos.

El Dr. Ben Shneiderman [15] clasifica los tipos de datos de la visualización de la información en siete grupos:

- **1D lineal:** Datos con una sola dimensión, como por ejemplo, la longitud de un texto.
- **2D Mapa:** Dos dimensiones, como son los sistemas de información geográfica, imágenes médicas, etc.
- **3D lineal:** Tres dimensiones, como los CAD (Diseño Asistido por Computadora), moléculas, arquitectura, etc.
- **Multivariados:** N dimensiones ($N > 3$).
- **Temporal:** Datos con una componente temporal.
- **Arboles:** Datos jerárquicos.
- **Redes:** Aquellos que se pueden representar mediante una red (nodos y aristas).

Cabe mencionar que este trabajo se centra en los datos multivariados.

En [15] se define el termino "Mantra" en la visualización de la información. Este concepto involucra siete tareas que debe tener una interfaz gráfica de usuario (UI) para la visualización de la información:

- Resumen de los datos: Tener una visualización general de todos los datos.
- Zoom: Permitir el *zoom* a partes de interés.
- Filtrado: Permitir quitar datos que no sean de interés.
- Detalles bajo demanda: Seleccionar un cierto rango de datos y presentar mayor detalle de ellos.
- Relaciones: Mostrar las relaciones entre los datos.
- Historia: Mantener un historial de las acciones del usuario para permitir dar marcha atrás (*undo*).
- Extracción: Permitir extraer un sub-conjunto de los datos.

Los cuatro primeros puntos son los principales, lo que toda UI debe permitir al usuario.

2.2.1. ¿Qué es la visualización de la información?

Existen diferentes definiciones para la visualización de la información [4, 16, 17, 18], sin embargo, en esta tesis se hace uso de la definida por Jin Zhang [18] la cual menciona:

"Visualización de la información es el proceso de transformación de los datos, información y conocimiento en una representación gráfica para apoyar tareas como el análisis de datos, predicción de tendencias, detección de patrones entre otros."

Dichos datos pueden representar objetos concretos o abstractos. Cuando los datos son abstractos, la visualización correspondiente también lo es, por ejemplo, si los

datos representan ventas o costos la visualización correspondiente suele ser una gráfica circular o de barras.

Los objetivos de la visualización de la información son representar información de una manera intuitiva y natural [19], es decir, la imagen resultante debe ser clara para el usuario, sin que presente ninguna ambigüedad, de lo contrario es una mala visualización.

Para nosotros, una buena visualización es una imagen que permite al usuario comprender las relaciones que los objetos mostrados guardan con sus diferentes atributos (o variables).

2.2.2. Historia de la visualización

La visualización no es algo nuevo. Por ejemplo, las pinturas rupestres encontradas en rocas y cavernas, las cuales representaban en su mayoría personas, animales y el medio ambiente con la finalidad de mostrar el comportamiento habitual de la comunidad. Los chinos crearon el primer mapa conocido en el siglo XII pero no fue sino hasta el siglo XIX que apareció la primera representación multidimensional.

Los doctores John Snow y Charles Joseph Minard crearon dos de los mejores ejemplos. En 1854 en Londres, se produjo un brote de cólera que mato a cientos de personas en menos de una semana. El Dr. Snow que en aquel entonces hacia uso de mapas en sus artículos y exposiciones, utilizó uno de ellos para marcar cada muerte así como cada pozo de agua (Figura 2.3). Snow se percató que la mayoría de las muertes habían ocurrido cerca de *Broad Street*; además descubrió que la cantidad de víctimas se concentraban en los alrededores del pozo de agua de *Broad Street*. John Snow concluyó que el problema del cólera provenía de los pozos contaminados, por lo cual las autoridades clausuraron todos los pozos contaminados dando fin a la epidemia de cólera [17, 19].

El Dr. Minard creó probablemente en 1861 la primer gráfica estadística alguna vez dibujada [17]. Creó un grafo (Figura 2.4) donde mostraba las pérdidas del ejercito de Napoleón en 1812 durante la invasión a Rusia. En dicho gráfico se muestran diferentes variables como son:

- Situación y dirección del ejército, mostrando como se dividen y agrupan

Figura 2.3: Mapa del Dr. John Snow. Los puntos son casos de cólera y las cruces representan los pozos de agua.

- La temperatura
- La pérdida de los soldados

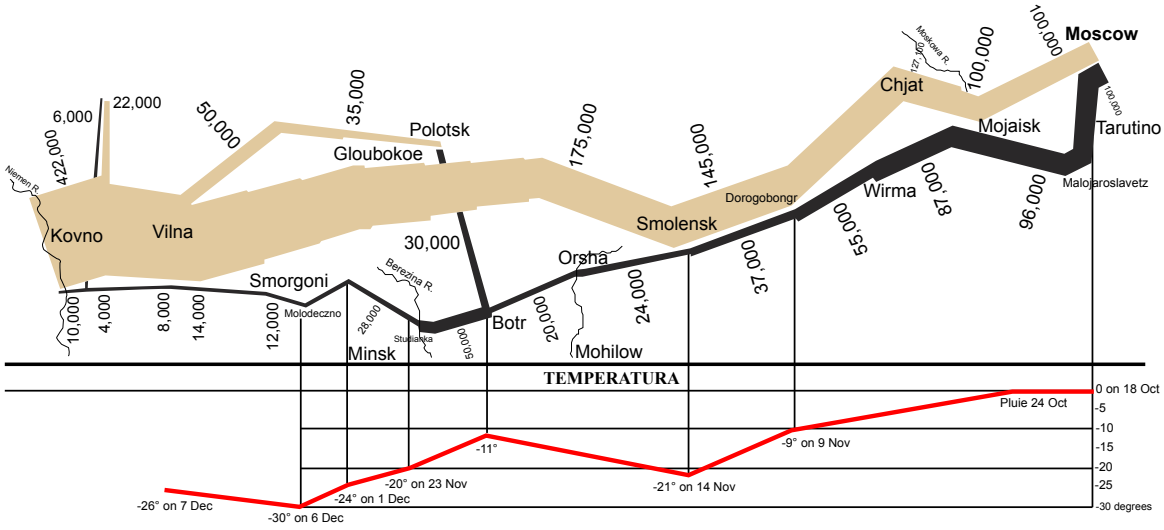


Figura 2.4: Mapa de las pérdidas de hombres del ejército de Napoleón durante la invasión a Rusia en 1812.

Minard representó al ejército mediante una banda cuyo ancho representaba el tamaño. Inicialmente el ejército contaba con 422 000 efectivos pero al momento de la llegada a Moscú solo quedaba 100 000 elementos. La banda de color negra representa la retirada y la línea roja la temperatura. Durante la retirada también sufrió pérdidas, cuando llegaron a París solo eran 10,000 elementos.

Concluyó que la temperatura fue la principal causa de la disminución del ejercito de Napoleón, es decir, que a menor temperatura el tamaño del ejercito disminuía más [17, 19].

2.3. Estado del arte

El área de la visualización de la información es un campo activo para la investigación, porque, aunque es sencillo para una computadora dibujar automáticamente una gráfica, no lo es mostrarla de tal manera que transmita al usuario la información

que los datos extraen de la realidad, sobre todo cuando son muchos datos o tienen muchos atributos.

Algunas técnicas para la representación de grafos son (más técnicas en [4, 19, 20, 21]):

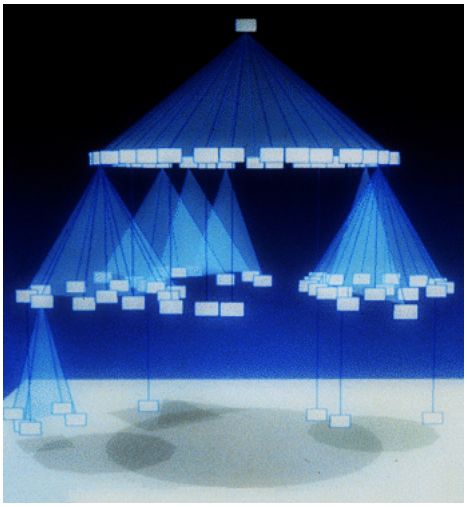


Figura 2.5: Ejemplo de un grafo Cone Tree. Imagen obtenida de [2]

- **Cone Tree:** Es una técnica propuesta por Robertson et al. [2] en la cual la información jerárquica es presentada como un árbol en 3D que luce como conos (Figura 2.5).
- **Tree-Map:** Técnica propuesta por Johnson B. y Shneiderman B. [22] que consiste en representar información jerárquica mediante rectángulos anidados. Cada rama del árbol será un rectángulo y dentro de este habrá otros rectángulos más pequeños que representan las sub-ramas (Figura 2.6).
- **Algoritmos de posicionamiento radial:** Algoritmos que buscan colocar los objetos a visualizar de manera circular. Cuando se usan con datos jerárquicos el centro del círculo suele representar al nodo raíz.
- **Cubos de información:** Es una técnica propuesta por Jun Rekimoto y Mark Green para representar información jerárquica usando cubos semi-transparentes anidados en 3D [23].

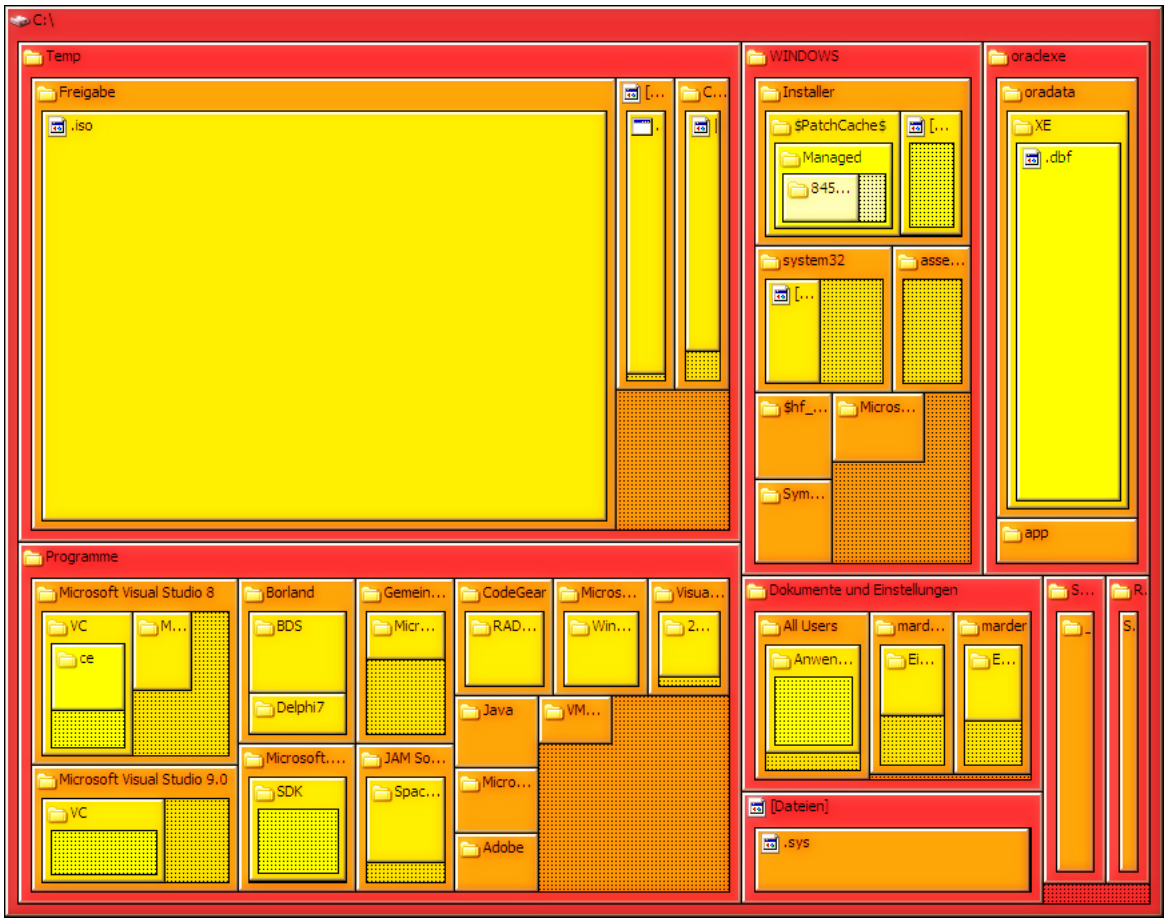


Figura 2.6: Ejemplo de un grafo Tree-Map. Muestra el uso del espacio del disco duro en Windows. Cada rectángulo representa un directorio o archivo del disco duro y están anidados para representar la jerarquía del árbol.

Estas son solo algunas de las de las muchas técnicas que han surgido para la representación de la información mediante grafos. A continuación se describe una técnica de visualización para datos relacionados y con una componente temporal, llamados mapas historiográficos.

2.3.1. Mapas historiográficos

Los mapas historiográficos han sido estudiados desde hace varias décadas. Eugene Garfield e Irving Sher propusieron en la década de los sesentas un algoritmo historiográfico. Pese a esto, no tuvo gran impacto en aquella época y no ha sido sino hasta recientemente que ha tenido un gran desarrollo gracias al software de HistCite y Alexander Pudovkin [24].

Un mapa historiográfico es un tipo de grafo donde los nodos están agrupados en periodos de tiempo (horas, días, meses, años etc.) y son mostrados en orden cronológico (Figura 2.7).

La metodología de HistCite se basa en varios pasos [25, 26]. Primero, obtener todos los datos. Segundo, limpiar los datos, es decir, eliminar datos duplicados, información irrelevante etc. Como tercer paso, hay que editar las variaciones en las referencias citadas que puedan impedir el correcto establecimiento de los arcos en el grafo. Al final se obtiene un archivo el cual es usado por HistCite para crear el mapa historiográfico.

Actualmente, el software está configurado para importar bibliografías creadas por las búsquedas de la *Web of Science* que ofrece *Thomson-Reuters Scientific*, sin embargo, es posible agregar de forma manual otras fuentes.

Los doctores Lutz Bornmann y Werner Marx [27] usaron el software de HistCite para analizar la estructura y relaciones de 45 artículos publicados entre el 2005 y 2010 usando el índice h[1]. Cada artículo fue clasificado en una de las seis categorías que propusieron y se calculó el Puntaje Global de Citas (GCS por sus siglas en ingles). Teniendo esta información crearon el mapa historiográfico. En el mapa pudieron observar que todos los artículos fueron citados al menos cinco veces. Este tipo de análisis ayuda a los investigadores y analistas a determinar los artículos principales

[1]Propuesto por Hirsch J. Es un sistema para la medición de la calidad profesional de los científicos en función del número de citas que reciben sus artículos.

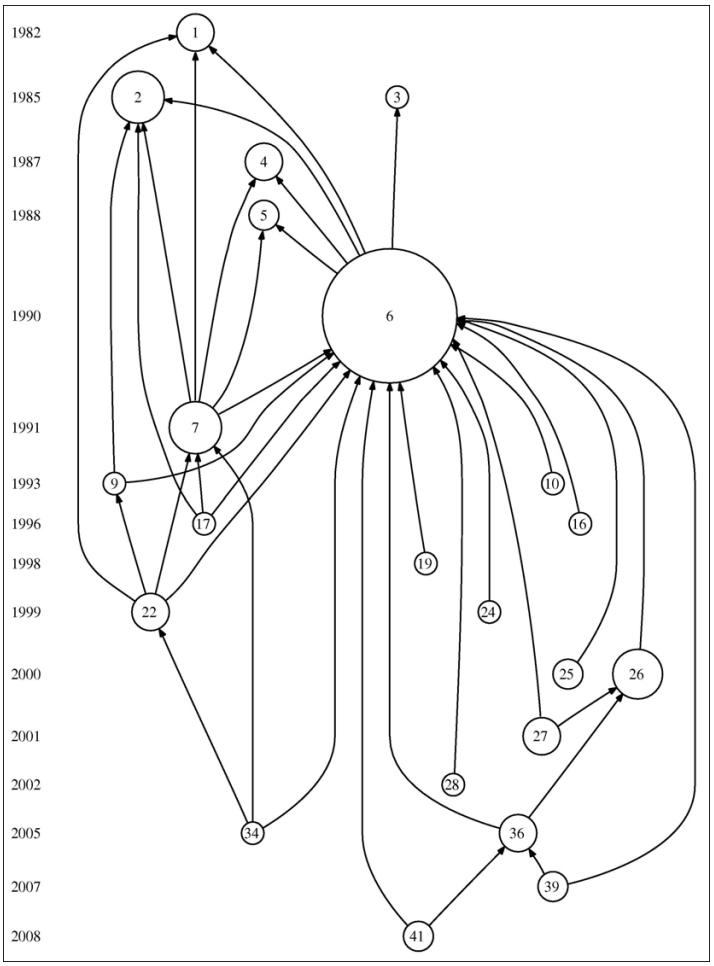


Figura 2.7: Ejemplo de un mapa historiográfico.

(más citados) así como identificar los temas que están en desarrollo entre otras cosas.

S. Raja y R. Balasubramani realizaron un análisis cienciométrico sobre las publicaciones de los investigadores referentes a la Malaria en el periodo de 2003 al 2007 [28]. Durante este período se publicaron 15685 artículos en este campo. Su análisis encontró entre otros resultados que en el 2006 fue el año en que más publicaciones hubo con 3731 además de identificar al investigador con mayor producción científica. Así mismo, se determinó que el país con mayor publicación de artículos fue Estados Unidos de América y el que tuvo menor publicación fue la India.

Ambos estudios utilizan técnicas diferentes, el primero se basa principalmente en técnicas de visualización para presentar la información de forma visual y obtener conocimiento mientras que el segundo se basa en la cienciometría. Son dos enfoques diferentes pero que suelen estar ligados.

2.3.2. Herramientas de visualización de la información

En la actualidad existen un sinfín de herramientas de visualización, algunas enfocadas a un rubro en particular, como puede ser la inteligencia de negocio (*business intelligence*), otras de propósito general e incluso algunas de propósito específico, por ejemplo HistCite, la cual es una herramienta que genera mapas historiográficos únicamente y así como esta existen muchas más que únicamente generan un solo tipo de visualización, como pueden ser árboles, redes, gráficas de barras, etc. Cabe mencionar que aquellas aplicaciones enfocadas a generar grafos de manera automática deben cumplir con los objetivos de la visualización antes mencionados, más aún, si no permiten la modificación por parte del usuario una vez creado.

Segun Chaomei Chen [19] existen diversos criterios que deben cumplir los algoritmos para la creación de grafos no dirigidos. Estos criterios son la velocidad de creación, la simetría, distribución uniforme de los nodos, longitud uniforme de las aristas y la minimización del cruce de aristas. Existen algoritmos que se enfocan únicamente a algunos criterios pues en ocasiones estos suelen ser excluyentes. Por ejemplo, crear un grafo simétrico podría requerir que algunas aristas se crucen y evitar que se crucen generaría un grafo no simétrico. Por lo tanto, los algoritmos deben permitir cierta flexibilidad sobre estos criterios.

A continuación se presentan algunas herramientas de visualización, algunas de

ellas fueron extraídas de *KDnuggets*[2] y de *Datavisualization.ch Selected Tools*[3] y otras por experiencia del autor.

Tableau Software

Esta es quizá una de las herramientas más completas para el análisis y visualización de datos. Se centra principalmente en la llamada inteligencia de negocios (*business intelligence*).

Entre sus principales características están:

- Conexión directa a bases de datos.
- Permite importar/exportar datos desde Excel.
- Genera diversas gráficas como son barras, dispersión, de líneas, mapas, etc. e incluso permite combinar algunas.
- Conexión con servicios de mapas de terceros, como Google Maps, Bing Maps, Yahoo Maps entre otros.
- La mayoría de las funciones son mediante arrastrar y soltar (*drag and drop*).
- Compatibilidad con dispositivos móviles.
- Trabaja con cubos de datos OLAP.
- Genera estadísticas directamente de los datos para luego hacer la visualización.

Tableau posee un motor de datos para una vez cargada la información poder acceder a ella de una manera rápida y eficiente. Entre los objetivos de dicho motor están:

- Innovación en la forma que se cargan los datos desde el disco y como se llevan a cabo las operaciones de búsqueda.

[2]Pagina especializada en Minería de datos. http://www.kdnuggets.com

[3]Pagina especializada en noticias y recursos sobre visualización de la información e infografías. http://selection.datavisualization.ch/

- No necesita de un modelo de datos fijo.
- No se requiere saber por adelantada la carga de trabajo de las consultas.
- Rendimiento consistente y predecible.
- Diseñado para consultas de análisis visual.
- Integración con bodegas de datos.

Más información en http://www.tableausoftware.com.

Graphviz

Es un paquete para la visualización de información estructurada mediante grafos. Posee muchos *layouts* para la construcción de los grafos, por lo cual el usuario únicamente debe definir los datos y al ser software libre es posible agregar nuevos *layouts*.

Posee versiones para Linux, Windows y Mac además de permitir su uso vía Web. Este software se centra en la creación de los grafos por lo cual no posee una interfaz gráfica de usuario muy desarrollada sino que está pensada para ser integrada dentro de otras aplicaciones.

Graphviz toma la descripción del grafo de un simple lenguaje de texto y genera los diagramas en múltiples formatos, tales como imágenes, SVG, páginas web, PDF o Postscript para incluirlos en otros documentos, también soporta GXL[4] y XML.

Posee muchas funciones útiles para la manipulación de los grafos, como opciones para colores, tipografías, diseño de tablas de nodos, estilos de líneas, formas personalizables entre otras.

Entre los *layouts* que posee están:

- **dot:** Es un *layout* para el dibujado de grafos dirigidos y árboles. Es la opción por defecto si las aristas son dirigidas.

[4]Graph eXchange Language, es una extensión de XML para el intercambio de grafos.

- **neato:** Es la opción recomendada para grafos no muy grandes (máximo 100 nodos).
- **fdp:** Es similar a neato.
- **sfdp:** Versión multiescala del fdp para el manejo de grafos grandes (más de 100 nodos).
- **twopi:** *Layout* radial. Los nodos son posicionados sobre círculos concéntricos dependiendo de su distancia a un nodo raíz definido.
- **circo:** *Layout* circular. Es recomendado para diagramas con estructuras cíclicas múltiples, tales como las redes de telecomunicaciones.

En la figura 2.8 se muestran algunos ejemplos de esta herramienta.

Para más información consultar http://www.graphviz.org.

HistCite

Es un paquete de software para la creación de mapas historiográficos. Es usado para análisis bibliométricos y visualización de la información. Fue desarrollado originalmente por el Dr. Eugene Garfield, fundador del *Institute for Scientific Information*.

Su dominio está restringido al *Web of Science*, es decir, el usuario solo puede visualizar los datos resultado de búsquedas en la *Web of Science*. Sin embargo, esto es suficiente para poder hacer estudios de investigación ya que el dominio es muy amplio. Entre las preguntas que permite resolver esta herramienta están:

- Artículos importantes en el desarrollo de un tema en específico.
- Artículos importantes descartados por el buscador.
- Los autores más citados.
- Países e instituciones desde los cuales los autores publican.
- Cita de autores en grupos.

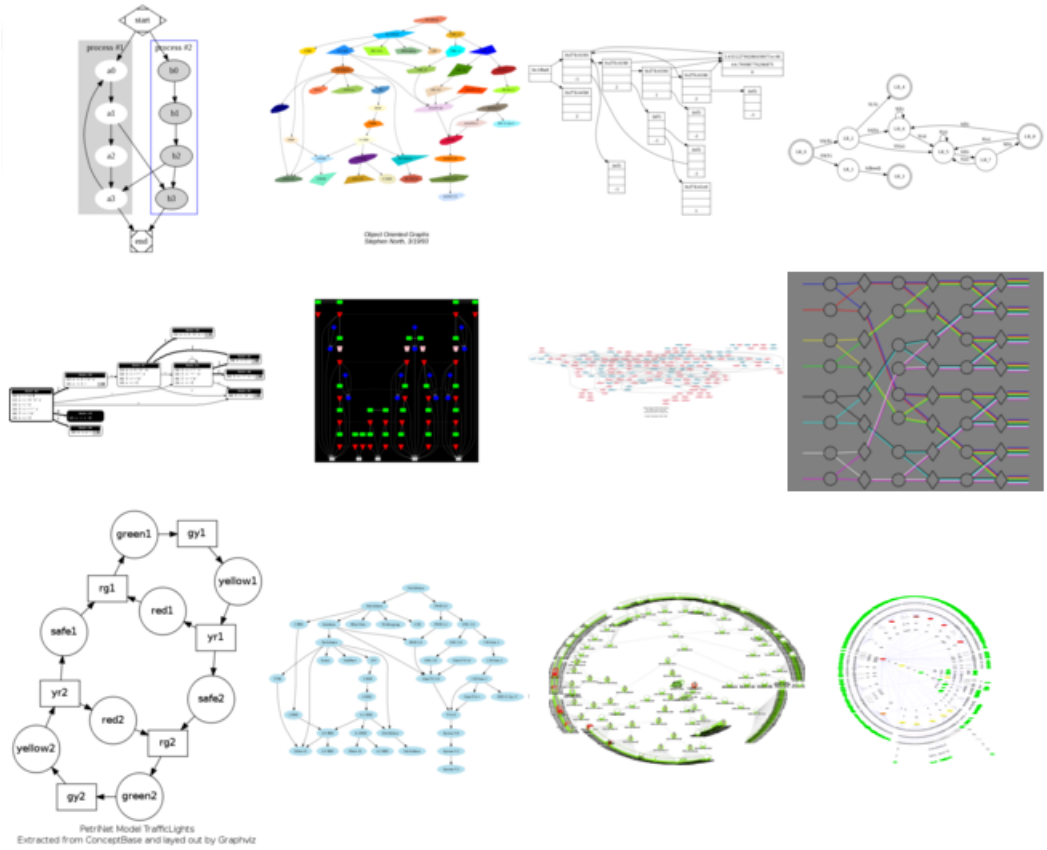


Figura 2.8: Ejemplos de grafos generados con Graphviz.

- Estadísticas de citas por grupos y subgrupos.
- Artículos altamente citados.
- Línea de tiempo de las publicaciones de los autores.

En http://thomsonreuters.com/products_services/science/science_products/a-z/histcite/ se puede obtener más información.

Graph

Herramienta útil y sencilla para graficar funciones matemáticas y datos en un sistema de coordenadas. Posee una interfaz sencilla e intuitiva y permite el cálculo de algunas operaciones básicas sobre la gráfica, como por ejemplo, permite calcular la pendiente en un punto dado y graficarla sobre la misma gráfica o calcular la integral en un cierto intervalo, todo de manera gráfica.

Entre sus características se encuentran:

- Dibujar funciones matemáticas.
- Dibujar relaciones, por ejemplo $sin(x) < cos(y)$.
- Calcula la integral de un cierto intervalo y asignar diferentes estilos y colores para cada intervalo.
- Crear series de puntos con diferentes marcas cada uno. Los datos pueden ser importados desde otros programas.
- Permite calcular de manera simbólica la primer derivada de la función y graficarla.
- Interacción con otros programas.
- Permite crear animaciones y ver el comportamiento de una función cuando los valores cambian.

En la figura 2.9 se muestran algunos ejemplos creados con esta herramienta.

Más información en http://www.padowan.dk/.

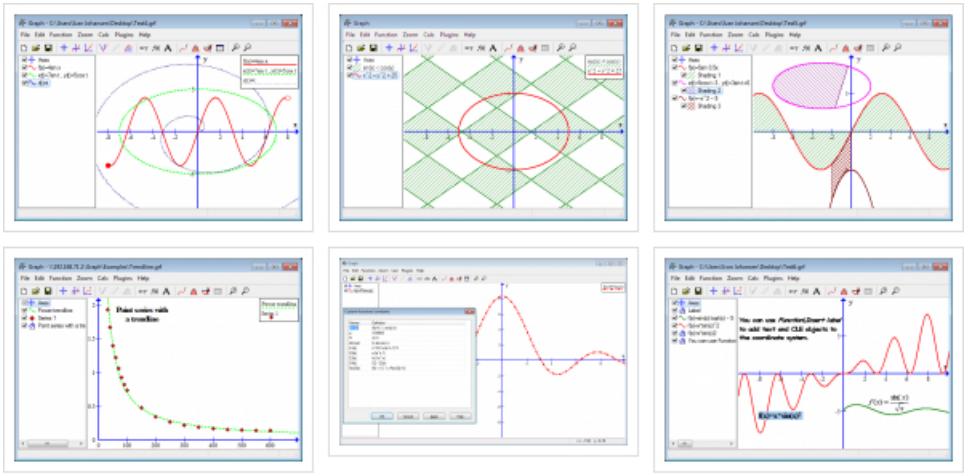


Figura 2.9: Ejemplos de graficas generadas con Graph.

Dplot

Herramienta para visualizar datos de hasta cuatro variables. Soporta el manejo de múltiples tipos de escala en cada eje, como es logarítmica, probabilística entre muchas otras.

Posee una alta personalización del grafo, como son seleccionar colores, poner una imagen de fondo, tipo de líneas, marcas entre otros.

Maneja varios tipos de grafos, como son de barras, de dispersión, de líneas, etc. además tiene un sistema de *plugins* con el cual se le puede agregar nuevas funcionalidades.

En la figura 2.10 se muestran algunos ejemplos.

Se puede encontrar más información en http://www.dplot.com/.

Autograph

Es una herramienta de propósito general que posee tres modos de operación:

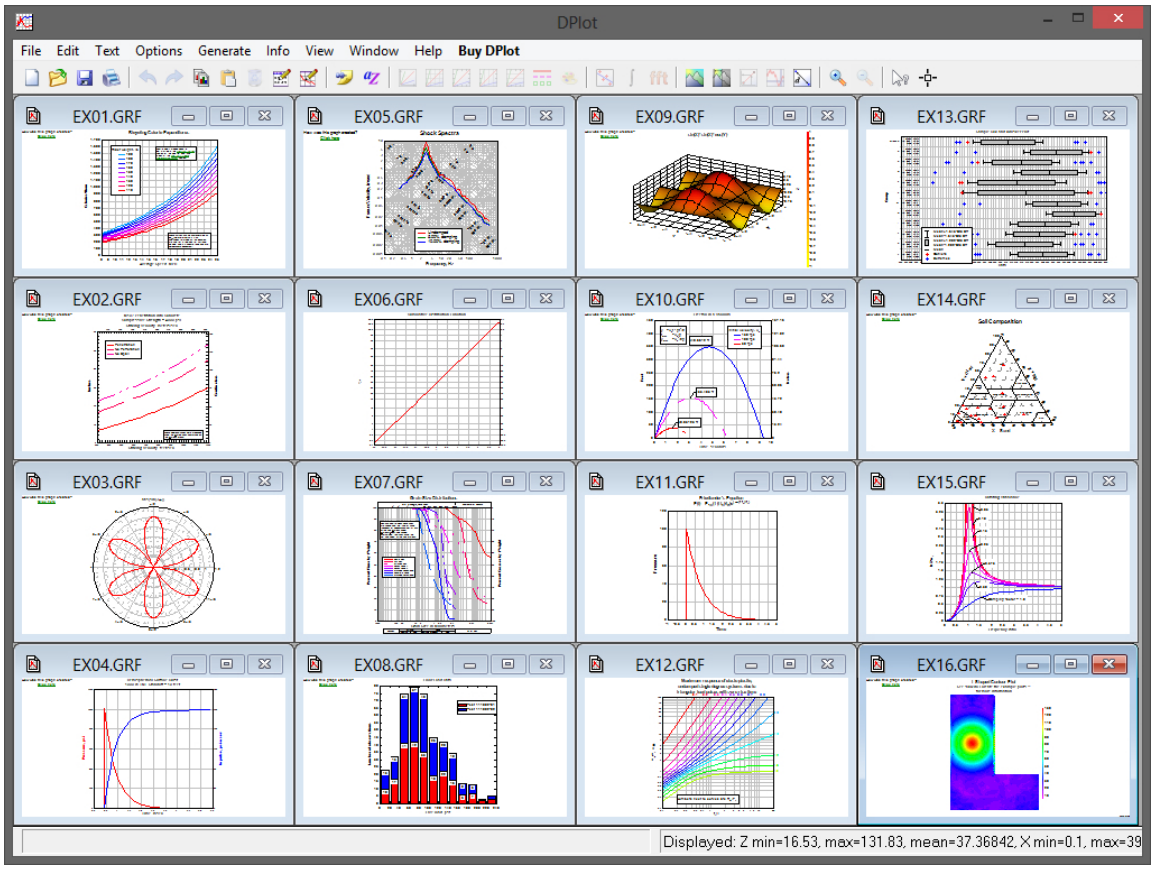


Figura 2.10: Ejemplos de graficas generadas con DPlot.

- 1D: Útil para la estadística y probabilidad.
- 2D: Para graficar datos con dos variables.
- 3D: Para tres variables.

Su interfaz es sencilla y clara, permitiendo personalizar un sin fin de opciones, tanto antes de crear el grafo como después. Esta únicamente disponible para plataformas Windows y es software con licencia comercial.

Para más información consultar http://www.autograph-maths.com/.

Gnuplot

Es una poderosa herramienta de visualización de datos en 2D y 3D mediante línea de comandos, es decir, no posee interfaz gráfica, lo cual lo hace ideal cuando se necesita interactuar con otras aplicaciones.

Actualmente existen versiones para Linux, Windows, OSX, UNIX etc. Fue creado en 1986 y desde entonces se mantiene su desarrollo gracias a ser software libre.

Entre sus características están:

- Múltiples opciones de personalización del grafo.
- Manejo de diferentes escalas de un mismo eje.
- Exporta imágenes en muchos formatos incluidos imágenes vectoriales.
- Fácil integración con otras aplicaciones.

Algunos ejemplos de graficas que se pueden crear con esta herramienta se muestran en la figura 2.11.

Más información en http://www.gnuplot.info/.

En la tabla 2.1 se muestra una comparativa de las herramientas antes mencionadas.

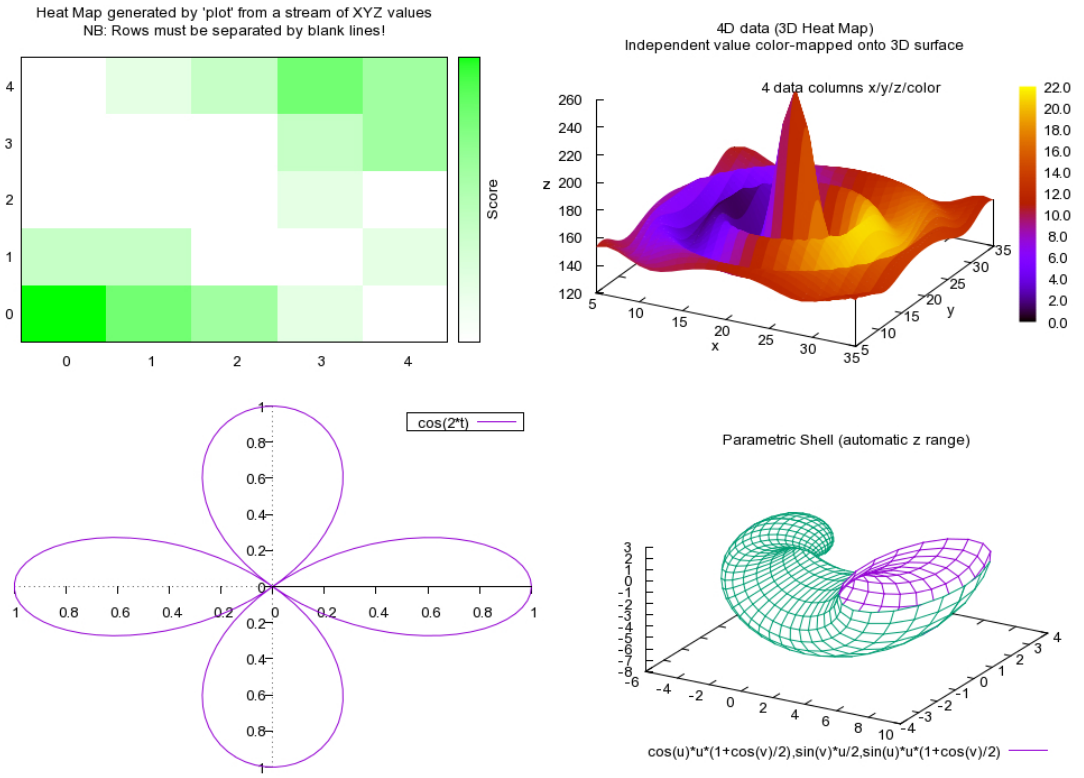


Figura 2.11: Ejemplos de graficas generadas con Gnuplot.

Nombre	Plataforma	F/C	Propósito	3D	Formato de salida
Tableau Software	Win	C	Inteligencia de negocios	No	Excel, Imágenes
Graphviz	Linux, Win, Mac y Web	F	General	Sí	SVG, HTML, PDF, PostScript, GXL y XML
HistCite	Win	C	Mapas historiográficos	No	Imágenes
Graph	Win	F	General, en sistema de coordenadas XY	No	Imágenes
DPlot	Win	C	General	Sí	Imágenes
Autograph	Win	C	General	Sí	HTML, imágenes
Gnuplot	Linux, Win, Mac, Unix	F	General, gráficos 2D y 3D	Sí	Imágenes

Tabla 2.1: Algunas herramientas de visualización de datos. F = Software libre, C = Comercial.

Capítulo 3

Análisis y diseño

Como ya se mencionó, el principal objetivo de la visualización de la información es el ayudar a entender el comportamiento de un conjunto de datos. Este trabajo busca encontrar la mejor forma de organizar las variables de un conjunto de datos y visualizarlos.

Para esto se requiere de identificar y seleccionar las variables que se desean mostrar, porque mostrar muchas variables no suele ser útil (más de cuatro variables puede ser confuso en algunos casos). También se deben considerar factores como posición de los nodos, tamaño, forma, color etc.

En este capítulo se describe el método para organizar las variables y se comenta de manera general los módulos que tendrá el software. Estos módulos son:

1. Módulo para el manejo de la base de datos.
2. Módulo para el cálculo de los mínimos cuadrados.
3. Módulo para el cálculo de MARS.
4. Módulo de búsqueda de particiones
    - Análisis de variables numéricas
    - Análisis de variables simbólicas

5. Generación de la visualización
    - Crear etiquetas, ejes cartesianos, etc.
    - Calcular posición de los objetos de interés (puntos en el espacio).
    - Selección de los colores.

A continuación se describen cada uno de estos de manera general y más adelante los algoritmos.

3.1. Módulo para el manejo de la base de datos

Este módulo es el encargado de realizar todas las operaciones que se requieran en la base de datos. Se optó por el uso de una base de datos en lugar de archivos de texto u otro medio de almacenamiento debido a que los datos pueden ser muy grandes y una base de datos permite almacenarlos y extraerlos de manera eficiente. Se maneja una estructura predefinida donde existe una tabla general llamada *structures* y otra donde se almacenan la información.

En la tabla donde se almacenan la información, la primera columna será una llave primaria y los siguientes campos serán los atributos de los datos, no importando el orden.

La tabla *structures* tiene dos atributos:

- **table**: Es el nombre de la tabla de datos.
- **struct**: Es un vector binario donde cada entrada corresponde a los atributos de la tabla de datos, siendo el 0 para datos numéricos y 1 para datos simbólicos. La llave primaria de la tabla de datos no se considera, sin embargo, el orden de los valores del vector debe ser el mismo que los atributos en la tabla de datos.

3.2. Módulo para el cálculo de los mínimos cuadrados.

Este módulo como su nombre lo indica, es el encargado de calcular los mínimos cuadrados de cada par de variables numéricas del conjunto de datos. Para ello deberá solicitar los datos al módulo 1. Sus resultados serán usados por el módulo 4 y posteriormente terminando el módulo 4 serán usados por el módulo 5 para generar la visualización. En la sección 3.6 se describe el algoritmo.

3.3. Módulo de búsqueda de particiones

Este módulo tiene dos sub-módulos, uno para el manejo de las variables numéricas y otro para el manejo de las variables simbólicas. Ambos sub-módulos tienen la función de buscar si una variable se particiona sobre otra variable (más adelante se explica a detalle). Los datos deben de ser solicitados al módulo 1 y al módulo 2 o 3 según el algoritmo a ejecutar. Al finalizar, el resultado es dado al módulo 5.

3.4. Módulo para el cálculo de MARS

Es el encargado de aplicar el algoritmo de MARS a cada par de variables numéricas del conjunto de datos para posteriormente pasar sus resultados al módulo 4 y 5. Los datos al igual que el módulo 2 deben ser solicitados al módulo 1. En la sección 3.7 se describe el algoritmo que usa este módulo.

3.5. Generación de la visualización

Módulo encargado de generar la visualización, entre otras cosas calcula la posición de cada objeto de interés en el plano cartesiano, las etiquetas, los colores de los objeto y al final el resultado es enviado al navegador del usuario. En el siguiente capítulo se mencionan las herramientas usadas para la visualización.

3.5.1. Selección de los colores

El color es un elemento muy importante para el ser humano, ya que a través de él se perciben diversas emociones por lo cual se debe tener sumo cuidado en la selección de estos a fin de mostrar una visualización adecuada [29].

No obstante, la asignación de colores es una tarea no trivial [29, 30, 31] y más cuando esta debe ser generada de manera automática. Zeileis et al. [31] propone que hay tres obstáculos que superar en la presentación de gráficos estadísticos los cuales pueden ser extrapolados a otros tipos de visualizaciones:

1. Colores atractivos
2. Colores en cooperación con otros: La idea de incluir colores en un grafo es la de distinguir entre diferente grupos o niveles de una variable. Para ello se generan varios colores llamados paleta.
3. Claridad del color donde sea: Los colores seleccionados para el grafo deben ser distinguibles unos de otros independientemente de donde se muestren, por ejemplo, en un LCD[1], proyector, impreso etc.

Cabe mencionar, que los objetivos anteriores no siempre se pueden garantizar pero es recomendable poner atención a esta situación con la finalidad de presentar al usuario un grafo lo más claro posible.

Modelos de color

Actualmente existen diferentes maneras de representar colores y esto se hace mediante vectores, de tal manera que al combinarlos linealmente generan todo el espacio de color. La mayoría de los modelos de color generan la mayor cantidad de colores visibles por el ojo humano (ver figura 3.1) aunque existen otros que solo generan un subconjunto de estos.

Dentro de los modelos de color populares están:

[1]Pantalla de cristal líquido por sus siglas en ingles

Figura 3.1: Espectro visible por el ojo humano.

- **RGB**: Está formado por los colores primarios, rojo, verde y azul (RGB por sus siglas en ingles). Estos colores fueron escogidos porque cada uno corresponde aproximadamente con uno de los tres tipos de conos sensitivos del ojo humano, esto es, 65 % sensible al rojo, 33 % al verde y 2 % al azul. Este modelo es usado por monitores para representar el color siendo el (0,0,0) el negro y (255,255,255) el blanco. En la práctica es común que el vector se encuentre normalizado.

- **CMYK**: Similar al RGB pero se basa en cuatro colores, cian, magenta, amarillo y negro. Su uso es principalmente en impresoras ya que logra mejores contraste comparado con el modelo anterior.

- **HSV**: Define el color en base a sus componentes que son:

  - Matiz (*Hue*): Se representa con un valor de 0 a 360 que corresponde a los grados de un círculo y su valor corresponde a un color.
  - Saturación (*Saturation*): Representa la distancia horizontal del centro del circulo hacia afuera. Los valores posibles van del 0 al 100 %. Se puede considerar como la mezcla de un color con blanco o gris.
  - Valor (*Value*): Es la intensidad de luz, es decir, establece que tan claro u obscuro es el color.

  Es útil para generar colores automáticamente por lo cual el sistema de visualización propuesto en este trabajo usa este modelo. Una representación gráfica puede verse en la imagen 3.2.

Estos modelos no son los únicos, existen otros como el RYB, YIQ, HCL etc. cada uno con ventajas y desventajas.

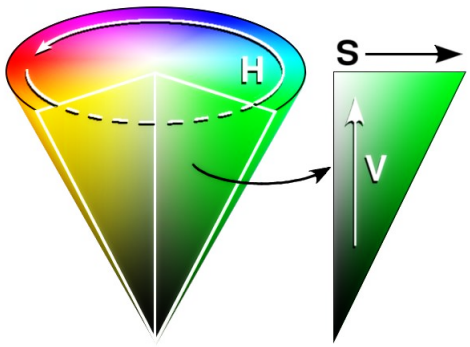


Figura 3.2: Modelo de color HSV.

Paleta de colores

Como ya se mencionó, el modelo de color a usar es el HSV por su facilidad de uso. Teniendo este modelo hay que generar una paleta de colores[2]. En [31] se describe la manera de construir diversas paletas de colores, como son paletas cualitativas, secuenciales y divergentes. El software aquí propuesto utiliza paletas cualitativas las cuales se generan al fijar los valores de S y V y solo variar H. Para generar la paleta de colores usados en los valores de los objetos a visualizar, se sigue el siguiente algoritmo:

1. Fijar S y V a 50 y 80 respectivamente (Con esto se obtienen colores pastel como se recomienda en [31]).
2. Se selecciona un valor aleatorio entre 0 y 360 que se asigna a H.
3. A partir del valor asignado a H este se ira variando en incrementos de 30 modulo 360.
4. Este procedimiento se repite hasta que no haya más colores que generar.
5. Si se requieren más colores, se seleccionan nuevos valores de S y V. Para garantizar que las tonalidades de los colores sean los suficientemente diferentes (descartar colores análogos) se propone variar S y V en 10 unidades donde $10 \leq S \leq 100$ y $50 \leq V \leq 100$

[2]Conjunto de colores disponibles. Pueden ser todos los colores generados por el modelo de color o bien un subconjunto de este.

6. Repetir lo anterior hasta que se obtenga el número de colores deseados. Si se requieren aún más colores, es necesario modificar los parámetros de incremento/decremento.

Con estos parámetros se pueden obtener 720 colores diferentes lo que se traduce en que el sistema puede manejar 720 categorías o clasificaciones.

3.6. Algoritmo usando mínimos cuadrados (Algoritmo LS)

Dado un conjunto de datos con n variables tanto numéricas como simbólicas, se pretende encontrar la mejor agrupación de éstas para poder generar una visualización. Para lograr esto, el trabajo se divide en dos partes, uno para el análisis de las variables numéricas y otro para el análisis de las variables simbólicas.

Este análisis consiste en ver cuáles variables numéricas y simbólicas pueden mostrarse sobre el mismo eje. Básicamente para las numéricas son aquéllas que están monótonamente relacionadas, es decir, que cuando una crece, la otra también crece (o decrece). Para las simbólicas, son cuando existe una partición sobre algún eje.

Los pasos generales (a detallarse más adelante) son los siguientes:

1. Sea C el conjunto de datos de n variables. Poner todas las variables numéricas en un conjunto A y todas las simbólicas en un conjunto B, es decir:

$$A = \{x_i \mid x_i \textit{ es una variable numérica},\ i \in [1, ..., k]\}$$
$$B = \{y_i \mid y_i \textit{ es una variable simbólica},\ i \in [k+1, ..., n]\}$$

2. Para cada x_i y x_j en A, $i < j$, se calcula la recta que mejor ajusta a estas dos variables usando mínimos cuadrados (LS) y se calcula su error cuadrático medio (ECM).

3. De la función resultante, se calcula una franja y se determina si el porcentaje de los valores que caen dentro de ella es igual o mayor que un umbral μ(ver figura 3.3).

4. En caso afirmativo, esas dos variables se desplegarán en el mismo eje. En caso negativo, estas variables irán en ejes distintos.

5. Se determina el acomodo de cada par de variables, buscando que la mayor cantidad de éstas quede en un solo eje.

6. A cada eje se le asigna una puntuación y aquellos con la puntuación más alta son los candidatos a ser desplegados.

7. Las variables numéricas sobrantes se tratarán de ajustar en alguno de los ejes resultantes buscando si hay una partición en el eje donde la variable tenga un buen encaje. Si no existe dicha partición en ningún eje la variable es colocada en un eje independiente.

8. Sea la variable simbólica $y_i \in B \ \forall i \in [k+1, ..., n]$. Analizar la frecuencia de los valores diferentes de dicha variable y descartar los valores que estén por debajo de un umbral δ.

9. Para cada $y_i \in B$ se particionan sus valores en dos o tres conjuntos disjuntos no triviales.

10. Sea e_i un eje. Se divide el eje en dos o tres partes, tratando de que cada elemento de la partición de y_i quede en un segmento de la partición del eje. Habrá elementos que caigan en un intervalo incorrecto del eje, estos elementos se llamaran "desobedientes" (ver figura 3.8).

11. Se selecciona el par $(partición, eje)$ cuya desobediencia sea menor y la variable será desplegada en ese eje.

12. Generar la visualización.

A continuación se detallan los pasos anteriores.

Como ya se mencionó, del conjunto de datos se crean dos conjuntos A y B, uno de variables numéricas y otro de variables simbólicas tales que:

$$A = \{x_i \mid x_i \ es \ una \ variable \ numérica, \ i \in [1, ..., k]\}$$
$$B = \{y_i \mid y_i \ es \ una \ variable \ simbólica, \ i \in [k+1, ..., n]\}$$

El siguiente paso es calcular la mejor recta que pasa por todos los puntos de cada par de variables numéricas. Para esto se hace uso del método llamado mínimos cuadrados (Ver figura 3.3).

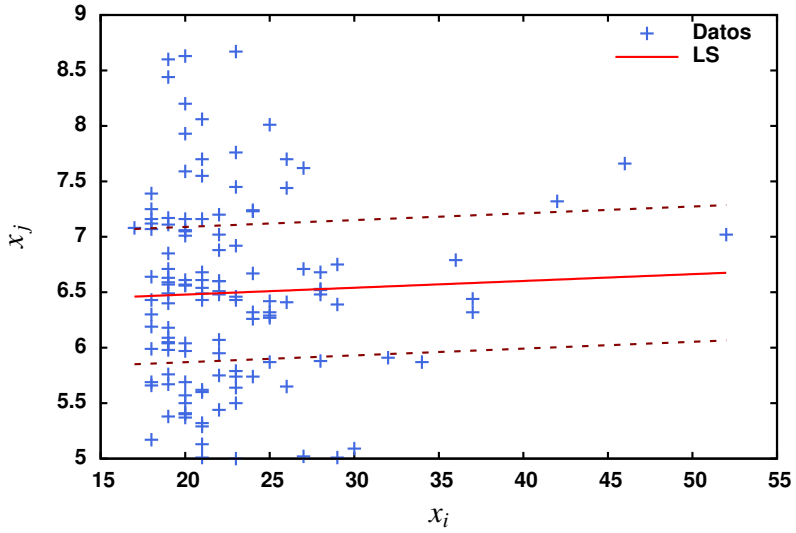


Figura 3.3: Ejemplo del método de mínimos cuadrados. En rojo, la recta que mejor ajusta a los datos (puntos azules).

Sean $x_i, x_j \in A$ tales que $i < j \ \forall i \in [1, ..., k-1], j \in [i+1, ..., k]$; se calculan los mínimos cuadrados y el ECM de la siguiente manera:

$$\begin{aligned} m &= \frac{k\sum x_i x_j - \sum x_i \sum x_j}{k\sum x_i^2 - |\sum x_i|^2} \\ b &= \frac{\sum x_j \sum x_i^2 - \sum x_i \sum x_i x_j}{k\sum x_i^2 - |\sum x_i|^2} \\ ECM &= \sqrt{\frac{1}{k}\sum_{l=1}^{k}[y_k - f(x_k)]^2} \end{aligned} \tag{3.1}$$

donde m representa la pendiente de la recta que mejor ajusta los puntos, b la ordenada al origen, k es el número de datos y

$$f(x) = mx + b. \tag{3.2}$$

es la mejor recta que ajusta los puntos.

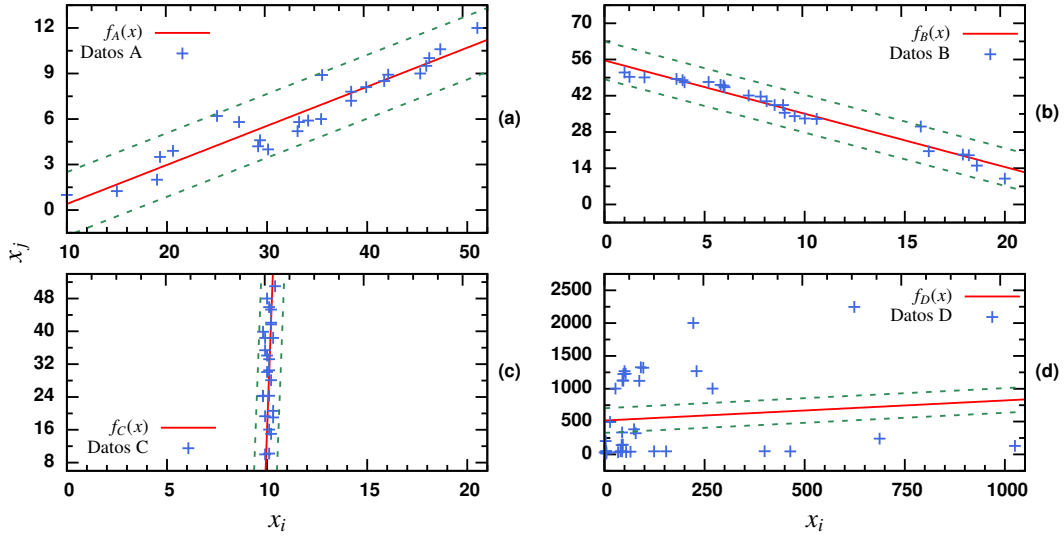


Figura 3.4: Diferentes casos posibles al calcular LS. En (a) $f_A(x)$ creciente y al menos μ de los puntos caen dentro de la franja. (b) $f_B(x)$ decreciente y al menos μ de los puntos caen dentro de la franja. (c) $f_C(x)$ constante o casi constante (los valores varían muy poco). (d) $f_D(x)$ no se ajusta, hay menos de μ puntos dentro de la franja. En nuestro caso siempre se uso $\mu = 90\,\%$.

Ahora bien, pueden suceder los siguientes casos:

(a) La recta que ajusta a x_i y x_j es creciente y al menos μ de los valores caen dentro de la franja (ver figura 3.4a). En todos los casos, solo se va a permitir que queden fuera a lo más $10\,\%$ del total de los datos. Este valor puede ser modificado dependiendo de la precisión que se desee, sin embargo, en este trabajo en todas las pruebas se usó este valor.

En este caso se puede usar el mismo eje para graficar ambas variables con diferentes escalas como se ve en la figura 3.5. Para todo par de variables que cumplan esto se puede usar el mismo eje siempre y cuando entre cualesquiera dos variables del eje, μ valores caigan dentro de la franja correspondiente.

(b) La recta que ajusta a x_i y x_j es decreciente y al menos μ valores caen dentro de la franja (ver figura 3.4b)

Similar al caso anterior, pero aquí una variable crece mientras la otra decrece, para esto se puede usar un eje para graficar ambas variables, salvo que una irá al revés tal y como se ve en la figura 3.6. Todas aquellas variables con esta

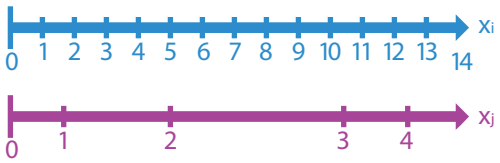


Figura 3.5: La escala de x_i y x_j es diferente pero ambas crecientes. Se usa un mismo eje para graficar las dos variables. Cabe mencionar que los rangos de la variable x_j no necesariamente tienen que ser uniformes, solo crecientes.

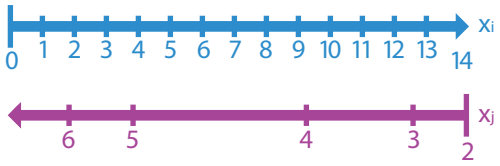


Figura 3.6: Las variables x_i y x_j son monótonas decrecientes. Se usa un mismo eje para graficar las dos variables, pero una variable irá en sentido opuesto. Los rangos de la variable x_j pueden no ser uniformes.

propiedad se podrán graficar en este mismo eje. Si existe un eje con variables monótonas crecientes (punto anterior) se puede utilizar este para graficar las monótonas decrecientes verificando que entre cualesquiera dos variables μ valores están dentro de la franja.

(c) La recta que ajusta a x_i y x_j es constante o casi constante, horizontal o vertical (ver figura 3.4c).

Aquí x_i o x_j es constante o casi constante, por lo cual, no es necesario graficar esta variable. Si la variable es casi constante, se calcula el promedio de sus valores descartando valores atípicos (*outliers*) y este valor es el que se despliega como un letrero en la gráfica resultante junto con los valores mínimo y máximo. Para verificar si una variable es constante o casi constante, se calcula el *boxplot* y se verifica que al menos μ puntos caigan dentro de los limites superior e inferior y que el valor absoluto de la diferencia de estos límites (superior e inferior) sea menor a 0.5. Este valor puede ser modificado según la precisión que se desee, pero en todas las pruebas se tomó este valor.

(d) Muchos valores de x_i y x_j caen fuera de la franja (ver figura 3.4d).

En este último caso, más del 10 % de los valores cae fuera de la franja, lo cual

significa que estas variables no se ajustaron. Más adelante se tratará a detalle el manejo de estas variables.

Ahora bien, puede suceder que dos o más variables se ajusten con otra variable, esto es, que fueron monótonas crecientes o decrecientes y al menos μ de los valores están dentro de la franja. Sin embargo, entre ellas no se ajustan, por lo cual para decidir cómo agruparlas se considera el ECM. Así las variables que tengan un menor ECM se agrupan.

Cabe mencionar que para cada par de variables el ancho de la franja es diferente debido a los rangos de las variables. Para esto se utiliza MAD ya que esta medida de distribución es poco afectada por valores atípicos y está dada por:

$$\begin{aligned} M &= \{|v_i - mediana(x)| : v_i \in x\} \\ MAD &= mediana(M). \end{aligned} \tag{3.3}$$

donde v_i son los valores de x (variable numérica) y $mediana(M)$ es la mediana de estos.

Con esto, algunas variables han sido agrupadas en diferentes ejes. Ahora bien, si tenemos uno o dos ejes libres se pueden usar para graficar las variables que no se ajustaron mediante este método. Tenemos cuando mucho tres ejes para graficar, por lo que las variables sobrantes deberán desplegarse mediante otra forma (color, tamaño, forma del objeto etc.).

Supóngase que varias variables numéricas se agruparon unas con otras en más de tres ejes, la pregunta es ¿cuáles ejes graficar? Una posible respuesta es dejar que el usuario decida, sin embargo, la intención es automatizar el proceso, por lo cual, hay que evaluar cada eje y determinar cuáles son los mejores. Para lograr esto se define la "bondad de un eje" tomando en cuenta dos propiedades:

- Mientras más variables tenga el eje, mejor.
- Mientras más pequeño sea el promedio del ECM de las variables que contiene el eje, mejor.

Aquellos ejes que únicamente tienen una variable, su bondad es cero, lo que

significa que tendrán menor prioridad sobre ejes con mas variables. Combinando ambas propiedades se define la bondad b como:

$$b = \alpha * m + (1 - \alpha)\frac{1}{p}. \tag{3.4}$$

donde m es el número de variables del eje, p es el promedio del ECM de las variables y α es un ponderador entre cero y uno que indica qué propiedad tiene mayor peso. Para todas las pruebas se tomó un valor constante de $\frac{1}{2}$.

Regresando a las variables numéricas que no se ajustaron. La pregunta es ¿hay alguna manera de poner estas variables en alguno de los ejes? Sea $D \subseteq A$ el conjunto de variables numéricas que no se ajustaron, esto es:

$$D = \{x_i | x_i \in A \ y \ x_i \ no \ se \ ajustó\}. \tag{3.5}$$

Sean $x_j \in D$ y e_i un eje, $i \in [1, ..., l]$ donde l es el número total de ejes. Dividir e_i en bloques o segmentos máximo cuatro. Se tomó este valor debido a que dividir en más segmentos puede resultar confuso para el usuario. Tomando los valores de la variable x_j, colocarlos en cada segmento del eje. Analizar los valores de cada uno de estos y determinar si hay valores similares, por ejemplo, si en un segmento hay valores pequeños, en otro medianos y en otro grandes como se muestra en la figura 3.7.

Puede suceder que algunos valores caigan en el segmento equivocado, a estos se les llamarán "valores desobedientes". Se va a permitir que exista un pequeño número de estos valores, es decir, que en cada segmento, no se sobrepase un cierto umbral β, de lo contrario se considera que x_j no se ajusta en e_i. En todas las pruebas $\beta = 10\,\%$ del total de los datos.

Con esto algunas variables sobrantes pueden ser ajustadas en los ejes. Sin embargo, existe el caso de que una variable se ajuste en dos o más ejes, por lo que se debe tomar aquel eje en el cual se ajuste mejor. En el siguiente capítulo se detalla como evaluar esto. Si la variable no se logró ajustar en ningún eje, irá en un eje independiente y las variables restantes de D también se tratarán de ajustar en este nuevo eje.

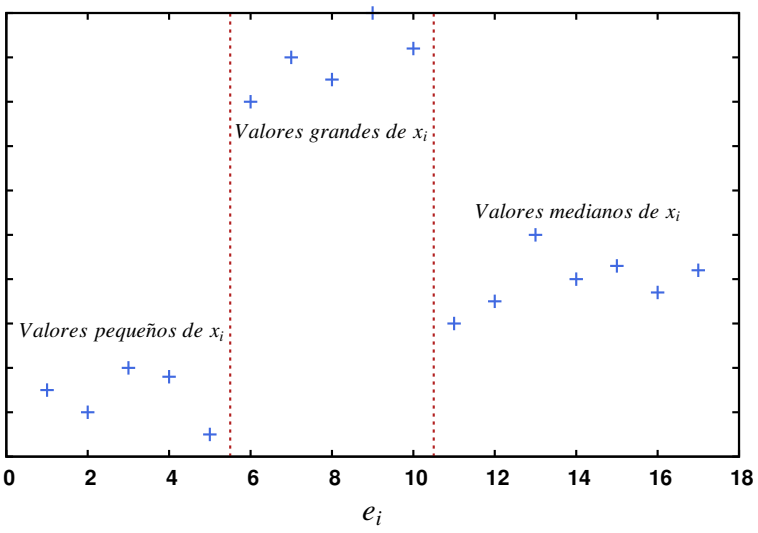


Figura 3.7: Ajuste de una variable numérica sobrante a lo largo de un eje.

Hasta aquí termina el análisis de las variables numéricas. Falta analizar las variables simbólicas, es decir, las $y_i \in B$.

Sea $y_i \in B \ \forall i \in [k+1, ..., n]$. Analizar la frecuencia de los valores diferentes de dicha variable esto con la finalidad de descartar los valores que estén por debajo de un umbral δ (en todas las pruebas realizadas $\delta = 5\,\%$ del total de los datos). Por ejemplo, supóngase que la variable y_i tiene diez valores diferentes, de los cuales un valor solo aparece una vez en todo el conjunto de datos. Si $\delta = 2$ este valor lo descartamos. Con esto, todas las variables simbólicas tendrán valores más o menos numerosos.

Ahora, para cada $y_i \in B$ se particionan sus valores en dos o tres conjuntos disjuntos no triviales (más subconjunto podría generar confusión al usuario). Sea e_i un eje. Se divide el eje en dos partes (tres máximo), tratando de que cada elemento de la partición de y_i quede en un segmento de la partición del eje. Habrá elementos que caigan del lado equivocado, estos elementos se llamarán "desobedientes" (ver figura 3.8). Se permite un máximo β de valores desobedientes.

En un caso ideal, no habrá elementos desobedientes (encaje perfecto), sin embargo, en general esto no ocurrirá, por lo que hay que buscar la división cuya desobediencia sea mínima.

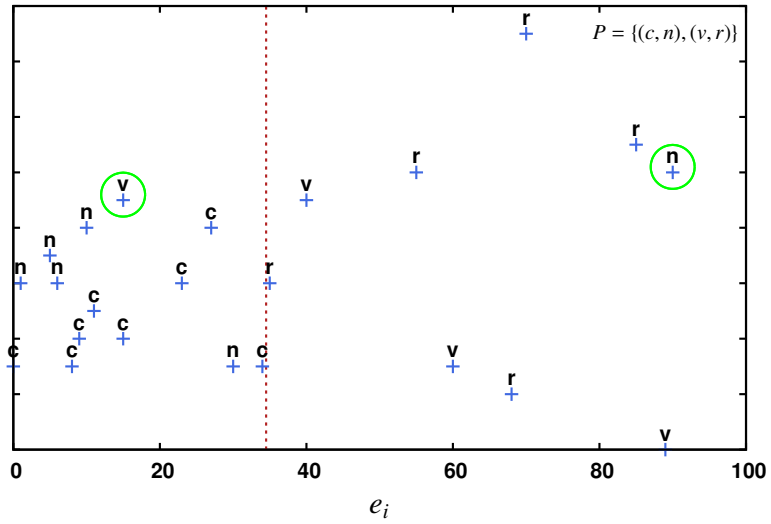


Figura 3.8: En rojo, la línea que mejor parte los valores de la variable simbólica en el eje e_i. En verde, los valores desobedientes. Nótese que si se mueve la línea roja, la desobediencia ya no es mínima. P indica la partición de la variable simbólica que genera la línea roja. Nota: Los valores de la variable simbólica no se distribuyen sobre el eje Y como se muestra en la gráfica, se hizo solo para fines explicativos.

Una vez calculada la desobediencia para cada partición y cada eje, se selecciona el par $(partición, eje)$ cuya desobediencia sea menor y dicha variable es desplegada en ese eje.

Con esto se puede graficar una variable simbólica por cada eje, e incluso, si hay ejes libres seleccionar uno para la variable simbólica. Aquellas variables simbólicas que no se lograron ajustar se buscarán graficar mediante color o forma.

3.7. Algoritmo usando MARS (Algoritmo MARS)

Este algoritmo es similar al anterior, sin embargo, en este no se utilizan mínimos cuadrados, sino un algoritmo que permite tener una mejor aproximación a los datos llamado *Multivariate Adaptive Regression Splines* (MARS) propuesto por Jerome H. Friedman en 1991 [32].

La idea detrás de MARS es dividir la variable independiente en sub-regiones

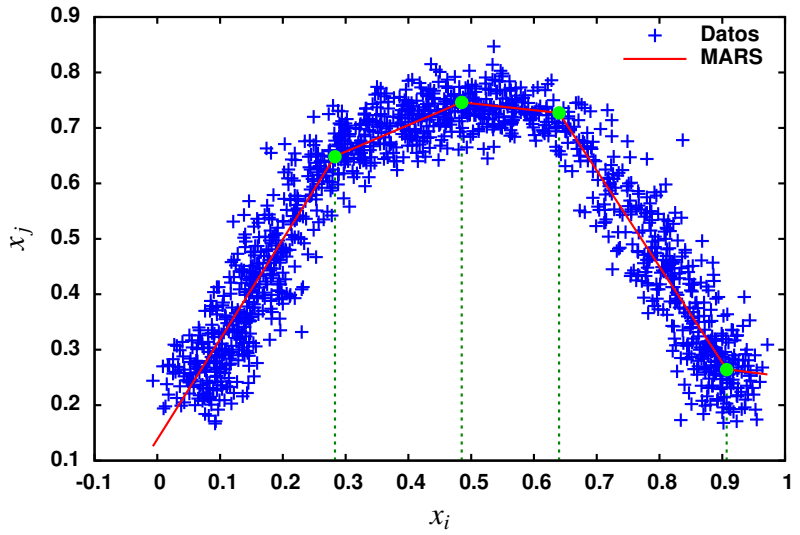


Figura 3.9: MARS. En rojo, la función que mejor aproxima los datos. En verde, los *knots* que dividen a x_i en sub-regiones.

como se ve en la figura 3.9. Así, cada sub-región estará definida por una línea recta diferente. Estas ecuaciones llamadas funciones base permiten relacionar la variable independiente con la variable dependiente y están definidas como sigue:

$$B_m^+ = \begin{cases} (t-x)^q, si\ x < t, \\ 0,\ en\ otro\ caso \end{cases}$$
$$B_m^- = \begin{cases} (x-t)^q, si\ x \geq t, \\ 0,\ en\ otro\ caso \end{cases} \tag{3.6}$$

donde q es el grado de la función base y t un punto de inflexión ($knot$).

El modelo final de MARS tiene la siguiente forma:

$$f(x) = a_0 + \sum_{m=1}^{M} a_m B_m(x). \tag{3.7}$$

donde x es la variable independiente, a_0 el coeficiente del termino constante, M el

número total de funciones base, B_m y a_m es la m-esima función base y su coeficiente respectivamente.

MARS tiene dos etapas:

- *Forward Stepwise*: Las funciones base de la ecuación 3.7 son definidas, sin embargo, por cuestiones de rendimiento suelen introducirse muchas funciones base lo que puede generar un problema de sobreajuste.
- *Backwards Stepwise*: Las funciones base redundantes que se agregaron en el paso anterior son eliminadas usando la validación cruzada generalizada (*Generalized Cross-Validation, GCV*).

El GCV está dado por:

$$GCV = \frac{\frac{1}{N}\sum_{i=1}^{N}[y_i - f(x_i)]^2}{[1 - \frac{C(M)}{N}]^2}. \tag{3.8}$$

donde N es el número de datos, M es el número de funciones base (sub-regiones), $C(M)$ es una función de penalidad que se incrementa según el número de funciones base y se define como:

$$C(M) = (M + 1) + \sigma M. \tag{3.9}$$

donde σ es un valor de penalidad.

Una vez terminado MARS, se va a analizar cada sub-región de la misma forma que en el algoritmo LS, es decir, determinar si en dicha sub-región la recta que ajusta los datos es creciente, decreciente o constante y se calcula el ECM global.

Ahora bien, para determinar si dos variables tienen un ajuste, se cuentan cuantos valores hay en las regiones crecientes y cuantos en las decrecientes y el valor máximo de estos determina el tipo de ajuste de estas variables. Al menos μ valores deben caer dentro de la franja.

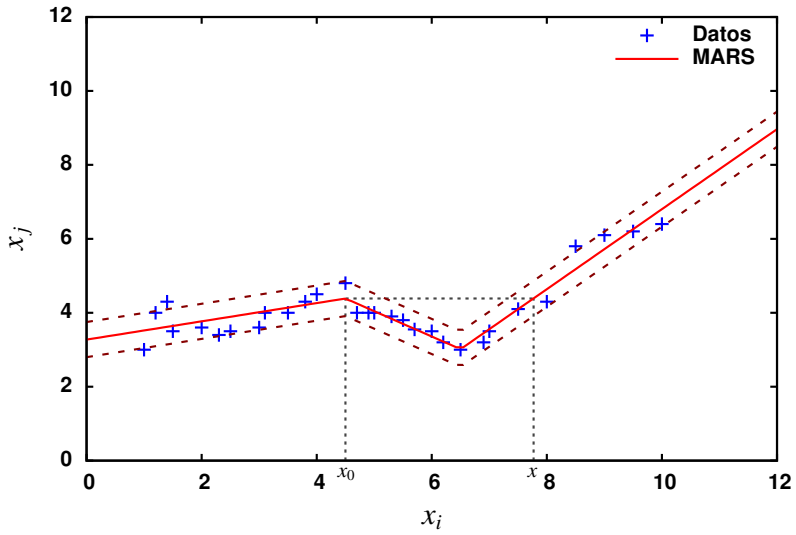


Figura 3.10: Hay tres sub-regiones, dos crecientes y una decreciente. Se contabilizan los valores que están entre x_0 y x y si es menor a un cierto umbral entonces descartamos estos valores para que se tenga un comportamiento monótono creciente. El punto x_0 es conocido al ser un *knot*, sin embargo, x debe ser calculado mediante la función inversa del segmento de recta correspondiente.

Por ejemplo, supóngase que las variables x_1 y x_2 tienen probabilidad de ser monótonas crecientes, esto es, hay más valores en las regiones crecientes que en las decrecientes. Para determinar si efectivamente pueden tener este ajuste, se contabilizan todos los valores de las sub-regiones decrecientes así como algunos segmentos de sub-regiones crecientes (ver figura 3.10) y si el total de estos valores es menor a un cierto umbral λ (se usó $\lambda = 10\,\%$ del total de los datos) y además al menos μ de los valores caen dentro de la franja entonces las dos variables tienen un ajuste. Para el caso de un ajuste decreciente el análisis es similar, salvo que ahora se contabilizan las sub-regiones crecientes. Si solo hay regiones crecientes (o decrecientes) únicamente es necesario verificar que al menos μ valores caen dentro de la franja. De igual forma que el algoritmo anterior, se toma el ECM como criterio de desempate cuando varias variables se ajustan a una pero entre ellas no hay ajuste.

Terminando este paso, el resto del algoritmo es igual al descrito en la sección anterior. En resumen el algoritmo sigue los siguientes pasos generales:

1. Sea C el conjunto de datos de n variables. Poner todas las variables numéricas en un conjunto A y las simbólicas en un conjunto B, es decir:

$$A = \{x_i \mid x_i \textit{ es una variable numérica},\ i \in [1, ..., k]\}$$
$$B = \{y_i \mid y_i \textit{ es una variable simbólica},\ i \in [k+1, ..., n]\}$$

2. Para cada x_i y x_j en A, $i < j$, se aplica el algoritmo de MARS y se calcula su ECM.

3. Se contabilizan los valores que caen en las regiones crecientes y decrecientes. Se toma el valor máximo y ese será el tipo de ajuste probable de las dos variables.

4. De la función resultante, se calcula una franja y se determina si el porcentaje de los puntos que caen dentro de ella es igual o mayor a un umbral μ.

5. En caso afirmativo, se verifica que los valores que caen en las sub-regiones que son contrarias al tipo de ajuste buscado no sobrepase un umbral λ. Este paso se omite si solo hay sub-regiones crecientes o decreciente.

6. Cada par de variables son agrupadas de acuerdo al tipo de ajuste que tuvieron.

7. Se determina el acomodo de cada par de variables, buscando que la mayor cantidad de éstas quede en un solo eje.

8. A cada eje se le asigna una puntuación y aquellos con la puntuación más alta son los candidatos a ser desplegados (bondad del eje).

9. Las variables numéricas sobrantes se tratarán de ajustar en alguno de los ejes resultantes buscando si hay una partición en el eje donde la variable tenga un buen encaje. Si no existe dicha partición en ningún eje la variable es colocada en un eje independiente.

10. Sea la variable simbólica $y_i \in B\ \forall i \in [k+1, ..., n]$. Analizar la frecuencia de los valores diferentes de dicha variable y descartar los valores que estén por debajo de un umbral δ.

11. Para cada $y_i \in B$ se particionan sus valores en dos o tres conjuntos disjuntos no triviales.

12. Sea e_i un eje. Se divide el eje en dos o tres partes, tratando de que cada elemento de la partición de y_i quede en un segmento de la partición del eje. Habrá elementos que caigan en un intervalo incorrecto del eje, estos elementos se llamaran "desobedientes" (ver figura 3.8).

13. Se selecciona el par $(partición, eje)$ cuya desobediencia sea menor y la variable será desplegada en ese eje.

14. Generar la visualización.

Como se observa, la única diferencia entre los dos algoritmos es el método usado en el ajuste de las dos variables. Más adelante se hace una comparativa de los resultados obtenidos de estos dos algoritmos.

3.8. Algoritmo para generar la agrupación de variables

A continuación se describe el algoritmo para generar los grupos de variables. En el algoritmo 1 se ve el pseudocódigo. Para generar los grupos se requiere tener una lista de las variables que tuvieron un ajuste mediante el algoritmo LS o el algoritmo MARS.

Posteriormente se debe analizar cada par de variables y tratar de que la mayoría quede en un solo grupo. Para esto el nuevo par de variables (x_i, x_j) que se busca colocar en algún grupo, se deben analizar de manera independiente, es decir, x_i se analiza si puede colocarse en algún grupo ya creado, para esto hay que verificar si x_i se ajusta con cada una de las variables del grupo. En caso afirmativo esta variable es colocada en el grupo que se está analizando, de lo contrario se analiza con el siguiente grupo y si en ningún grupo puede quedar se crea uno nuevo y x_i se coloca en ese. Lo mismo se hace para x_j. Después de agregar una variable a algún grupo ya existente se ordena de mayor a menor cardinalidad con la finalidad de que la mayoría de las variables queden en un solo grupo. Este procedimiento lo efectúa la función *Buscar* del pseudocódigo mostrado en el algoritmo 1.

Al final se tendrá una matriz bidimensional donde cada fila es un grupo de variables. Cabe mencionar que si ninguna de las variables se ajustó monotónicamente, entonces cada grupo tendrá únicamente una variable.

Algoritmo 1 Genera los grupos de variables que irán en cada eje.

Require:

global $ajuste$: Lista donde cada entrada es un par de variables que tuvieron ajuste monotónico.
global $total$: Numero de elementos de la variable $ajuste$.
global $grupos$: Matriz bidimensional.

```
1: Agrupa()
2: for i ← 0; i < total; i++ do        ▷ Para cada par de variables que se ajustaron
3:     if grupos == NULL then
4:         grupos[0][0] ← ajuste[i][0]
5:         grupos[0][1] ← ajuste[i][1]
6:     else
7:         grupos ← Buscar(ajuste[i])        ▷ Coloca ajuste[i] en el primer grupo donde pueda ir
8:     end if
9: end for
10: grupos ← AñadeVariablesFaltantes()        ▷ Variables que no se ajustaron van solas.
11: return
```

Capítulo 4

Implementación

En esta sección se describe la implementación del sistema que hace uso del método descrito en el capítulo anterior. Se divide en 3 secciones:

- **Sección 1**.- Se mencionan los requerimientos no funcionales, es decir, requerimientos que debe tener el sistema donde se instalará la aplicación.
- **Sección 2**.- Arquitectura del sistema.
- **Sección 3**.- Implementación. Se comenta a mayor detalle el algoritmo y como se resolvieron algunos problemas que se presentaron.

4.1. Requerimientos no funcionales

Los requerimientos no funcionales como ya se mencionó, son aquellos que debe tener el sistema donde se instalará la aplicación. Dichos requerimientos son los siguientes:

- Base de datos: Es necesaria una base de datos relacional donde se almacene la información a visualizar. En este caso se hace uso de MySQL 5.5.

- Servidor Web: Puede ser cualquier servidor web que permita la ejecución de código en PHP, por ejemplo Apache HTTP Server o Internet Information Services (IIS). Para las pruebas del sistema se utilizó Apache HTTP Server 2.2.
- PHP 5: Lenguaje de programación interpretado, el cual permite la generación de páginas web dinámicas. Se utilizó la versión 5.4 para las pruebas.
- Para el uso de MARS, se requiere la biblioteca Orange de Biolab[1].
- Un navegador web con soporte de WebGL.

4.2. Arquitectura del sistema

El prototipo se desarrolló como una aplicación web con una arquitectura MVC (Modelo-Vista-Controlador) que permite dividir todo el sistema en capas las cuáles son:

- Modelo: En esta capa se encapsula todo lo referente al sistema de base de datos así como la lógica de negocio.
- Vista: Es la encargada de dar formato al modelo para ser presentado al usuario, usualmente se denomina interfaz de usuario.
- Controlador: Es la capa encargada de recibir las peticiones del usuario o de capturar algún evento y solicitar peticiones al modelo e incluso a la vista.

Se incluyeron también las herramientas JQuery, WebGL y Three.js para tener un desarrollo ágil del prototipo.

4.2.1. JQuery

Es una biblioteca en JavaScript, creada por John Resig la cual permite simplificar la manera de interactuar con elementos en HTML. Posee funciones para

[1]Es una biblioteca de código abierto para la visualización y análisis de datos. Se puede obtener en http://orange.biolab.si/

la manipulación del DOM de HTML, manejo de eventos, animaciones, Ajax entre muchas cosas más. Al poseer un sistema de plugins se pueden agregar nuevas funcionalidades. Actualmente se cuenta con un repositorio de plugins bastante grande, desde menús, animaciones, elementos para la interfaz de usuario, etc.

Para más información consultar http://www.jquery.com/.

4.2.2. WebGL

WebGL es una API[2] para el manejo de gráficos 3D basado en OpenGL ES 2.0 mediante un navegador web lo que permite tener aceleración de hardware 3D [33]. Gracias a esto se pueden crear nuevas aplicaciones basadas en la web que antes eran exclusivas de escritorio, como por ejemplo los videojuegos 3D.

Actualmente los únicos navegadores que soportan esta API son Safari, Chrome, Firefox, Opera e Internet Explorer 11.

El prototipo expuesto en este trabajo hace uso de esta tecnología a través de la biblioteca 3D llamada "three.js" para generar las visualizaciones por lo cual este prototipo no es compatible con aquellos navegadores que no soporten WebGL.

Más información en http://www.khronos.org/webgl/.

4.2.3. Three.js

Three.js es una biblioteca de código abierto en JavaScript que permite mostrar y manipular gráficas en 3D sobre un navegador web.

Algunas de sus características son:

- Renderizado mediante canvas, gráficas vectoriales (SVG) o WebGL.
- Permite agregar y eliminar objetos de una escena durante la ejecución.

[2]Interfaz de programación de aplicaciones, es un conjunto de funciones y procedimientos que ofrece una biblioteca para ser usado por otro software.

- Manejo de diferentes tipos de luces, ambiental, direccional, etc. con lo cual se puede tener sombras.
- Uso de materiales como *lambert*, *phong*, texturas y más.
- Manejo de partículas, huesos, *sprites*, etc.
- Animaciones.
- Implementa dos tipos de cámara, la perspectiva y la ortográfica.

Se puede descargar de http://threejs.org/ así como consultar su documentación.

4.3. Implementación

En esta sección se detalla más a fondo la implementación de ambos algoritmos, comenzando por el algoritmo LS.

4.3.1. Algoritmo LS

Como ya se mencionó, el algoritmo comienza ajustando un par de variables numéricas a una recta haciendo uso del método denominado "mínimos cuadrados". Para determinar si dos variables tienen un buen ajuste, primero se debe determinar si la mayoría de los valores caen dentro de una franja, es decir, que a lo más 10 % del total de los datos quede fuera de dicha franja, de lo contrario, las variables no se ajustaron. Recordando que el ancho de la franja está dada por la ecuación 3.3.

Ahora bien, si las dos variables pasan este primer filtro, se debe ahora determinar si el comportamiento general es monótono creciente, decreciente o si alguna variable es constante. Para esto se analiza el valor de la pendiente de la recta resultante, si es positiva será monótona creciente, si es negativa decreciente y si es constante o casi constante se analiza si alguna de las dos variables es constante o casi constante.

Con esto, todos los pares de variables donde al menos μ de los puntos caigan dentro de la franja y cuya pendiente sea positiva son agrupadas. Otro grupo se

forma para las que tengan pendiente negativa. Después cada grupo es analizado para determinar si las variables que lo conforman pueden ir todas en un mismo eje, de lo contrario se van creando nuevos grupos.

Al final de este procedimiento se van a tener varios grupos, cada grupo será un eje. Se calcula la bondad de cada uno como se describió en el capítulo anterior y se ordenan de mayor a menor.

Si todos los ejes resultantes tienen más de una variable, es decir, de dos en adelante entonces no hay variables numéricas sobrantes y se prosigue al análisis de las variables simbólicas. Esto en general no va a suceder, habrá ejes con una sola variable.

Aquellos ejes que tengan solo una variable se van a tratar de ajustar a aquellos que tengan más de una variable. Si todos tienen una variable, entre ellos se tratará de ver si se pueden fusionar dos o más ejes. Para esto, se va a tomar un eje y se va a dividir en dos, tres y cuatro segmentos como máximo (un mayor número puede resultar confuso en la visualización para el usuario). Para generar esta división, se ordenan los valores del eje y se descartan los valores repetidos. Luego, se recorre esta lista de valores descartando los extremos y en cada valor se determina si genera una partición. Al detectar una, se toman los extremos izquierdo y derecho como si fueran un nuevo conjunto de datos y se busca en cada uno de ellos si hay una partición. Con esto se podrán generar 2, 3 y 4 particiones. Ahora bien, los valores de la variable que se busca ajustar en ese eje se colocan en cada segmento como se observa en la figura 3.7. Para cada segmento se va a determinar el rango de los valores descartando valores atípicos (*outliers*). Esto se logra mediante el cálculo del *boxplot* (figura 4.1).

Solo se va a permitir que haya β valores atípicos (desobedientes), de lo contrario se considera que no se parte bien sobre el eje. En caso de que una variable tenga un encaje perfecto sobre un eje, es decir, la intersección de los rangos de cada partición es vacía, entonces se termina el análisis y se coloca en dicho eje. Este procedimiento se aplica con cada eje que tenga una sola variable. Cabe mencionar que las particiones no necesariamente tienen que ser iguales, pues se busca la mejor partición donde la variable mejor encaje, es decir, haya el menor número de valores desobedientes.

Con esto termina la primer parte del algoritmo, es decir, el análisis de las variables numéricas. A continuación se detalla el procedimiento de análisis de las variables simbólicas.

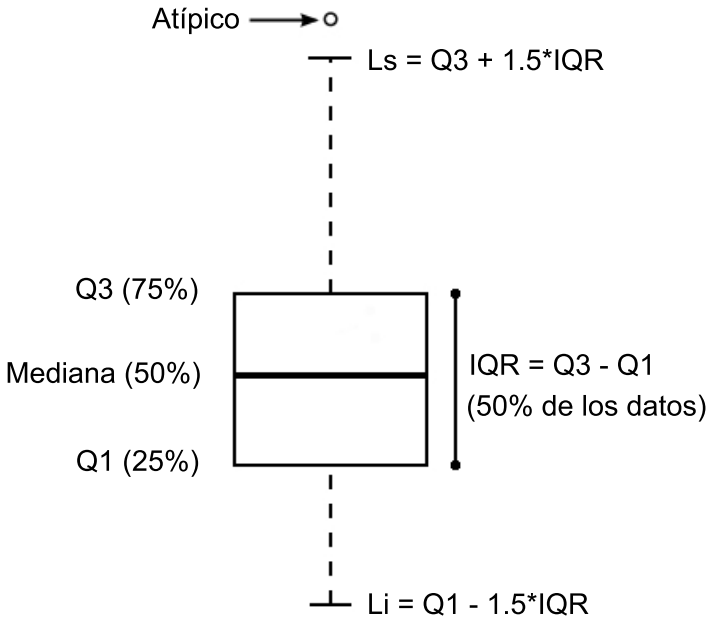


Figura 4.1: Diagrama de caja o *boxplot*

Sea $y_i \in B \; \forall i \in [k+1, ..., n]$. Analizar la frecuencia de los valores diferentes, esto para descartar aquellos valores que estén por debajo de un umbral δ. Se va a tratar de ajustar y_i en alguno de los ejes resultantes del análisis de las variables numéricas. Para esto se divide cada eje en dos y tres segmentos (más puede resultar confuso para el usuario) y se busca si existe una partición de los valores de y_i que encaje en la división del eje. Si existe una partición P de y_i tal que:

$$\bigcap P_i = \emptyset \; \forall i \in P. \tag{4.1}$$

se dice que y_i tiene un encaje perfecto en el eje. De ser el caso, el análisis termina y la variable es desplegada en dicho eje. En general, esto no va a ocurrir, habrá valores desobedientes como se muestra en la figura 3.8. Se contabilizan estos valores y si el total es menor a β se considera que y_i se parte bien en el eje. Al final y_i se puede partir en varios ejes por lo que hay que determinar en cual tiene mejor encaje, para esto se consideran los siguientes criterios, en orden decreciente:

- Los valores desobedientes: Se selecciona el eje cuya desobediencia sea menor.

- Número de particiones: Se considera aquel eje donde y_i tuvo mayor número de segmentos (particiones).
- Número de variables del eje: y_i se coloca en el eje que tenga mayor número de variables.

Las propiedades anteriores son usadas en ese orden para determinar en qué eje se colocará a y_i. Si no se llega a un criterio de desempate se coloca en el que sea.

Con esto se logran graficar algunas variables simbólicas, sin embargo, es posible que aun queden otras. Para estas se buscarán graficar mediante el color y forma. Para el color solo se van a graficar aquellas variables simbólicas que no tengan más de 10 valores diferentes y para la forma se permite un máximo de tres valores diferentes. Si hay más variables simbólicas, estas no podrán ser graficadas, sin embargo, se debe permitir al usuario intercambiar entre estas variables para graficarlas mediante el color y forma siempre que cumplan las restricciones antes comentadas.

Hasta aquí se ha llegado al término del algoritmo. En el siguiente capítulo se comentan las pruebas y resultados que se tuvieron.

4.3.2. Algoritmo MARS

Consiste en aplicar MARS para obtener los *knots* que dividen a la variable independiente en sub-regiones para posteriormente analizar cada una de estas y determinar si hay algún comportamiento monótono. En las funciones base (ecuaciones 3.6) el parámetro q debe ser uno, pues se busca un ajuste mediante líneas.

Como ya se comentó, MARS tiene dos etapa, la *forward stepwise* y *backwards stepwise*. En esta última se lleva a cabo la poda de funciones base bajo el criterio del GCV que entre más pequeño mejor. Recordando su ecuación se tiene que:

$$GCV = \frac{\frac{1}{N}\sum_{i=1}^{N}[y_i - f(x_i)]^2}{[1 - \frac{C(M)}{N}]^2}. \tag{4.2}$$

de donde

$$C(M) = (M + 1) + \sigma M. \tag{4.3}$$

.

Ahora bien, σ es una valor de penalización de las funciones base. Así, valores grandes de σ reducen el número de *knots* por lo cual se produce un ajuste más suave, sin embargo, para lo expuesto en este trabajo se busca que el ajuste no sea tan suave (mayor precisión) por lo cual σ debe ser pequeño. Friedman y Silverman [32] recomiendan que este valor sea 2, por lo cual, se usa este valor.

Cuando se contabilizan los valores de sub-regiones para determinar el probable ajuste de un par de variables (una vez que se determinó que al menos μ de los puntos caen dentro de una franja), es decir, si la variable tiene más regiones crecientes y el número de valores de las regiones decrecientes no excede el 10 % del total de los datos, entonces estos valores se omiten (ver figura 3.10). Los mismos parámetros son tomados para cuando son decrecientes. Si la función resultante fue estrictamente creciente (o decreciente) este último paso se omite, es decir, solo hay que verificar que al menos μ de los puntos estén dentro de la franja.

Continuando en este punto, se procede a agrupar las variables en diferentes ejes y se analizan las variables sobrantes para determinar si es posible ajustarlas en algún eje reduciendo su precisión mediante un particionado de los datos y por último se analizan las variables simbólicas, todo esto se efectúa de la misma manera que en lo descrito en el algoritmo LS.

Capítulo 5

Pruebas y resultados

Las pruebas se efectuaron sobre diez conjuntos de datos, nueve con datos reales y uno con datos sintéticos.

5.1. Conjunto de datos sintéticos

Consta de 125 registros cuya descripción y rango de valores se muestra en la tabla 5.1. Simula una encuesta realizada a 125 estudiantes.

Los datos fueron generados de tal manera que tuvieran el comportamiento mostrado en la tabla 5.2 donde también se muestra el ECM de las dos variables y la desviación absoluta de la mediana para considerar que esas dos variables puedan graficarse juntas. Adicionalmente en la tabla 5.3 se muestra el comportamiento de las variables simbólicas. Cabe recordar, que para que una variable simbólica se ajuste en un eje, solo se permite un máximo de 10 % del total de datos de valores desobedientes.

El resultado del algoritmo LS determina que las variables 1, 2, 4, 9, 7 y 10 deben ir en un eje, 5 en otro eje y la 8 en otro, tal como se observa en la figura 5.1. Las variables 4, 7 y 10 se ajustan mediante un particionado de sus valores. En la tabla 5.4 se observan los resultados finales. Cabe mencionar que el máximo de valores atípicos permitidos es 13, pues redondeando el 10 % de 125 da ese valor. Como se observa

Atributo	Descripción	Tipo	Valores ó Rango
(0) Genero	Indica el género de la persona	Sim.	Mujer=M, Hombre=H
(1) Edad		Num.	[17, 52]
(2) Miembros	Número de miembros de la familia	Num.	[1, 6]
(3) Trabaja		Sim.	Si=1, No = 0
(4) Ingreso_men	Ingreso mensual	Num.	[0, 1800]
(5) Gasto_ocio	Gasto semanal en ocio	Num.	[0, 120]
(6) Lugar_resid	Lugar de residencia	Sim.	Hospitalet, BCN, Altres municipis, Baix Llobregat, Altres del Barcelones
(7) Medio_transp	Medio de transporte utilizado en el desplazamiento al centro de estudio.	Sim.	Bus, NA, Bici y otros, metro, moto, coche, tren
(8) Tiempo_viaje	Tiempo empleado en el viaje de ida al centro de estudio	Num.	[0, 90]
(9) Nota_acceso	Nota de acceso a la universidad	Num.	[5, 8.67]
(10) Asig_matric	Número de asignaturas matriculadas en el curso anterior	Num.	[6, 10]
(11) Asig_aprob	Número de asignaturas aprobadas del curso anterior	Num.	[2, 8]

Tabla 5.1: Descripción del conjunto de datos sintéticos. En negritas las variables simbólicas. Total de datos: 125.

Variables	Tipo	ECM	MAD
1,2	Monótonas decrecientes	0.733	1
1,4	Monótonas crecientes	253.423	200
1,5	Monótonas crecientes	28.283	20
1,9	Monótonas crecientes	0.371	0.61
1,10	Monótonas decrecientes	1.576	0
2,4	Monótonas decrecientes	271.618	200
2,5	Monótonas decrecientes	27.835	20
2,9	Monótonas decrecientes	0.300	0.61
2,10	Monótonas crecientes	1.350	0
4,5	Monótonas crecientes	28.338	20
4,9	Monótonas crecientes	0.349	0.61
4,10	Monótonas decrecientes	1.148	0
5,9	Monótonas crecientes	0.822	0.61
5,10	Monótonas decrecientes	1.995	0
9,10	Monótonas decrecientes	1.240	0

Tabla 5.2: Comportamiento de los datos sintéticos (variables numéricas). MAD se calcula sobre la variable dependiente (variable después de la coma).

todas aquellas parejas donde está presente la variable 10 el número de valores fuera de la franja es todo el conjunto, esto se debe a que el MAD de esta variable es cero porque la mediana es 6 y al hacer la resta con cada uno de los valores, aquellos que son 6 se hacen cero y al calcular la nueva mediana el valor resultante es cero y por lo tanto no hay margen de error. Por otro lado, de la tabla 5.4 se observa que las únicas parejas candidatas a ajustarse son (1,2), (1,9), (2,9) y (4,9) mediante el método de mínimos cuadrados. Ahora bien, la pareja (1,4) no puede ir junta, pues según la tabla 5.4 no se ajusta, sin embargo, fue posible graficar la variable 4 junto con la variable 1 mediante un particionado de sus valores (se detectó una partición de los valores de la variable 4 sobre los valores de la variable 1). No se detectaron variables que permanecen constantes.

Ahora bien, la variable 10 posee solo dos valores, 6 y 10 que están perfectamente separados en dos grupos, por lo cual, esta variable tiene probabilidad de ajustarse en algún eje disponible. Esta variable con respecto a la variable 1 tiene una partición perfecta, pues existe un x en la variable 1 de tal forma que todos los valores menores o iguales a dicha x le corresponde un valor de la variable 10 y para todos los valores mayores a x le corresponde el otro valor. Así, la variable 10 debe ser graficada con

Variables (Sim:[Vars. Num])	**¿Se particiona?**	**Num. de particiones**	**Valores desobedientes**
0:[1]	Si	2	5
0:[2]	Si	2	7
0:[4]	**Si**	**2**	**0**
0:[9]	Si	2	6
0:[5,8,10,11]	No	-	-
3:[Todas]	No	-	-
6:[Todas]	No	-	-
7:[1]	**Si**	**3**	**0**
7:[2]	Si	2	12
7:[4]	Si	2	1
7:[10]	Si	2	9
7:[3,5,8,9,11]	No	-	-

Tabla 5.3: Comportamiento de los datos sintéticos (variables simbólicas). En negritas se muestra la mejor partición.

la variable 1 pues no hay otra variable donde esta tenga un encaje perfecto.

Como se observa en la tabla 5.3 solo dos variables simbólicas se particionan sobre algún eje. La variable 0 se particiona sobre las variables 1,2, 4 y 9, sin embargo, con unas variables tiene mejor particionado que con otras. Con la variable 4, tiene un encaje perfecto, esto es, no hay valores desobedientes, cosa que con ninguna otra variable numérica sucede esto, por lo cual, la variable 0 debe ser graficada en el eje donde se localiza la variable 4. Sin embargo, en la figura 5.1 no se muestra la variable 0 y si la variable 4, esto se debe a que la variable 4 se está graficando mediante un particionado con lo cual no se analiza la variable 0 con la variable 4. La variable 7 tiene un encaje perfecto con la variable 1, pese a que con la variable 4 solo hay un valor desobediente. Así esta variable debe ser graficada con la variable 1. Las otras dos variable simbólicas restantes, se graficaran mediante el color y forma, si cumplen las restricciones mencionadas anteriormente.

En la figura 5.1 se observa el resultado de aplicar el algoritmo LS.

A continuación se comentan los resultados usando el algoritmo MARS. Únicamente se comentaran los resultados para las variables numéricas, debido a que el

Variables	**Outliers**	**Ajuste**
1,2	12	Si
1,4	63	No
1,5	61	No
1,9	9	Si
1,10	125	No
2,4	60	No
2,5	61	No
2,9	8	Si
2,10	125	No
4,5	61	No
4,9	9	Si
4,10	125	No
5,9	68	No
5,10	125	No
9,10	125	No

Tabla 5.4: Resultados del análisis de las variables numéricas del algoritmo LS para el conjunto de datos sintéticos. Total de registros: 125.

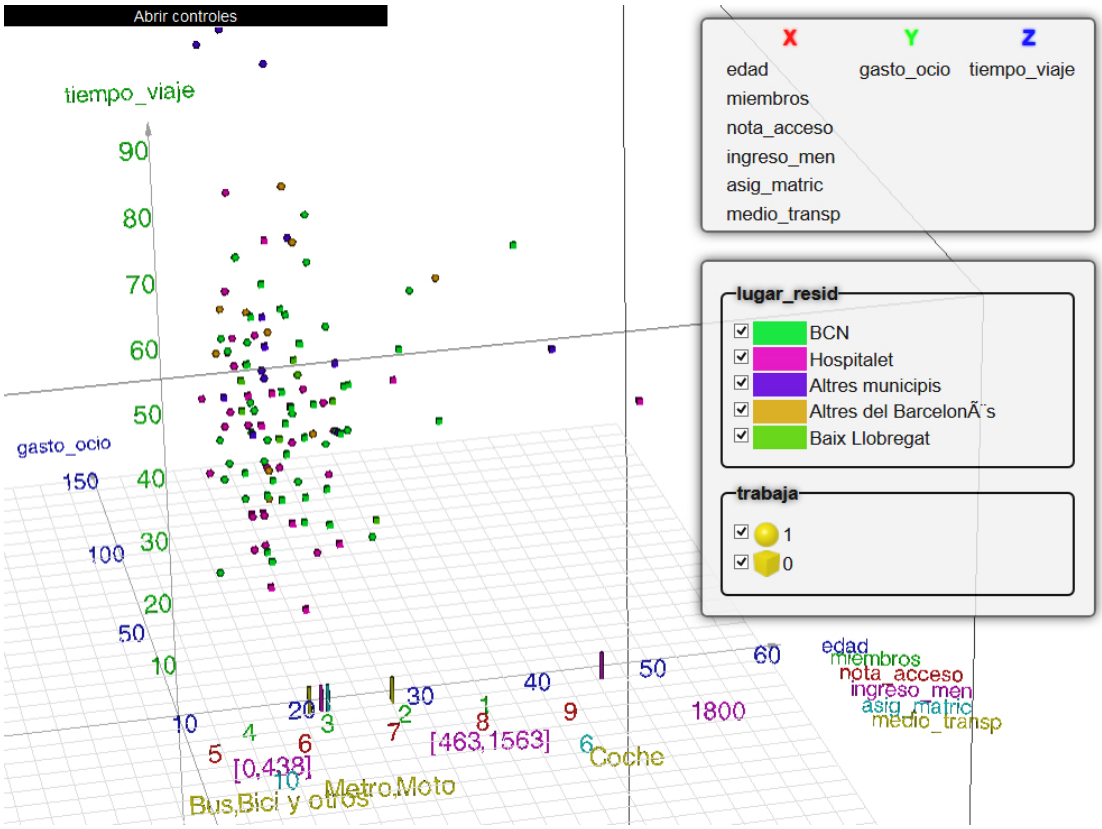


Figura 5.1: Resultado del algoritmo LS con el conjunto de datos sintéticos (ejemplo 5.1).

análisis de las variables simbólicas es el mismo en ambos algoritmos, recordando que la única diferencia entre ambos algoritmos es el método de ajuste usado en las variables numéricas.

En la tabla 5.5 se muestra el resultado del algoritmo MARS para todas las variables. Como se observa, el par (1,4) se ajusta con este algoritmo mediante MARS, mientras que en el anterior no sucedió, los demás par de variables que se ajustaron son los mismos que el algoritmo anterior, salvo que con este el número de *outliers* se redujo considerablemente. En la figura 5.2 se muestra una gráfica con las variables 1 y 4 así como las dos funciones de ajuste de ambos algoritmos.

A diferencia del algoritmo anterior, en este caso sí es posible graficar las variables 1 y 4 juntas de forma lineal, es decir, no mediante particionado, por lo tanto, las variables 1, 4 y 9 son candidatas a ir en un solo eje, la variable 5 en el segundo y en el tercer eje la variable 8. Nótese que las variables 2 y 4 no pueden ir juntas, pues no

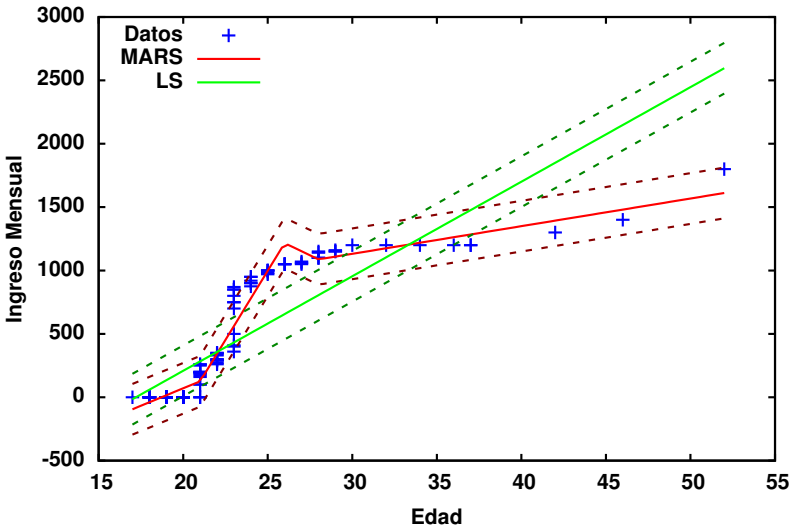


Figura 5.2: Mínimos cuadrados (verde) vs MARS (rojo). Variables 1 y 4 del conjunto de datos sintéticos (ejemplo 5.1).

tienen un ajuste con MARS, sin embargo, en la figura 5.3 se observa que estas dos variables se grafican juntas, esto se debe a que la variable 2 se particiona sobre la variable 4 con lo que las variables 1, 2, 4 y 9 pueden ir juntas. De igual modo que el algoritmo anterior, las variables 7 y 10 se grafican junto con la variable 1 por lo cual en un eje irán las variables 1, 2, 4, 7, 9 y 10. Como la variable 4 se está graficando de forma lineal es posible graficar la variable 0, pues existe una partición de esta sobre la variable 4 dando como resultado que en un eje vayan las variables 0, 1, 2, 4, 7, 9 y 10, en otro la variable 5 y por último en el tercer eje la variable 8.

En conclusión, el algoritmo MARS presenta un mejor ajuste de los datos, por lo cual la visualización resultante tiene un menor error, pues los valores de las etiquetas en los ejes no se posicionan de forma lineal como en el algoritmo LS, sino que están en función del resultado de MARS acercándose más al valor real. Con el algoritmo LS se lograron graficar 10 variables de 12 mientras que con el algoritmo MARS 11 de las 12 variables.

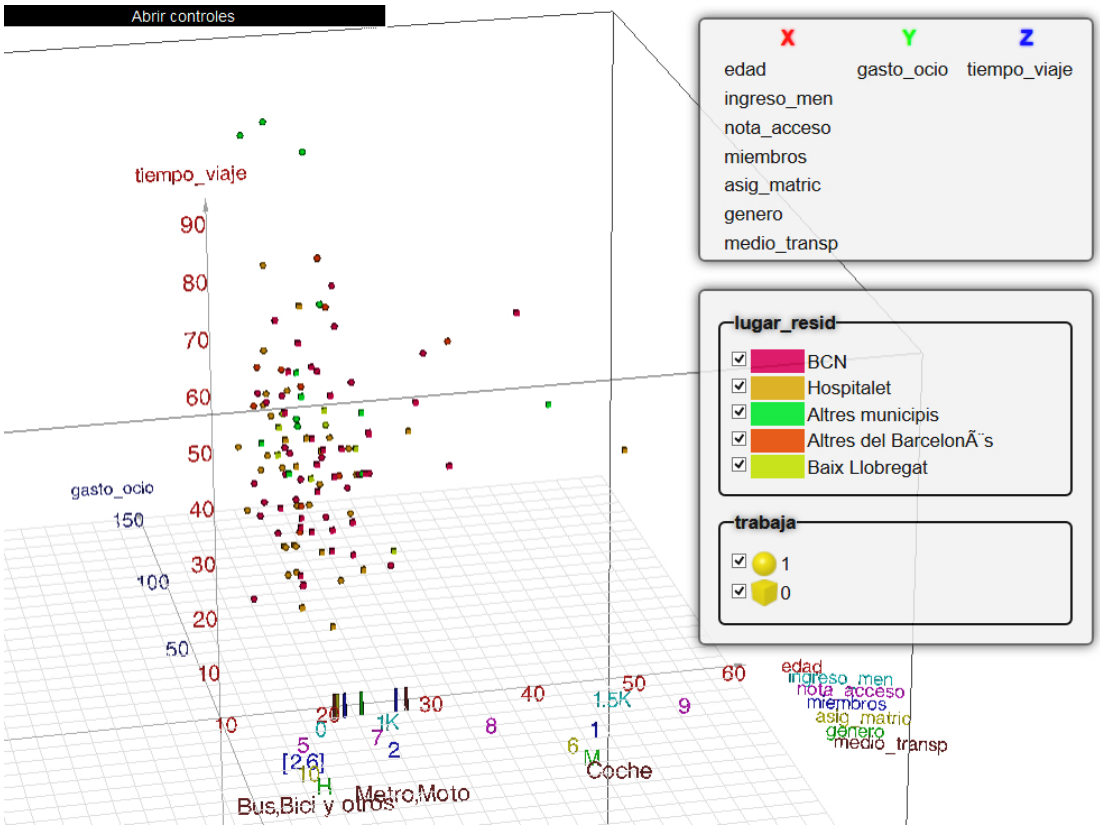


Figura 5.3: Resultado del algoritmo MARS con el conjunto de datos sintéticos (ejemplo 5.1).

5.2. Encuesta alumnos

Este conjunto de datos[1] es de una encuesta real realizada a 125 estudiantes cuyos atributos son los mismos que el conjunto de datos anterior, es decir, los mostrados en la tabla 5.2. Los resultados del algoritmo LS y MARS se muestran en la tabla 5.6. Como se observa, en general, en este caso, MARS presenta un peor ajuste comparado con el método de mínimos cuadrados, pues hay más *outliers*, esto se debe a que los datos están muy esparcidos y cuando esto sucede MARS entrega una recta, aunque no siempre es la mejor recta que ajusta los datos. En la figura 5.4 se observa el comportamiento de ambos métodos para las variables 1 y 2 y es claro como MARS tuvo un peor ajuste en este caso. Sin embargo, ambos métodos obtienen el mismo resultado final, ningún par de variables tiene un comportamiento monótono, esto significa que los datos están esparcidos, en otras palabras, cada atributo o variable

[1]Se obtuvo de http://hdl.handle.net/2445/16662 el 25-feb-2013

es independiente una de otra, por lo tanto, cada variable estará colocada en un eje y se muestran las tres primeras variables pues todos los ejes tienen la misma "bondad".

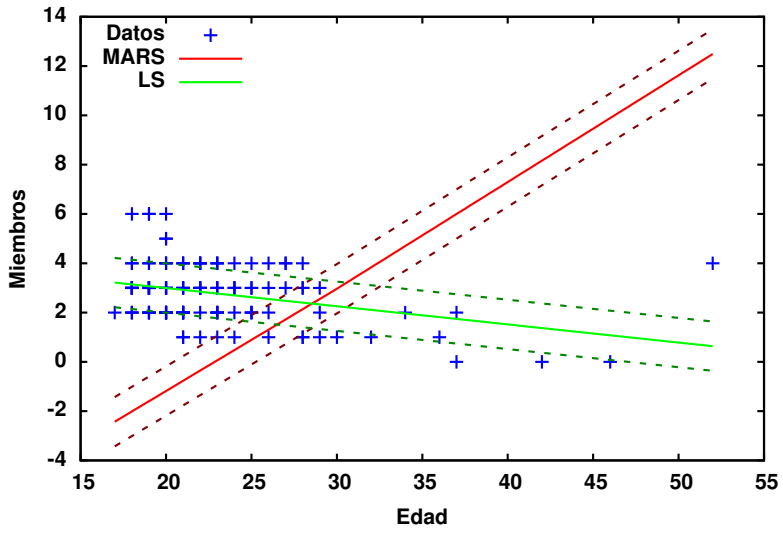


Figura 5.4: Comportamiento de las variables 1 y 2 en el conjunto de datos "Encuesta alumnos" (ejemplo 5.2). En rojo MARS y en verde mínimos cuadrados.

En lo que respecta a las variables simbólicas, solo la variable 3 logra ajustarse con la variable 4. Intuitivamente es lógico, pues la variable 4 representa el ingreso mensual que tiene el estudiante y la variable 3 si trabaja o no. Como se observa en la figura 5.5 todos aquellos estudiantes cuyo ingreso mensual es cero son porque no trabajan.

Como se pudo notar, en ambos algoritmos se lograron graficar 6 variables de 12 en una sola gráfica. En la figura 5.6 se muestra una comparativa de la visualización propuesta en este trabajo y un programa llamado SpotFire de TIBCO Software. Cabe mencionar que ambas visualizaciones muestran en un solo grafo 6 variables, sin embargo, la de SpotFire en los ejes solo muestra tres variables a diferencia de las cuatro que se muestran en este trabajo. La diferencia está en que SpotFire muestra una variable haciendo uso del tamaño. Esto en muchos casos puede generar confusión como se puede ver en el grafo de SpotFire, pues aunque solo se manejan dos tamaños (pequeño y grande como se ve en las etiquetas) la perspectiva de la propia visualización hace que se vean de muchos tamaños, razón por la cual se descartó en este trabajo.

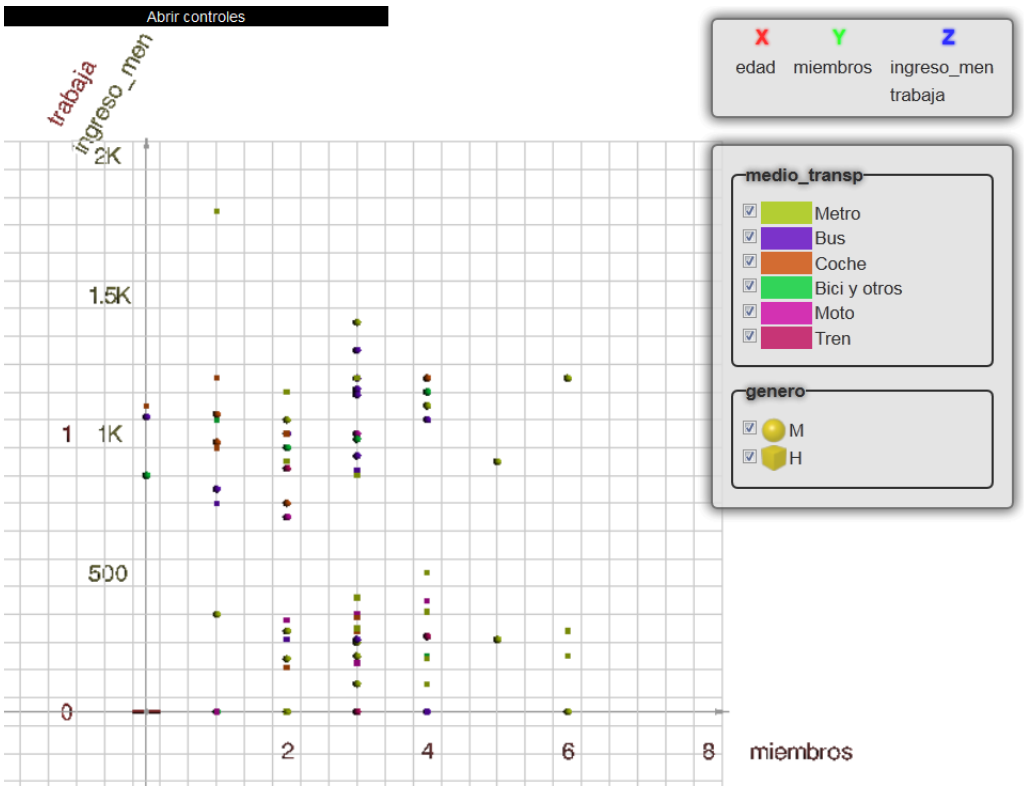


Figura 5.5: (Ejemplo 5.2). Resultado final del algoritmo LS y MARS para el conjunto de datos "Encuesta alumnos".

Variables	**Outliers**	**Ajuste**
1,2	4	Si
1,4	5	Si
1,5	71	No
1,9	0	Si
1,10	125	No
2,4	15	No
2,5	76	No
2,9	2	Si
2,10	125	No
4,5	71	No
4,9	9	Si
4,10	125	No
5,9	62	No
5,10	125	No
9,10	125	No

Tabla 5.5: Resultados del algoritmo MARS para el conjunto de datos sintéticos. Total de registros: 125.

		Outliers		**Ajuste**	
Variables	**MAD**	**LS**	**MARS**	**LS**	**MARS**
1,2	1	48	117	No	No
1,4	200	96	119	No	No
1,5	20	62	73	No	No
1,8	10	61	64	No	No
1,9	0.61	64	62	No	No
1,10	0	125	125	No	No
1,11	1	51	53	No	No
2,4	200	105	112	No	No
2,5	20	66	71	No	No
2,8	10	64	64	No	No
2,9	0.61	62	62	No	No
2,10	0	125	125	No	No
2,11	1	53	53	No	No
4,5	20	56	92	No	No
4,8	10	62	61	No	No
4,9	0.61	63	62	No	No
4,10	0	125	125	No	No
4,11	1	55	86	No	No
5,8	10	63	64	No	No
5,9	0.61	61	62	No	No
5,10	0	125	125	No	No
5,11	1	55	53	No	No
8,9	0.61	54	58	No	No
8,10	0	125	125	No	No
8,11	1	52	53	No	No
9,10	0	125	125	No	No
9,11	1	50	53	No	No
10,11	1	52	52	No	No

Tabla 5.6: Resultados del algoritmo LS y MARS con el conjunto de datos "Encuesta alumnos". Total de registros: 125.

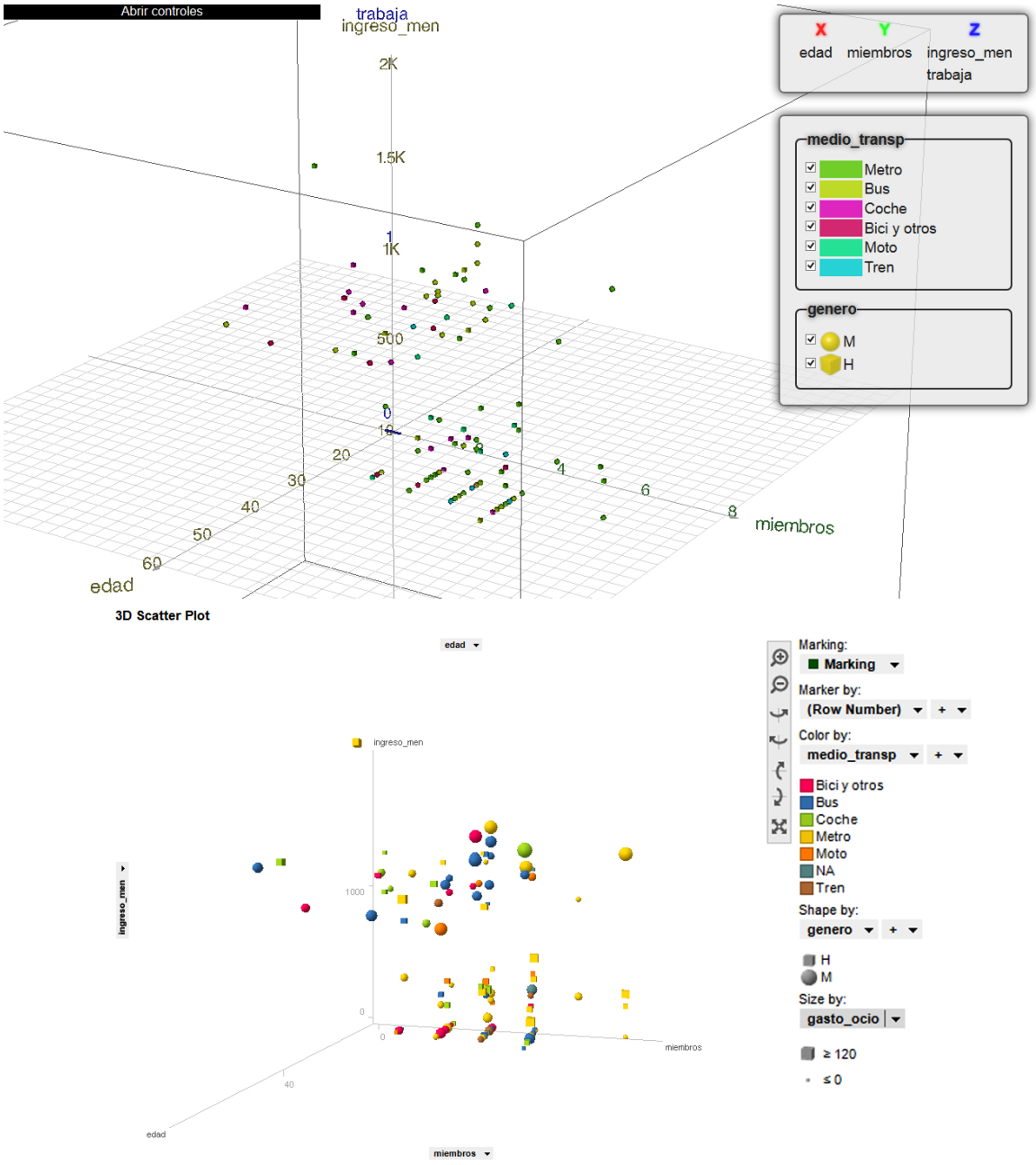


Figura 5.6: En la imagen superior, la visualización propuesta en este trabajo, en la inferior, una visualización del mismo conjunto de datos "Encuesta alumnos" usando un software externo (SpotFire).

5.3. Datos de reconocimiento de vino (*Wine Recognition Data*)

Este conjunto llamado *Wine Recognition Data*[2] consta de 178 registros y 13 atributos, 12 numéricos y uno simbólico. En la tabla 5.7 se muestran sus atributos así como su rango de valores.

Variable	Rango/Valores
(0) Alcohol_class	1, 2 ó 3
(1) Malic acid	[11.03, 14.83]
(2) Ash	[0.74, 5.80]
(3) Alcalinity of ash	[1.36, 3.23]
(4) Magnesium	[10.6, 30]
(5) Total phenols	[70, 162]
(6) Flavanoids	[0.98, 3.88]
(7) Nonflavanoid phenols	[0.34, 5.08]
(8) Proanthocyanins	[0.13, 0.66]
(9) Color intensity	[0.41, 3.58]
(10) Hue	[1.28, 10]
(11) OD280/OD315 of diluted wines	[0.48, 1.71]
(12) Proline	[1.27, 4]

Tabla 5.7: Descripción del conjunto "datos de reconocimiento de vino". En negrita la variable simbólica. Total de registros: 178.

Los resultados de ambos algoritmos se muestran en la tabla 5.8. Como se puede observar, usando mínimos cuadrados hay un ajuste de las variables 6 y 7, mientras que, usando MARS no hay ajuste. En la figura 5.7 se observa el resultado de ambos métodos. Como ya se comentó, cuando MARS no logra encontrar *knots* devuelve una recta, la cual no necesariamente es la mejor recta que ajusta los datos, razón por la cual, en este caso usando mínimos cuadrados se obtiene mejor resultado.

En general, cuando MARS no detecta puntos de inflexión, el método de mínimos cuadrados da mejores resultados, porque devuelve la recta que mejor ajusta los datos.

[2]Tomado de la UCI *Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets/Wine 17-04-2013.*

		Outliers		Ajuste	
Variables	**MAD**	**LS**	**MARS**	**LS**	**MARS**
6,7	0.835	13	24	Si	No

Tabla 5.8: Resultado del algoritmo LS y MARS para el conjunto de datos "Reconocimiento de vino". Solo se muestran las dos variables que tuvieron un ajuste con algún método, cualquier otra pareja no se ajustó. Total de registros: 178.

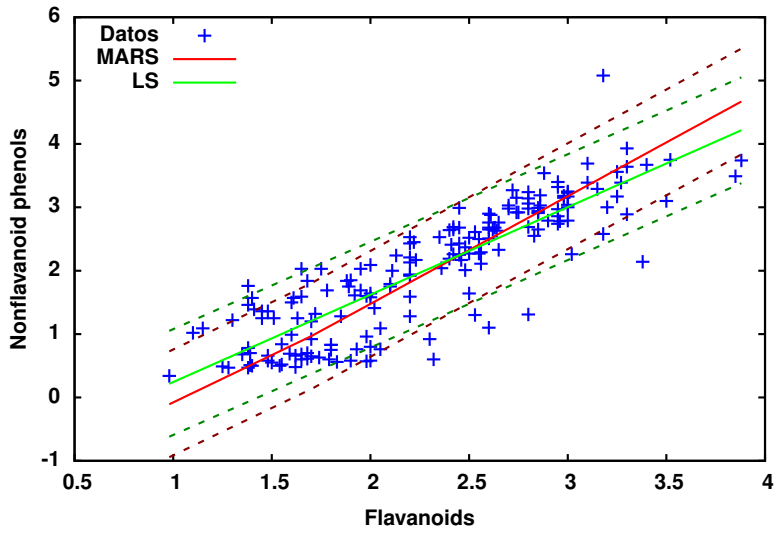


Figura 5.7: Ajuste de las variables 6 y 7 del conjunto de datos "reconocimiento de vino" (ejemplo 5.3) donde LS tiene mejor ajuste que MARS.

No se detectaron variables constantes. La variable simbólica no se logra ajustar mediante un particionado, sin embargo, se gráfica mediante el color debido a que únicamente posee tres valores diferentes.

En conclusión, usando mínimos cuadrados se graficaron 5 de 13 variables mientras que con MARS solo 4. En la figura 5.8 se muestra el resultado final de la visualización.

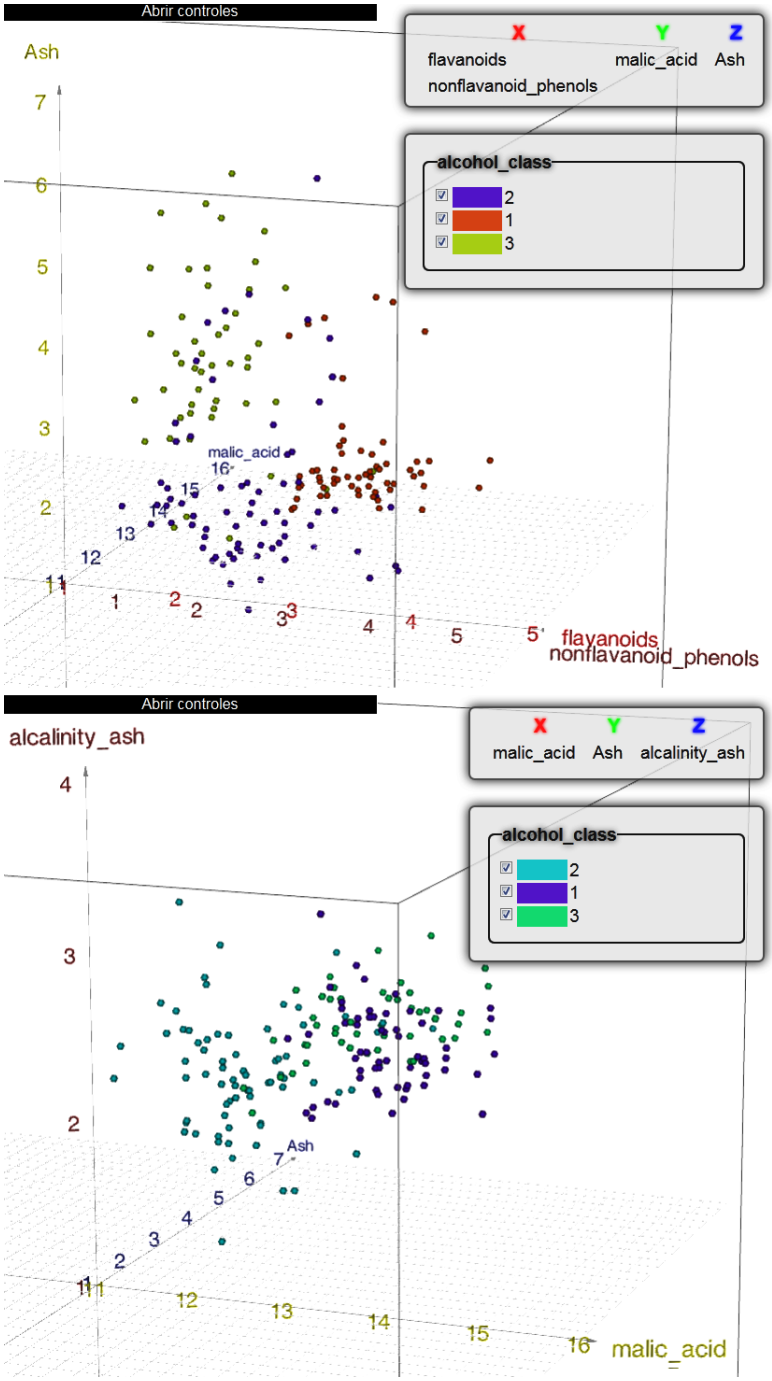


Figura 5.8: Resultado final de los algoritmos para el conjunto "Datos de reconocimiento de vino" (ejemplo 5.3). Arriba mínimos cuadrados, abajo MARS.

5.4. Datos de vivienda de Boston (*Boston Housing Data*)

Este conjunto llamado *Boston Housing Data*[3] consta de 506 registros y 14 atributos, 13 numéricos y uno simbólico cuya descripción se muestra en la tabla 5.13. Los datos corresponden a un censo de vivienda de 1970 de la ciudad de Boston.

En este conjunto de datos no se detectó ningún tipo de comportamiento monótono entre las variables, por lo cual, cada variable ira en un eje independiente, teniendo así 12 posibles ejes que el usuario puede seleccionar (ver figura 5.9). Sin embargo, hay una variable numérica que no fue considerada en los ejes, esta es la 4 (NOX). Esto se debe a que su variación es muy pequeña, el rango de la variable va de 0.385 a 0.871, variando en 0.486 que es menor al umbral tomado para determinar si las variables tienen probabilidad de ser constantes, el cual es de 0.5. La variable simbólica (CHAS) solo tiene dos valores diferentes, por lo cual, es graficada mediante el color. En la figura 5.10 se muestra el resultado de la visualización.

[3]Tomado de la biblioteca StatLib mantenida por la universidad de Carnegie Mellon.

Atributo	**Descripción**	**Tipo**	**Rango/Val.**
(0) CRIM	Tasa de criminalidad por ciudad	Num.	[0, 88.97]
(1) ZN	Proporción de suelo residencial dividido en porciones de más de 25000 pies cuadrados	Num.	[0, 100]
(2) INDUS	Proporción de acres de negocio por ciudad	Num.	[0.46, 27.74]
(3) CHAS	Indica si una área de tierra limita con el rio Charles River	Sim.	1 si limita, 0 en otro caso
(4) NOX	Concentración de ácido nítrico (partes por 10 millones)	Num.	[0.38, 0.87]
(5) RM	Número promedio de habitaciones por vivienda	Num.	[3.56, 8.78]
(6) AGE	Proporción de unidades ocupadas por sus propietarios construidas antes de 1940	Num.	[2.9, 100]
(7) DIS	Distancias ponderadas a cinco centros de empleo en Boston	Num.	[1.12, 12.13]
(8) RAD	Índice de accesibilidad a las autopistas radiales	Num.	[1, 24]
(9) TAX	Valor total de la tasa de impuesto a la propiedad por $10000	Num.	[187, 711]
(10) PTRATIO	Tasa alumno-maestro por ciudad	Num.	[12.6, 22]
(11) B	$1000(Bk-0.63)^2$ donde Bk es la proporción de negros por ciudad.	Num.	[0.32, 396.9]
(12) LSTAT	Porcentaje inferior del status de la población	Num.	[1.73, 37.97]
(13) MEDV	Valor medio de viviendas ocupadas por sus propietarios en $1000	Num.	[5, 50]

Tabla 5.9: Descripción del conjunto "Datos de vivienda de Boston". En negrita la variable simbólica. Total de registros: 506.

Figura 5.9: (Ejemplo 5.4). Posibles ejes a seleccionar del conjunto "Datos de vivienda de Boston".

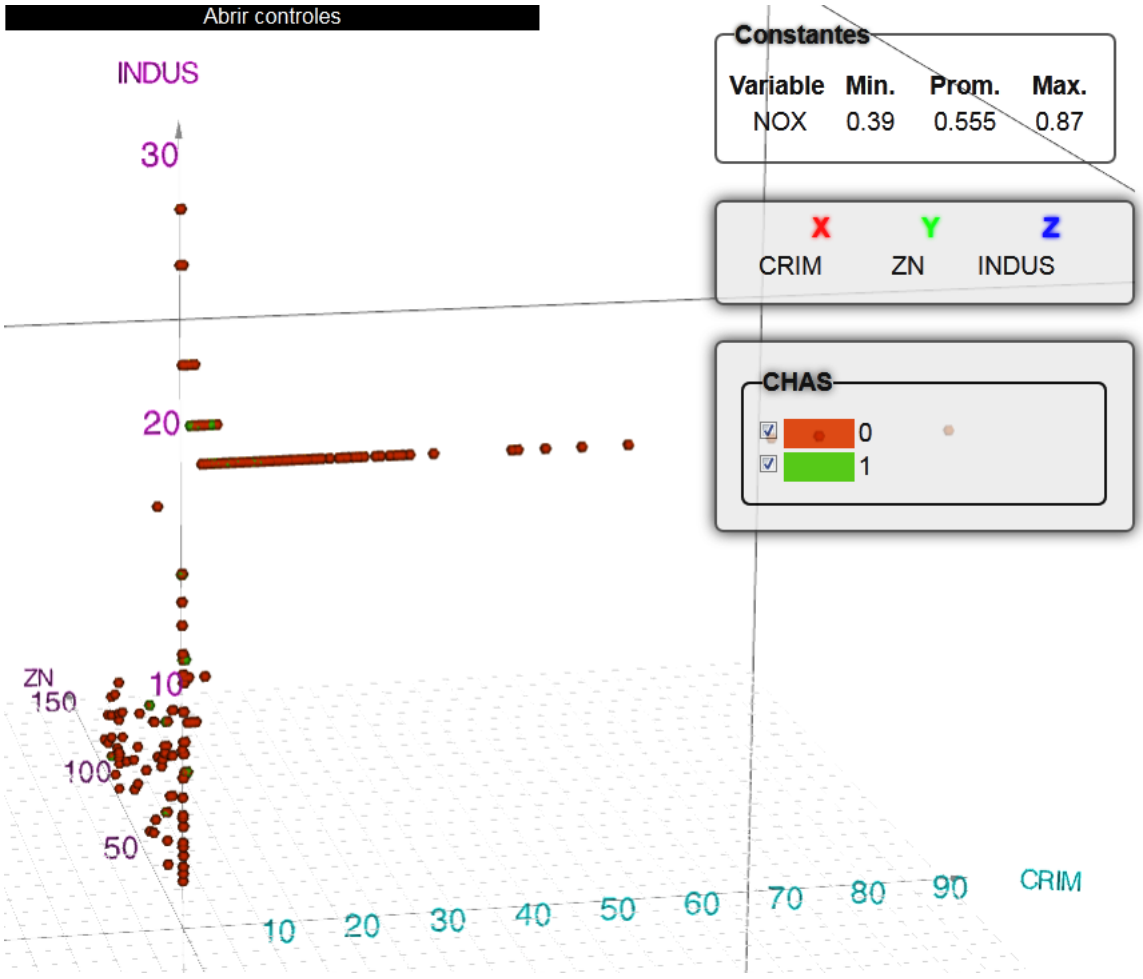


Figura 5.10: (Ejemplo 5.4). Resultado final del algoritmo LS y MARS para el conjunto "Datos de vivienda de Boston".

5.5. Factores determinantes de los salarios de la población en 1985

Conjunto de datos llamado *Determinants of Wages from the 1985 Current Population Survey*[4] él cual consta de 534 registros y 11 atributos, 4 numéricos y 7 simbólicos. Su descripción se muestra en la tabla 5.10.

Variable	**Descripción**	**Rango/Valores**
(0) Education	Número de años de educación	[2, 18]
(1) South	Indicador de la región sur	1=Persona vive en el sur, 0 en otro caso
(2) Sex	Género de la persona	1=Femenino, 0=Masculino
(3) Experience	Años de experiencia en el trabajo	[0, 55]
(4) Union	¿pertenece a un sindicato?	Si=1 ó No=0
(5) Wage	Salario (dólares por hora)	[1, 44.50]
(6) Age	Edad	[18, 64]
(7) Race	Raza	1=Otro, 2=Hispano, 3=Blanco
(8) Occupation	Ocupación	6 valores diferentes
(9) Sector	Sector	0=Otro, 1=Manufactura, 2=Construcción
(10) Marr	¿Es Casado?	Si=1 ó No=0

Tabla 5.10: Descripción del conjunto de datos "salarios de la población". En negritas las variables simbólicas. Total de datos: 534.

Ambos algoritmos dieron los mismos resultados en cuanto a agrupación de las variables, pues solo dos variables tienen un comportamiento monotónico, que son las variables 3 y 6. Su comportamiento es monótono creciente. En la figura 5.11 se observa el resultado de estas dos variables. Este conjunto de datos permite ver como ambos métodos (mínimos cuadrados y MARS) dan resultados muy similares, salvo que MARS es ligeramente mejor, aunque al final en la visualización es despreciable esta mejora, pues las etiquetas quedan prácticamente en el mismo lugar como se ve

[4]Tomado de http://lib.stat.cmu.edu/datasets/ 30-04-2013.

en la figura 5.12. Para las variables simbólicas no se detectó ninguna partición, por lo cual, solo se grafican dos usando color y forma.

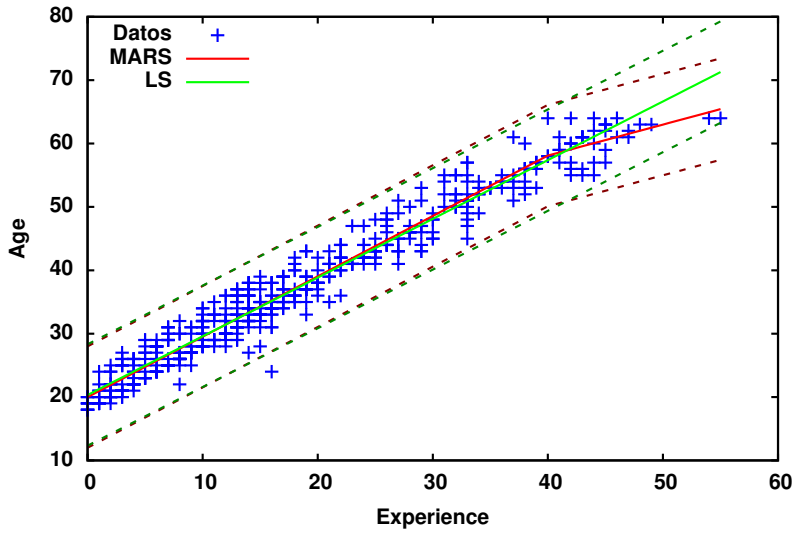


Figura 5.11: Resultado del ajuste de las variables 3 y 6 del conjunto de datos "salarios de la población" (Ejemplo 5.5).

Este conjunto de datos al tener pocas variables numéricas, todas lograron graficarse juntas, cosa que no pasa si se usa algún otro software como SpotFire, Tableau, Excel, etc. por lo que se logra obtener mayor información.

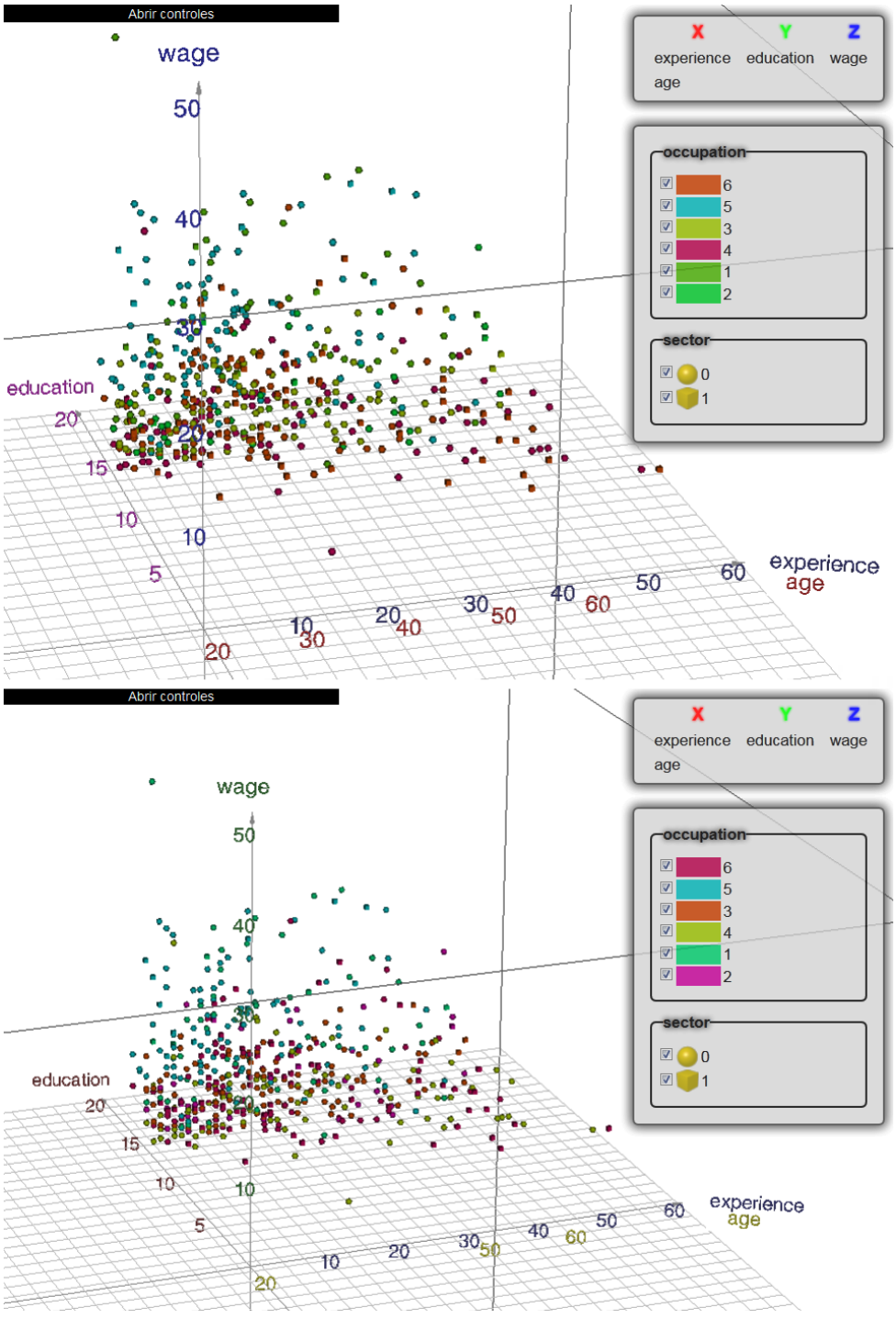


Figura 5.12: (Ejemplo 5.5). Resultado final del conjunto "salarios de la población". Arriba mínimos cuadrados y abajo MARS.

5.6. Aprobación de crédito (*Credit Approval*)

Este conjunto llamado *Credit Approval*[5] tiene 690 registros con 16 atributos, seis numéricos y diez simbólicos. No todos los registros fueron usados, algunos se omitieron debido a que el conjunto presenta valores faltantes los cuales se descartaron, por lo que únicamente se trabajó con 653 datos. En la tabla 5.11 se muestran los atributos así como su tipo y rango.

Variable	Tipo	Rango/Valores
(0) A1	Simbólica	{a, b}
(1) A2	Numérica	[14, 77]
(2) A3	Numérica	[0, 28]
(3) A4	Simbólica	{l, u, y}
(4) A5	Simbólica	{g, gg, p}
(5) A6	Simbólica	14 valores diferentes
(6) A7	Simbólica	9 valores diferentes
(7) A8	Numérica	[0, 28.5]
(8) A9	Simbólica	{f, t}
(9) A10	Simbólica	{f, t}
(10) A11	Numérica	[0, 67]
(11) A12	Simbólica	{f, t}
(12) A13	Simbólica	{g, p, s}
(13) A14	Simbólica	164 valores diferentes
(14) A15	Numérica	[0, 100 000]
(15) A16	Simbólica	{+, -}

Tabla 5.11: Descripción del conjunto de datos "Aprobación de crédito". En negritas las variables simbólicas. Total de registros: 653.

En este conjunto de datos no se encontraron comportamientos monótonos con ninguno de los dos métodos descritos, por lo que a cada variable numérica le corresponde un eje. Tampoco se encontraron particiones para las variables, tanto numéricas como simbólicas ni valores constantes. En la figura 5.13 se muestra la visualización final.

[5]Se puede obtener de http://archive.ics.uci.edu/ml/datasets/Credit+Approval 14-05-2013.

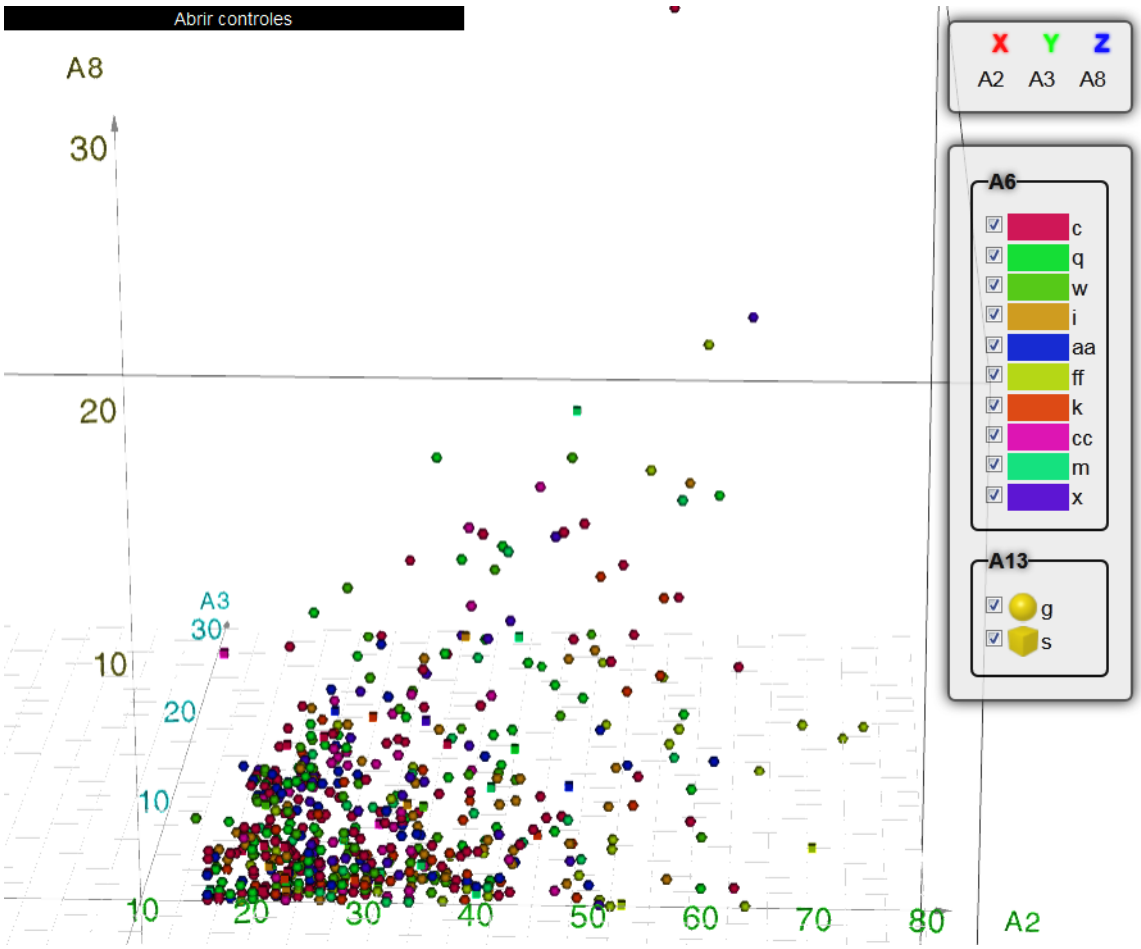


Figura 5.13: (Ejemplo 5.6). Resultado final del conjunto de datos"Aprobación de crédito". Ambos algoritmos dan el mismo resultado.

5.7. Dermatología (*Dermatology*)

Llamado *Dermatology*[6] es un conjunto de datos con 34 atributos, 33 numéricos y uno simbólico. Tiene 366 registros de los cuales se trabajó únicamente con 358 debido a valores faltantes en ocho registros. El la tabla 5.12 se muestran los atributos asi como sus valores o rango.

Atributo	**Tipo**	**Valores**	**Atributo**	**Tipo**	**Valores**
(0) Erythema	Num.	[0, 3]	(17) Hyperkeratosis	Num.	[0, 3]
(1) Scaling	Num.	[0, 3]	(18) Parakeratosis	Num.	[0, 3]
(2) Definite_borders	Num.	[0, 3]	(19) CRR	Num.	[0, 3]
(3) Itching	Num.	[0, 3]	(20) ERR	Num.	[0, 3]
(4) Koebner_ph	Num.	[0, 3]	(21) TSE	Num.	[0, 3]
(5) Poly_papules	Num.	[0, 3]	(22) Spongiform_p	Num.	[0, 3]
(6) Foll_papules	Num.	[0, 3]	(23) Munro_m	Num.	[0, 3]
(7) OMI	Num.	[0, 3]	(24) Focal_hyper	Num.	[0, 3]
(8) KEI	Num.	[0, 3]	(25) DGL	Num.	[0, 3]
(9) Scalp_inv	Num.	[0, 3]	(26) VDBL	Num.	[0, 3]
(10) Fam_his	Sim.	{0, 1}	(27) Spongiosis	Num.	[0, 3]
(11) Melanin_inc	Num.	[0, 3]	(28) SAR	Num.	[0, 3]
(12) Eosinophils	Num.	[0, 2]	(29) FHP	Num.	[0, 3]
(13) PNL	Num.	[0, 3]	(30) PP	Num.	[0, 3]
(14) FPD	Num.	[0, 3]	(31) IMI	Num.	[0, 3]
(15) Exocytosis	Num.	[0, 3]	(32) BLI	Num.	[0, 3]
(16) Acanthosis	Num.	[0, 3]	(33) Age	Num.	[0, 75]

Tabla 5.12: Descripción del conjunto de datos "Dermatología". En negrita la variable simbólica. Total de registros: 358.

Al igual que el conjunto anterior, no se detectaron comportamientos monótonos ni particionado, sin embargo, de los 34 atributos se lograron graficar siete, debido a que tres de estas variables son constantes. En este caso se puede graficar una variable más que usando un software como SpotFire que permite graficar hasta seis variables. En la figura 5.14 se muestra el resultado final de los dos métodos, tanto mínimos cuadrados como MARS (da el mismo resultado en ambos al no haber comportamientos monótonos).

[6]Descargado de http://archive.ics.uci.edu/ml/datasets/Dermatology 15-05-2013

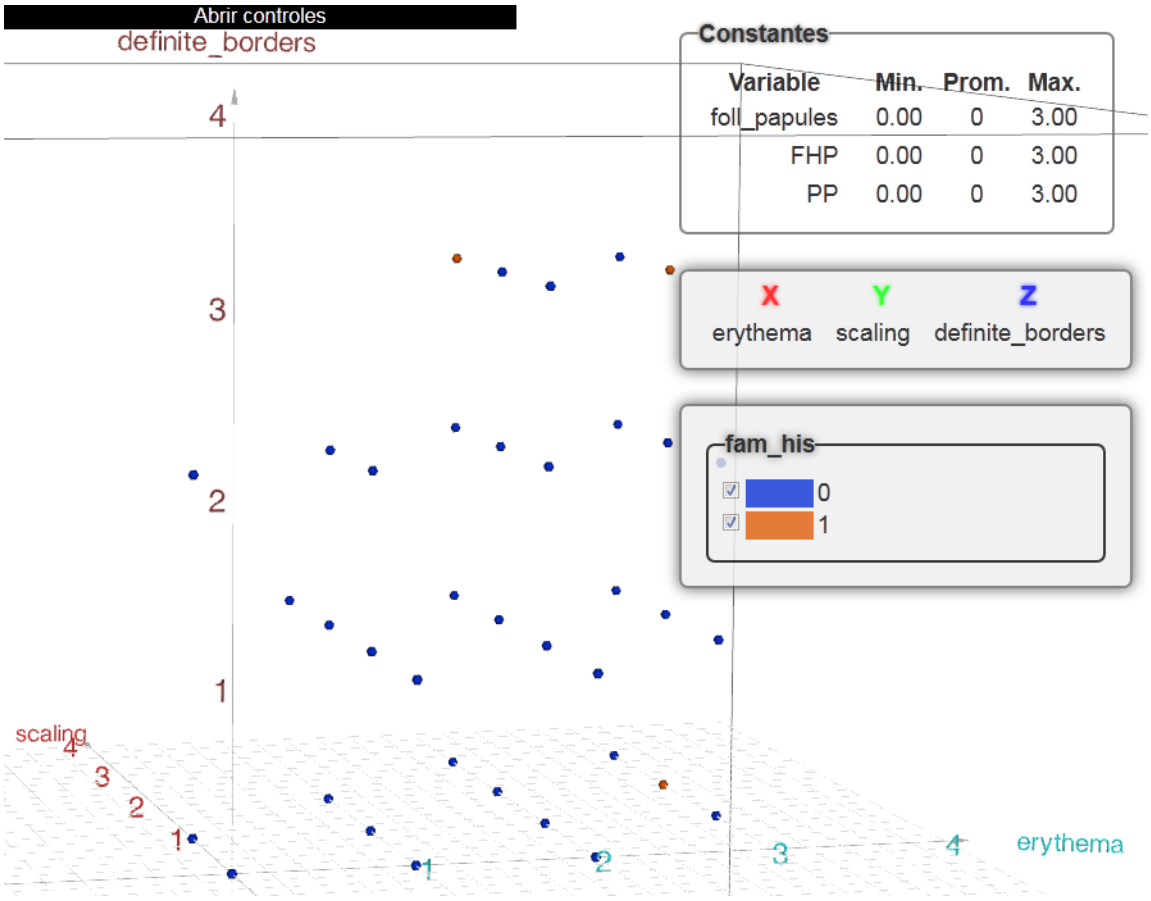


Figura 5.14: Resultado final de la visualización para el conjunto de datos "Dermatología" (ejemplo 5.7). Ambos algoritmos dan el mismo resultado.

5.8. Plantas Iris (*Iris Plants*)

Este conjunto de datos es llamado *Iris Plants*[7] creado por R.A. Fisher en 1988. Consta de 150 registros y cinco atributos, cuatro numéricos y uno simbólico. En la tabla 5.13 se describen los atributos de este conjunto.

Atributo	Descripción	Tipo	Ran./Val.
(0) sepal_len	Longitud en cm. del sépalo	Num.	[4.3, 7.9]
(1) sepal_w	Ancho del sépalo en cm.	Num.	[2, 4.4]
(2) petal_len	Longitud en cm. del pétalo	Num.	[1, 6.9]
(3) petal_w	Ancho del pétalo en cm.	Num.	[0.1, 2.5]
(4) class	Clasificación	Sim.	{setosa, versicolor, virginica}

Tabla 5.13: Descripción del conjunto "Plantas Iris". En negrita la variable simbólica. Total de registros: 150.

En este ejemplo se logra tener un ajuste de dos variables usando mínimos cuadrados, no así con MARS. Esto se debe a que MARS no logra encontrar un ajuste adecuado de los datos como se ve en la figura 5.15.

En la figura 5.16 se observa el resultado final de ambos métodos. Al ser pocos atributos, usando mínimos cuadrados se lograron graficar todos, sin embargo, con MARS no se consigue graficar una variable.

[7]Obtenido de http://archive.ics.uci.edu/ml/datasets/Iris 17-05-2013

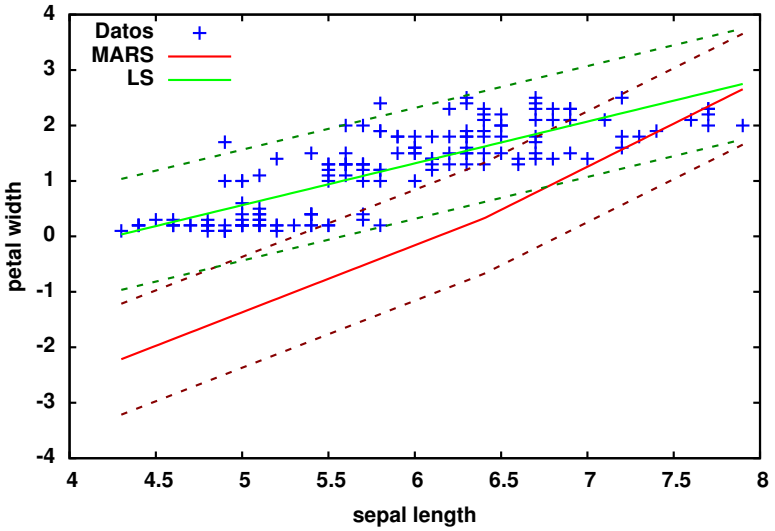


Figura 5.15: Comparativa en el ajuste usando mínimos cuadrados (verde) y MARS (rojo). MARS no obtiene un buen ajuste.

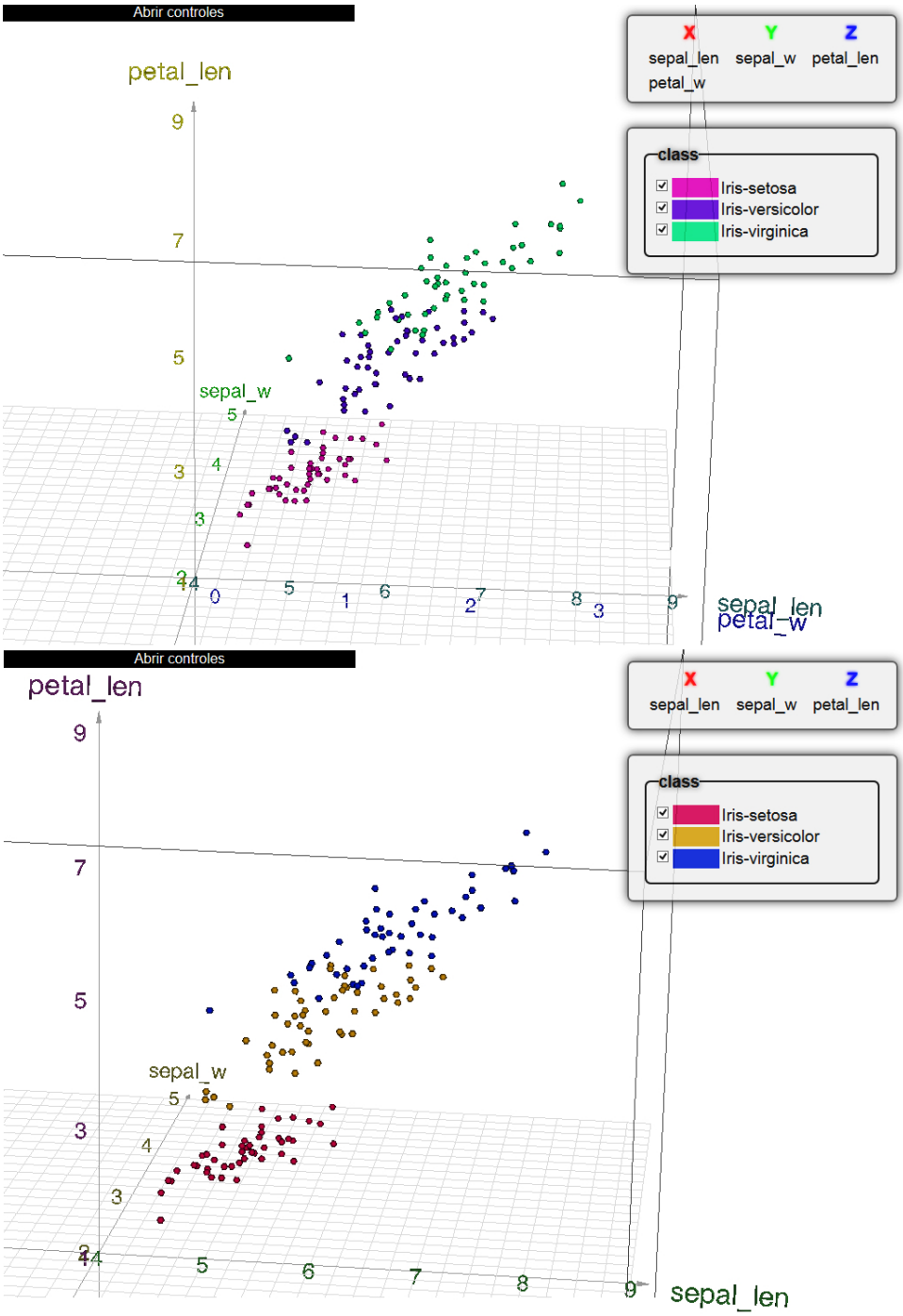


Figura 5.16: Resultados finales del conjunto "Plantas Iris" (ejemplo 5.8). Arriba mínimos cuadrados, abajo MARS.

5.9. Estadísticas de población de E.U.A (*USA - MapStats*)

Conjunto de datos llamado *USA MapStats*[8] el cual posee 3194 registros, 52 atributos, 51 son numéricos y uno simbólico. En la tabla 5.14 se muestra el nombre de cada uno de los atributos así como su tipo y rango. La descripción de ellos puede ser consultada en http://www.fedstats.gov/qf/download_data.html.

Atributo	**Tipo**	**Rango/Valores**	**Atributo**	**Tipo**	**Rango/Valores**
(0) fips	Sim.	3194 val. dif.	(26) HSG096211	Num.	[0, 98.4]
(1) PST045212	Num.	[71, 38 041 430]	(27) HSG495211	Num.	[0, 993 900]
(2) PST040210	Num.	[82, 37 253 956]	(28) HSD410211	Num.	[27, 12 433 172]
(3) PST120212	Num.	[-18.1, 25.6]	(29) HSD310211	Num.	[1.2, 4.77]
(4) POP010210	Num.	[82, 37 253 956]	(30) INC910211	Num.	[7 887, 61 290]
(5) AGE135211	Num.	[0, 13.3]	(31) INC110211	Num.	[19 344, 120 332]
(6) AGE295211	Num.	[0, 41]	(32) PVY020211	Num.	[0, 53.5]
(7) AGE775211	Num.	[3.7, 45.5]	(33) BZA010210	Num.	[0, 849 875]
(8) SEX255211	Num.	[28.7, 56.8]	(34) BZA110210	Num.	[0, 12 536 402]
(9) RHI125211	Num.	[3.9, 99.7]	(35) BZA115210	Num.	[-82.7, 385.8]
(10) RHI225211	Num.	[0, 84.7]	(36) NES010210	Num.	[0, 2 814 409]
(11) RHI325211	Num.	[0, 93.3]	(37) SBO001207	Num.	[0, 3 425 510]
(12) RHI425211	Num.	[0, 43.6]	(38) SBO315207	Num.	[0, 66.7]
(13) RHI525211	Num.	[0, 48.9]	(39) SBO115207	Num.	[0, 71.8]
(14) RHI625211	Num.	[0, 29.2]	(40) SBO215207	Num.	[0, 56.6]
(15) RHI725211	Num.	[0, 95.6]	(41) SBO515207	Num.	[0, 10.5]
(16) RHI825211	Num.	[3.2, 98.9]	(42) SBO415207	Num.	[0, 78]
(17) POP715211	Num.	[49, 100]	(43) SBO015207	Num.	[0, 56.2]
(18) POP645211	Num.	[0, 63.4]	(44) MAN450207	Num.	[0, 593 541 502]
(19) POP815211	Num.	[0, 95.9]	(45) WTN220207	Num.	[0, 598 456 486]
(20) EDU635211	Num.	[46.3, 98.6]	(46) RTN130207	Num.	[0, 455 032 270]
(21) EDU685211	Num.	[4.2, 72]	(47) RTN131207	Num.	[0, 80 800]
(22) VET605211	Num.	[0, 1 997 566]	(48) AFN120207	Num.	[0, 80 852 787]
(23) LFE305211	Num.	[4.3, 42.5]	(49) BPS030211	Num.	[0, 97 450]
(24) HSG010211	Num.	[48, 13 720 462]	(50) LND110210	Num.	[2, 570 640.95]
(25) HSG445211	Num.	[0, 93.7]	(51) POP060210	Num.	[0, 69 467.5]

Tabla 5.14: Atributos del conjunto "Estadísticas de población de E.U.A". En negrita la variable simbólica. Total de registros: 3194

Las variables 1, 2, 4, 24 y 28 tienen un comportamiento monótono creciente, esto fue detectado usando mínimos cuadrados. Con MARS solo las variables 1, 2, 4 y 28

[8]Descargado de http://www.fedstats.gov/ 20-05-2013

tuvieron este mismo comportamiento. En la tabla 5.15 se muestran los resultados finales para estas variables y se observa que con MARS hay más del 10 % de valores fuera de la franja para la pareja (1,24), sin embargo, en la gráfica 5.17 se puede ver que el ajuste entre los mínimos cuadrados y MARS es muy similar para estas dos variables.

		Outliers		**Ajuste**	
Variables	**MAD**	**LS**	**MARS**	**LS**	**MARS**
1,2	19077.5	64	67	Si	Si
1,4	19077.5	64	67	Si	Si
1,24	8834	311	337	Si	No
1,28	7242	278	241	Si	Si
2,4	19077.5	0	0	Si	Si
2,24	8834	296	319	Si	Si
2,28	7242	262	223	Si	Si
4,24	8834	296	319	Si	Si
4,28	7242	262	223	Si	Si
24,28	7242	245	253	Si	Si

Tabla 5.15: Resultados del algoritmo LS y MARS con el conjunto de datos "Estadísticas de población de E.U.A". Solo se muestran los pares de variables que tuvieron un ajuste. El valor máximo permitido de *outliers* es 319. Total de registros: 3194.

En la figura 5.17 se muestra el resultado del ajuste de ambos algoritmos para las variables 1 y 24. MARS detectó un punto de inflexión en (9962789, 4303987.31), sin embargo, no es suficiente para tener mejor ajuste que usando mínimos cuadrados. Las variables numéricas sobrantes no fueron posibles ajustar mediante un particionamiento. La variable simbólica corresponde a los estados y condados de Estados Unidos, por lo cual, cada valor únicamente aparece una vez, descartando así esta variable para ser mostrada en la visualización. Se detectaron dos variables constantes, la 13 y la 41. En total, con el algoritmo LS se pudieron graficar nueve variables de 52 mientras que con el algoritmo MARS solo ocho. Sin embargo, esto es más que lo que se puede graficar usando otro software, como SpotFire, Excel, Tableau entre otros. En la figura 5.18 se observa el resultado final de ambos algoritmos.

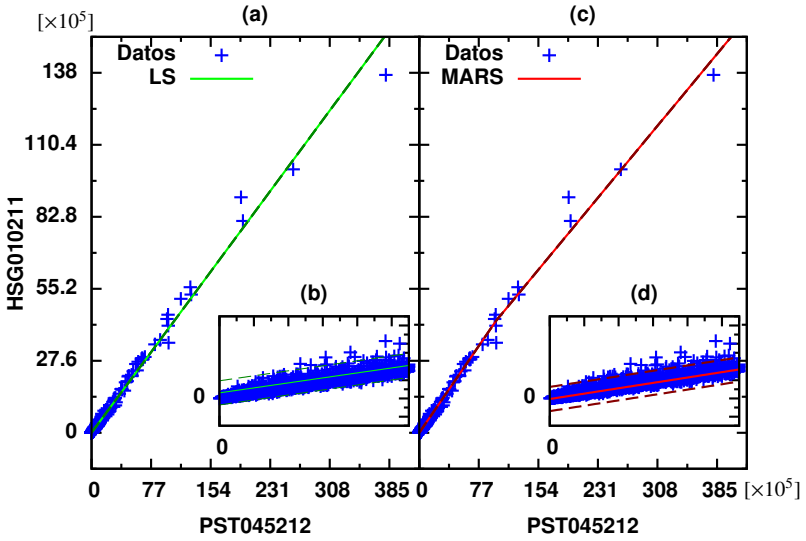


Figura 5.17: Comparativa en el ajuste usando mínimos cuadrados (a) y MARS (c) para las variables 1 y 24 del ejemplo 5.9. En (b) y (d) se muestra el *zoom* de una pequeña área donde se observa la franja.

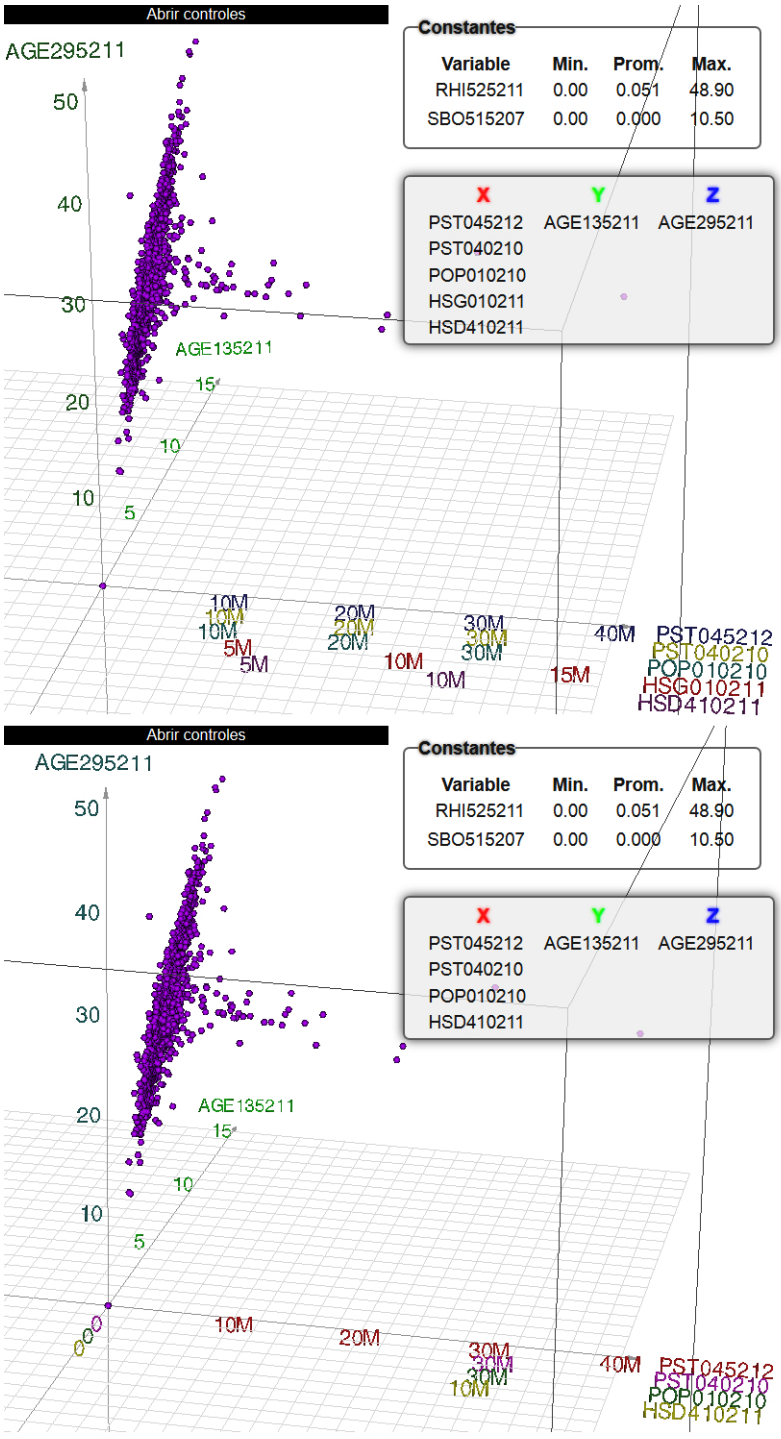


Figura 5.18: (Ejemplo 5.9). Resultados finales del conjunto de datos "Estadísticas de población de E.U.A". Arriba el algoritmo LS, abajo el algoritmo MARS. Se detectaron dos variables casi constantes.

5.10. Calidad del vino (*Wine Quality*)

Conjunto llamado *Wine Quality*[9]. Tiene 4898 registros con 12 atributos, 11 numéricos y uno simbólico:

- (0) fixed_acidity: Acidez fija, [3.8, 9.99].
- (1) volatile_acidity: Acidez volátil, [0.08, 1.1].
- (2) citric_acid: Acido cítrico, [0, 1.66].
- (3) residual_sugar: Azúcar residual, [0.6, 65.8].
- (4) chlorides: Cloruros, [0, 0.34].
- (5) free_sulfur_dioxide: Dióxido de azufre libre, [2, 289].
- (6) total_sulfur_dioxide: Dióxido de azufre total [9, 440].
- (7) density: Densidad, [0.98, 1.04].
- (8) pH: pH, [2.72, 3.82].
- (9) sulphates: Sulfatos, [0.22, 1.08].
- (10) alcohol: Cantidad de alcohol, [8, 14.2].
- **(11) quality**: Calidad del vino (entre 0 y 10).

Con estos datos no se logró encontrar ningún tipo de ajuste monotónico con ninguno de los dos métodos, es decir, cada variable numérica ira en un eje. Sin embargo, se detectan cuatro variables constantes o casi constantes. La variable simbólica que representa la calidad del vino posee siete valores diferentes de los cuales únicamente se grafican tres, esto se debe a que los otros valores tienen baja frecuencia. En la tabla 5.16 se observan los distintos valores de dicha variable así como su frecuencia. Por ultimo en la figura 5.19 se muestra la visualización.

En conclusión, se lograron visualizar en un mismo grafo ocho de las doce variables disponibles.

[9]Obtenido de http://archive.ics.uci.edu/ml/datasets/Wine+Quality 20-05-2013

Valores	Frecuencia
3	20
4	163
5	1457
6	2198
7	880
8	175
9	5

Tabla 5.16: Valores y frecuencia de la variable simbólica del conjunto de datos "Calidad del vino". Valores con frecuencia menor a 245 (5 % del total de datos) no se consideran. Total de registros: 4898.

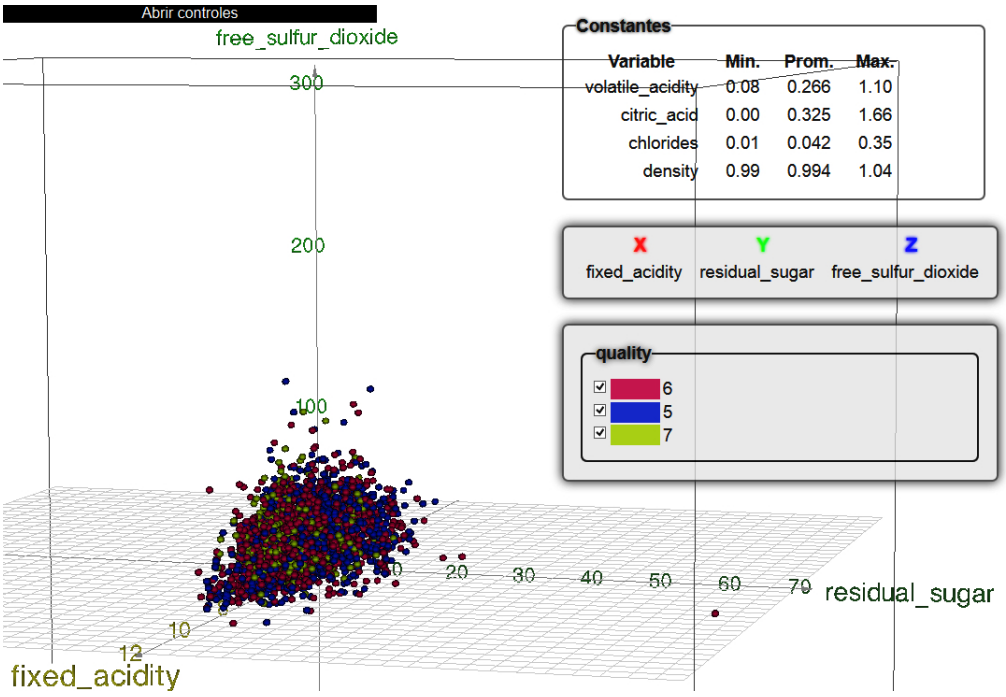


Figura 5.19: (Ejemplo 5.10). Resultado final del conjunto de datos "Calidad del vino". Ambos algoritmos dan el mismo resultado. Se detectaron cuatro valores constantes. Ver también figura 5.20.

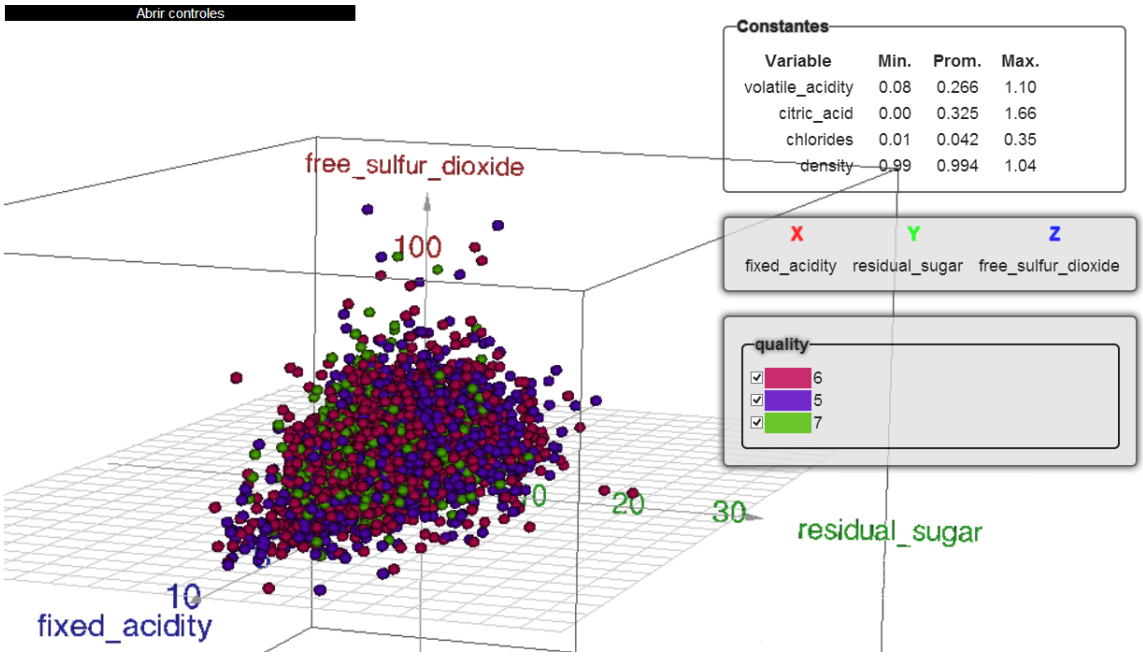


Figura 5.20: (Ejemplo 5.10). Resultado final del conjunto de datos "Calidad del vino". A diferencia de la figura 5.19, aquí se observa con mayor detalle la parte significativa de la gráfica, esto implica que algunos *outliers* queden fuera.

Capítulo 6

Conclusiones y trabajo futuro

Con el objetivo de mostrar la mayor cantidad de información en forma entendible para el usuario, se presentó un método alternativo de visualización de la información para datos multivariados que busca varios comportamientos predefinidos, entre ellos los monótonos dentro de las variables numéricas para poder graficarlas agrupadas sobre ejes. Todo el proceso es automático, presentando al usuario la mayor cantidad de relaciones posibles en una misma gráfica tridimensional, permitiendo interactuar con ella una vez generada, así como cambiar los ejes a visualizar en caso de que existan más de tres grupos de variables numéricas asignadas por el método. A diferencia de otros programas, como SpotFire de TIBCO Software, Tableau Software e incluso Excel, donde el usuario debe seleccionar de forma manual el tipo de gráfico a visualizar así como las variables, aquí se buscó automatizar este proceso a partir de que se definan algunos parámetros (porcentaje máximo permitido para valores fuera de la franja, valor de α en la bondad, etc.). Como pudo comprobarse en las pruebas y resultados, cuando existen comportamientos monótonos en las variables, la visualización resultante ofrece mayor información que solo graficando tres variables independientes, porque las variables con dicho comportamiento se grafican juntas con diferente escala, lo que permite al usuario tener un panorama general de cómo se están comportando los datos, permitiéndole detectar cosas ocultas en ellos (tendencias, patrones, etc.). Aun si no existe un comportamiento monótono, es posible tratar de ajustar algunas variables mediante un particionado tal como se explicó anteriormente, lo que permite aun obtener más información (si se encuentra una partición). En los casos que se detecten variables que varíen poco (casi constantes), la visualización las muestra, con lo que la gráfica resultante proporciona

mayor información. También se detectaron en algunos casos variables constantes o casi constantes, con lo cual, la visualización proporciona aún mayor información.

Para desplegar variables simbólicas, el algoritmo busca agruparlas y particionarlas, para desplegarlas sobre los ejes cartesianos ya obtenidos. Como último recurso, se recurre al color y la forma de los puntos desplegados.

Por otro lado, en el desarrollo del software (llamado VisRL) se trató de cumplir los cuatro requerimientos indispensables de una visualización de la información, propuestos por B. Shneiderman en [15], los cuales son resumen de los datos, zoom, filtrado y detalles bajo demanda.

En resumen, los objetivos mencionados en este trabajo se lograron cumplir porque se le presenta al usuario la mejor agrupación de las variables, que le permite observar las relaciones presentes en los datos de su interés.

Trabajando con un conjunto de hechos que tienen varias dimensiones o propiedades (datos multivariados), las principales aportaciones de este trabajo fueron:

1. Un método y su implementación para mostrar el mayor número de variables posibles, tanto numéricas como simbólicas, en una gráfica o despliegue tridimensional que facilita el entendimiento de los fenómenos o hechos registrados en los datos.

2. Un método y su implementación, que permite agrupar variables numéricas y simbólicas en un mismo eje, simplificando el entendimiento de los datos por parte del usuario. En fenómenos complejos, esta asociación de variables facilita captar cambios de fase (cuando los datos registrados tienen múltiples dimensiones).

3. El método 2 también proporciona reducción de dimensiones, porque las variables que se despliegan sobre un mismo eje están correlacionadas monotónicamente y la correlación se indica mediante los valores de cada variable, desplegados sobre el mismo eje. El método no "esconde" o elimina variables dependientes, sino las muestra todas juntas (sobre el mismo eje), porque a menudo el usuario o explorador de los datos encuentra útil ver cómo están relacionadas estas variables entre sí, a sabiendas de que unas dependen o varían con las otras.

Un punto que no se abordó pero que es importante y que queda como trabajo futuro, es el manejo de miles e incluso millones de registros, pues aplicar este modelo con grandes cantidades de datos puede ser poco óptimo, debido al tiempo de análisis que esto requiere (VisRL es lento para millones de datos), por lo cual, se deben buscar alternativas para optimizar el proceso, como tomar una muestra de todo el conjunto, aplicar *clustering*, etc. esto con la finalidad de reducir el tamaño de los datos para así poder disminuir el tiempo de análisis.

En este trabajo todas las variables se ajustaron a líneas, o segmentos de líneas, sin embargo, uno podría preguntarse por qué no ajustarlo a curvas, buscando el mismo comportamiento, es decir, que sea creciente o decreciente. Se propone como trabajo futuro.

Bibliografía

[1] Francisco de la Rosa Troyano, Rafael Martínez Gasca, Luis González Abril y Francisco Velasco Morente. Análisis de redes sociales mediante diagramas estratégicos y diagramas estructurales. *REDES-Revista hispana para el Análisis de redes sociales*, 8(0), 08 2005.

[2] George G. Robertson, Stuart K. Card, and Jack D. Mackinlay. Information visualization using 3d interactive animation. *Commun. ACM*, 36(4):57–71, April 1993.

[3] Daniel A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, January 2002.

[4] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[5] Larkin, J. H. y Simon, H. A. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, (11):65–100, 1987.

[6] Charles Kemp and Joshua B. Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692, August 2008.

[7] Abraham Silberschatz, Henry Korth, and S. Sudarshan. *Database Systems Concepts*. McGraw-Hill, Inc., New York, NY, USA, 6 edition, 2010.

[8] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[9] I. Herman, G. Melancon, and M.S. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24 –43, jan-mar 2000.

[10] Gilberto Lorenzo Martínez Luna y Guzmán Arenas A. Búsqueda de patrones de comportamiento en cubos de datos. In *2nd International Workshop on Data Mining*, pages 163–179, Texcoco, Edo. de México, Septiembre 2000.

[11] Guzmán Arenas A. Uso y diseño de Mineros de Datos. *Soluciones avanzadas*, Junio 1996.

[12] Gilberto Lorenzo Martínez Luna. Minería de datos: Cómo hallar una aguja en un pajar. *Revista Ciencia*, 62(3), Jul-Sep 2011.

[13] Dong Xin Jiawei Han, Hong Cheng and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Min. Knowl. Discov.*, 15(1):55–86, 2007.

[14] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.

[15] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336 –343, 1996.

[16] Robert Spence and A. Press. *Information Visualization*. Addison Wesley, December 2000.

[17] David P. Tegarden. Business information visualization. *Communications of the Association for Information Systems*, 1, 01 1999.

[18] Jin Zhang. *Visualization for Information Retrieval (The Information Retrieval Series)*. Springer, 1 edition, 2007.

[19] Chaomei Chen. *Information Visualization: Beyond the Horizon*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[20] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, January 2009.

[21] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4(5):235–282, October 1994.

[22] Brian Johnson and Ben Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on Visualization '91*, VIS '91, pages 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[23] Jun Rekimoto and Mark Green. The information cube: Using transparency in 3d information visualization. In *In Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93*, pages 125–132, 1993.

[24] Eugene Garfield and A. Pudovkin. The HistCite system for mapping and bibliometric analysis of the output of searches using the isi web of knowledge. In *2004 ASIS&T Annual Meeting*, 2004.

[25] E. Garfield, S.W. Paris, and W.G. Stock. Histcite: A software tool for informetric analysis of citation linkage. *Information Wissenschaftund Praxis*, 57(8):391–400, 2006.

[26] Eugene Garfield, A. I. Pudovkin, and Soren W. Paris. A bibliometric and historiographic analysis of the work of tony van raan: a tribute to a scientometrics pioneer and gatekeeper. *Research Evaluation*, pages 161–172, September 2010.

[27] Lutz Bornmann and Werner Marx. HistCite analysis of papers constituting the h index research front. *Journal of Informetrics*, 6(2):285–288, April 2012.

[28] S. Raja and R. Balasubramani. Scientometric study of the research publication on malaria 2003-2007: A global perspective. *International Research Journal of Library, Information and Archival Studies*, 1(3):114–125, 2011.

[29] Christopher G. Healey and James T. Enns. A perceptual colour segmentation algorithm. Technical report, Vancouver, Canada, 1996.

[30] Luigi Troiano, Cosimo Birtolo, and Gennaro Cirillo. Interactive Genetic Algorithm for choosing suitable colors in User Interface. 2009.

[31] Achim Zeileis, Kurt Hornik, and Paul Murrell. Escaping rgbland: Selecting colors for statistical graphics. *Comput. Stat. Data Anal.*, 53(9):3259–3270, July 2009.

[32] Jerome H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[33] Catherine Leung and Andor Salga. Enabling webgl. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1369–1370, New York, NY, USA, 2010. ACM.